MI890

Intel® Pineview-M/ Pineview-D +ICH8M
Luna-Pier / Luna-Pier Refresh Platform
Mini-ITX Motherboard

USER'S MANUAL

Version 1.0

Acknowledgments

AMI BIOS is a trademark of American Megatrends Inc.
PS/2 is a trademark of International Business Machines Corporation.
Intel and Atom 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

Introduction 1

Checklist 2
MI890 Specifications 3
Board Dimensions 4

Installations 5

Setting the Jumpers 6
Connectors on MI890 10

BIOS SETUP 17

Drivers Installation 37

Appendix 45

A. I/O Port Address Map 45
B. Interrupt Request Lines (IRQ) 48
C. Watchdog Timer Configuration 49
D. Digital I/O Sample Code 54

This page is intentionally left blank.

Introduction

MI890 MINI ITX MOTHERBOARD

MI890 EDGE CONNECTORS

Checklist

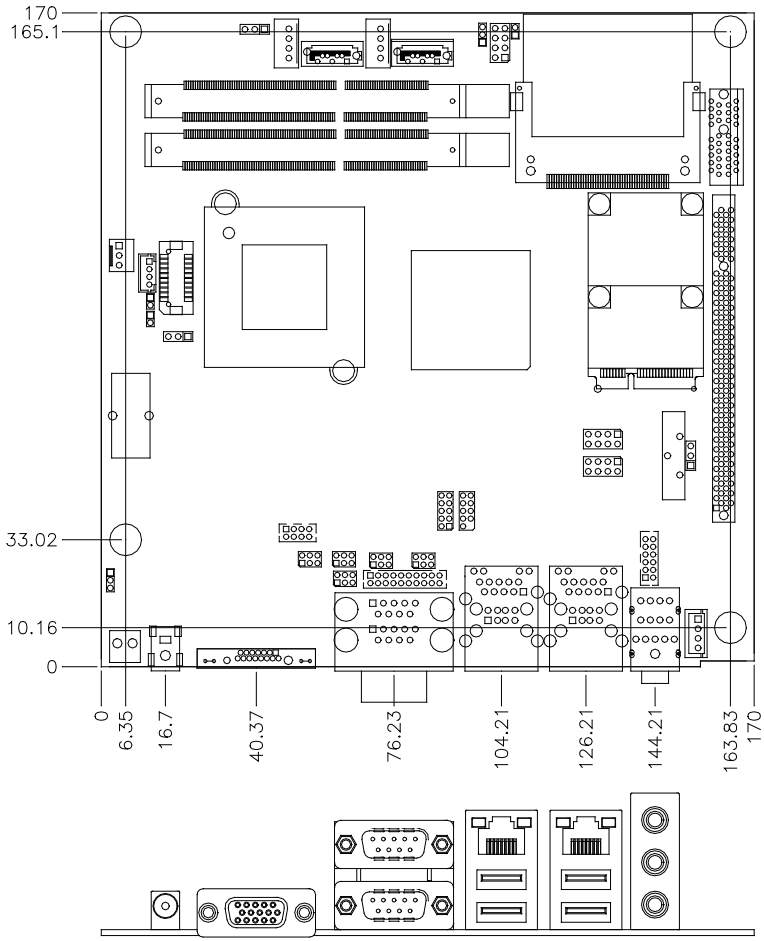
Your MI890 package should include the items listed below.

- The MI890 Intel® Atom Mini-ITX motherboard
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility
- Serial ATA cable

MI890 Specifications

Product Name	**MI890-N4/ MI890-D4 / MI890-D5**
Form Factor	MiniITX
CPU Type	Intel® Atom™ SC N455 (512KB L2 cache, 1 Core/ 2 Threads. TDP=6.5W) Intel® Atom™ SC D425 (512KB L2 cache, 1 Core/ 2 Threads. TDP=10W) Intel® Atom™ DC D525 (1MB L2 cache, 2 Core/ 2 Threads. TDP=13W) 22mm x 22mm, Micro-FCBGA8
CPU Speed	**1.66GHz (N455) / 1.80GHz(D425/D525)**
Chipset	ICH8M: 31mm x 31mm, 676-pin T-PBGA (2.4W)
BIOS	AMI BIOS
Memory	DDR3-800 (Single channel) SO-DIMM x2 (Non-ECC), Max.= 2GB(N455) / 4GB(D425/D525)
VGA	Intel® Integrated Graphics Controller Supports DirectX 9 Graphic (200MHz for N455 ; 400MHz for D425/D525) OpenGL 1.4
LVDS	18-bit one channels LVDS interface w/DF13 socket x1
LAN	Intel 82583V x 2 for Dual GbE
USB	ICH8M built-in USB 2.0 host controller, support 9 ports
Serial ATA Ports	ICH8M built-in SATA controller, supports 2 ports
Parallel IDE	ICH8M built-in one channel Ultra DMA 33/66/100, for CF Type II (Component side)
Audio	Intel ICH8M built-in HD audio controller w/ Realtek ALC269 Codec with integrated 2W amplifier (4 Ohm) Supports 2 Channel audio (Line-out & Line-in)
LPC I/O	Winbond NCT6627UD : COM1 (RS232/422/485), COM2~COM4(RS232) with pin-9 with power for 2 ports (500 mA for each port) [COM3/4] Hardware monitor (2 thermal inputs, 4 voltage monitor inputs, VID0-4 & 1 x fan header)
Digital IO	4 in & 4 out
Expansion Slots	Mini PCI-e x 1 w/USB for WiFi or TV-tuner module (Reserved screw holes for half-mini type also) PCI slot x 1 + PCI-Express(1x) slot x1
Edge Connector	DC-Jack x 1 DB15 connector x1 for VGA Dual DB9 Stack connector x 1 for COM1/ COM2 RJ45 + Dual USB stack connector x1 for LAN 1 + USB 1/2 RJ45 + Dual USB stack connector x1 for LAN 2 + USB 3/4 Audio 3-port connector x 1 (Line-out , Line-in& Mic)
On Board Header/Connector	2x4 pin header x2 for 4 USB ports DF13 socket x 1 for LVDS(18-bit single channel) DF11 socket 20-pin connector x 1 for COM3, COM4 CF type II connector x1 SATA II connector x2 2x5 pin header x1 for Digital I/O 2x6 pin header x1 for front audio 4-pin header x2 for SATA power (Mini type)
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
Others	Optional EuP feature (Fintek F75160) UL 60950-1 2nd Edition compatible
Power Connector	12V ~ 24V DC-in
Board Size	170mm x 170mm

Board Dimensions



Installations

This section provides information on how to use the jumpers and connectors on the MI890 in order to set up a workable system. The topics covered are:

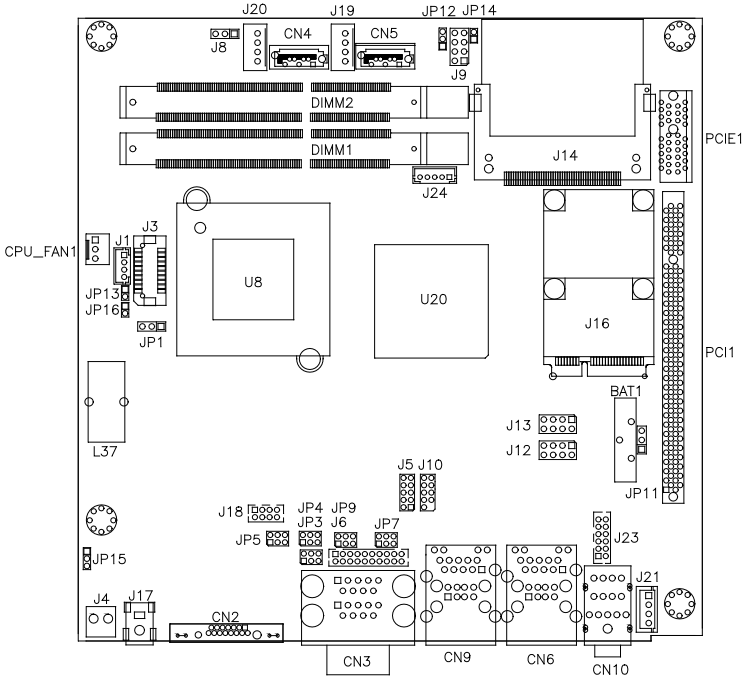
Setting the Jumpers..... 6
Connectors on MI890 10

Setting the Jumpers

Jumpers are used on MI890 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 MI890 and their respective functions.

Jumper Locations on MI890 7
JP1: LCD Panel Power Selection 8
JP3, JP4, JP5: RS232/422/485 (COM1) Selection..... 8
JP7, JP9: COM4 RS232 RI/+5V/+12V Power Setting 8
JP11: Clear CMOS Setting 8
JP12, JP14: ATX / AT Mode Select Setting..... 9
JP15: EuP Control (option) 9
JP13: +3VS For ID381 9
JP16: +5VS For ID381 9

Jumper Locations on MI890



Jumper Locations on MI890 7
JP1: LCD Panel Power Selection 8
JP3, JP4, JP5: RS232/422/485 (COM1) Selection..... 8
JP7, JP9: COM4 RS232 RI/+5V/+12V Power Setting 8
JP11: Clear CMOS Setting 8
JP12, JP14: ATX / AT Mode Select Setting..... 9
JP15: EuP Control (option)..... 9
JP13: +3VS For ID381 9
JP16: +5VS For ID381 9

JP1: LCD Panel Power Selection

JP1	LCD Panel Power
1 2 3	3.3V
1 2 3	5V

JP3, JP4, JP5: RS232/422/485 (COM1) Selection

2 4 6
1 3 5

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

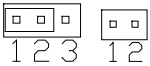
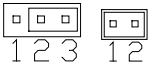
JP7, JP9: COM4 RS232 RI/+5V/+12V Power Setting

JP7/JP9	Setting	Function
1 2 5 6	Pin 1-2 Short/Closed	+12V
	Pin 3-4 Short/Closed	RI
	Pin 5-6 Short/Closed	+5V

JP11: Clear CMOS Setting

JP11	Function
1 2 3	Normal
1 2 3	Clear CMOS

JP12, JP14: ATX / AT Mode Select Setting

JP12 / JP14	Setting	Function
	JP12 1-2, JP14 open	ATX Mode
	JP12 2-3, JP14 short	AT Mode

JP15: EuP Control (option)

JP15	Setting	Function
	Pin 1-2 Short/Closed	Normal On **Default**
	Pin 2-3 Short/Closed	EuP Off

JP13: +3VS For ID381

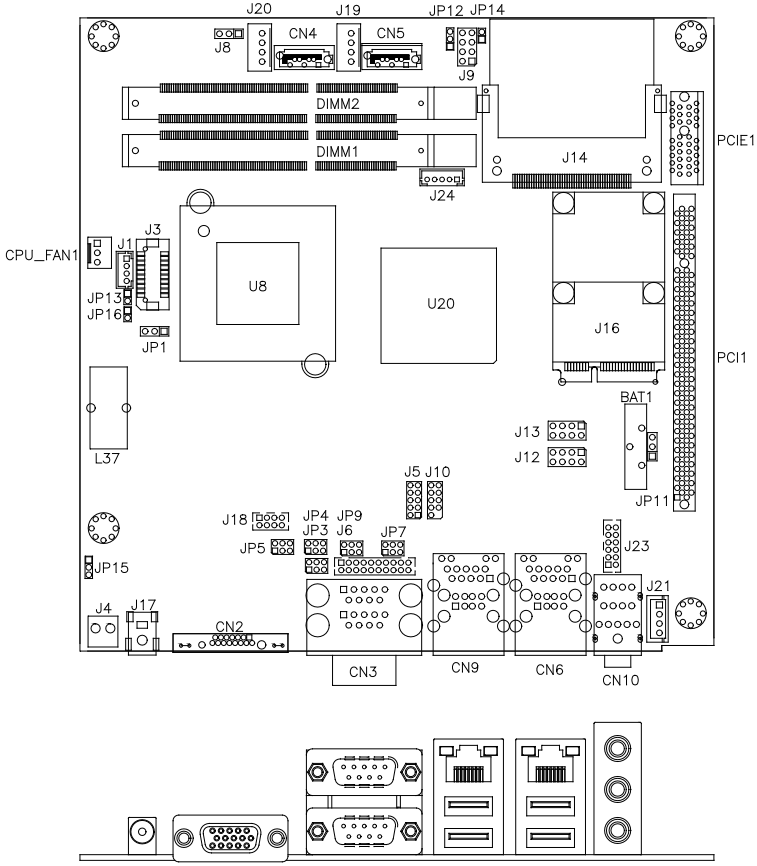
JP16: +5VS For ID381

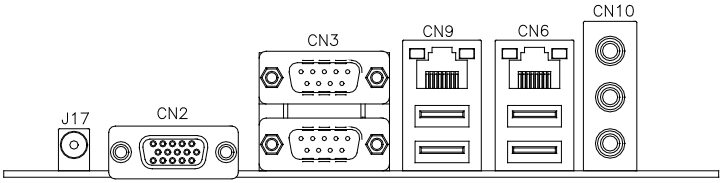
Connectors on MI890

The connectors on MI890 allows you to connect external devices such as keyboard, floppy disk drives, hard disk drives, printers, etc. The following table lists the connectors on MI890 and their respective functions.

Connector Locations on MI890 11
J17: DC_IN Jack (+12V~24V) 12
CN2: VGA Connector 12
CN3: COM1 and COM2 Serial Port Connectors 12
CN6: 10/100/1000 RJ-45 and USB1/2 Ports 13
CN9:10/100/1000 RJ-45 and USB3/4Ports 13
CN10: Line-in, Line-out & Microphone Connector 13
CPU_FAN1: CPU Fan Power Connector (DC Fan) 13
J1: LCD Backlight Connector 13
J3: LVDS Connectors (18-bit) 13
J4: DC_IN Connector 14
J5: Digital I/O 14
J8: Power LED Connector 14
J9: System Function Connector 14
J10: SPI Flash Connector (factory use only) 14
J12: USB5/USB6 Connector 15
J13: USB7/USB8 Connector 15
J14: Compact Flash Connector 15
J15: Audio Connector (DF11 Connector) 15
J16: Mini PCIE Connector 15
J18: PS/2 Keyboard and PS/2 Mouse Connectors 15
J19/J20: SATA Power Output 15
J21: Audio Amplifier 15
CN4/CN5: SATA Connectors 15
J24: Smart Battery Connector (option) 16
ID381 LVDS Daughter Card 16
J1: LVDS 24bit Output 16
J2: LVDS Input from MI890 16
CN1: DVI-I Output 16

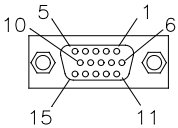
Connector Locations on MI890





J17: DC_IN Jack (+12V~24V)

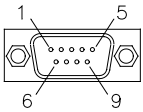
CN2: VGA Connector



VGA

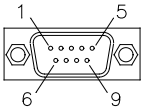
Signal Name	Pin #	Pin #	Signal Name
Red	1	2	Green
Blue	3	4	N.C.
GND	5	6	GND
GND	7	8	GND
N.C.	9	10	GND
N.C.	11	12	N.C.
HSYNC	13	14	VSYNC
NC	15		

CN3: COM1 and COM2 Serial Port Connectors



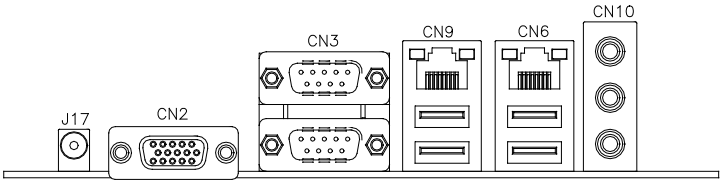
COM2

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



COM1

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



CN6: 10/100/1000 RJ-45 and USB1/2 Ports

CN9:10/100/1000 RJ-45 and USB3/4Ports

CN10: Line-in, Line-out & Microphone Connector

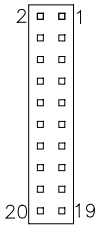
CPU_FAN1: CPU Fan Power Connector (DC Fan)

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

J1: LCD Backlight Connector

Pin #	Signal Name
1	+12V
2	Backlight Enable
3	Brightness Control
4	Ground

J3: LVDS Connectors (18-bit)

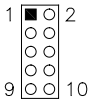


Signal Name	Pin #	Pin #	Signal Name
TX0-	2	1	TX0+
Ground	4	3	Ground
TX1-	6	5	TX1+
5V/3.3V	8	7	Ground
NC	10	9	NC
TX2-	12	11	TX2+
Ground	14	13	Ground
TXC-	16	15	TXC+
5V/3.3V	18	17	ENABKL
+12V	20	19	+12V

J4: DC_IN Connector

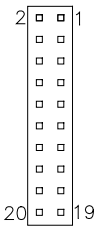
Note: J4 input voltage is +12V only.

J5: Digital I/O



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

J6: COM3/ COM4 RS232 Serial Ports (option)

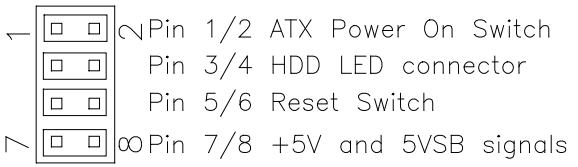


Signal Name	Pin #	Pin #	Signal Name
DSR1	2	1	DCD1
RTS1	4	3	RXD1
CTS1	6	5	TXD1
RI1	8	7	DTR1
NA	10	9	Ground
DSR2	12	11	DCD2
RTS2	14	13	RXD2
CTS2	16	15	TXD2
RI2	18	17	DTR2
NA	20	19	Ground

J8: Power LED Connector

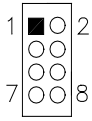
Pin #	Signal Name
1	+5V
2	NC
3	Ground

J9: System Function Connector



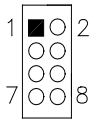
J10: SPI Flash Connector (factory use only)

J12: USB5/USB6 Connector



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

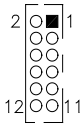
J13: USB7/USB8 Connector



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

J14: Compact Flash Connector

J15: Audio Connector (DF11 Connector)

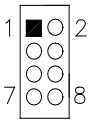


Signal Name	Pin #	Pin #	Signal Name
LINEOUT R	2	1	LINEOUT L
Ground	4	3	JD FRONT
LINEIN R	6	5	LINEIN L
Ground	8	7	JD_LINEIN
MIC-In	10	9	MIC_L
Ground	12	11	JD MIC1

J16: Mini PCIE Connector

J18: PS/2 Keyboard and PS/2 Mouse Connectors

Remarks: For testing use only.



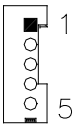
Signal Name	Pin	Pin	Signal Name
Vcc	1	2	VCC
MDA	3	4	KBDA
MCL	5	6	KBCL
Ground	7	8	Ground

J191/J201: SATA Power Output

J21: Audio Amplifier

CN4/CN5: SATA Connectors

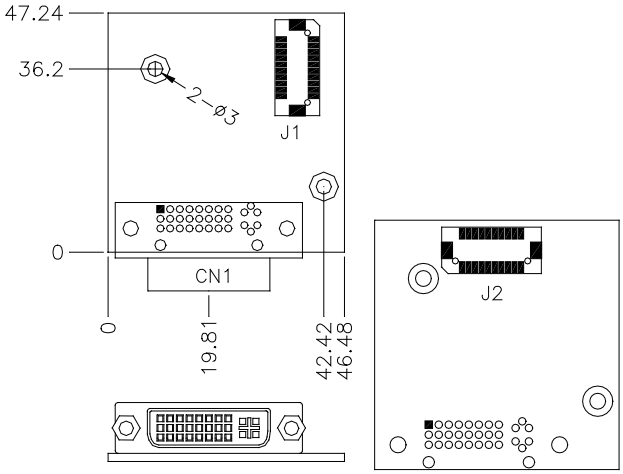
J24: Smart Battery Connector (option)



Pin #	Signal Name
1	PCIRST-
2	EMTSMI-
3	Ground
4	SMBDATA
5	SMBCLK

ID381 LVDS Daughter Card

The ID381 card supports 24-bit LVDS and DVI interface.



Top View **Bottom View**

J1: LVDS 24bit Output
J2: LVDS Input from MI890
CN1: DVI-I Output

BIOS SETUP

This chapter describes the different settings available in the AMI (American Megatrends, Inc.) BIOS that comes with the board. The topics covered in this chapter are as follows:

BIOS Introduction..... 18
BIOS Setup..... 18
Main BIOS Setup 19
Advanced Settings..... 20
PCIPnP Settings..... 28
Boot Settings..... 29
Security Settings 31
Advanced Chipset Settings 32
Exit Setup..... 35
Load Optimal Defaults..... 35
Load Failsafe Defaults 35

BIOS Introduction

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

BIOS Setup

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

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

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

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

Main BIOS Setup

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

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

System Overview		Use[ENTER], [TAB] or [SHIFT-TAB] to select a field.
Processor		
Intel(R) Atom (TM) CPU D525	@ 1.80GHz	Use [+] or [-] to configure system Time.
Speed : 1800MHz		
Count : 1		
System Memory		
Size : 2038MB		<- Select Screen
		↑↓ Select Item
System Time	**[17:00:00]**	+- Change Field
System Date	[Tue 05/25/2010]	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

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

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Advanced Settings	Configure CPU.
WARNING: Setting wrong values in below sections may cause system to malfunction.	
► CPU Configuration	
► IDE Configuration	
► Super IO Configuration	
► Hardware Health Configuration	<- Select Screen
► ACPI Configuration	↑↓ Select Item
► AHCI Configuration	Enter Go to Sub Screen
► APM Configuration	F1 General Help
► USB Configuration	F10 Save and Exit
	ESC Exit

The Advanced BIOS Settings contains the following sections:

► CPU Configurations
► IDE Configuration
► Super IO Configuration
► Hardware Health Configuration
► ACPI Configuration
► AHCI Configuration
► APM 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

Advanced	
Configure advanced CPU settings	**Disabled for WindowsXP**
Module Version:3F.18	
Manufacturer: Intel	
Intel(R) Atom (TM) CPU D525 @ 1.80GHz	
Frequency : 1.80GHz	
FSB Speed : 800MHz	
Cache L1 : 48KB	
Cache L2 : 1024KB	
Ratio Actual Value : 9	
	<- Select Screen
	↑↓ Select Item
Max CPUID Value Limit **[Disabled]**	+- Change Option
Execute-Disable Bit Capability [Enabled]	F1 General Help
Hyper Threading Technology [Enabled]	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

Max CPUID Value Limit

Disabled for WindowsXP.

Execute-Disable Bit Capability

XD can prevent certain classes of malicious buffer overflow attacks when combined with a supporting OS.

Hyper Threading Technology

Enabled for Windows XP and Linux (OS optimized for Hyper-Threading Technology) and Disabled for other OS (OS not optimized for Hyper-Threading Technology). When Disabled, only one thread per enabled core is enabled.

BIOS SETUP UTILITY

Advanced		
IDE Configuration		**Options**
ATA/IDE Configuration	**[Enhanced]**	**Disabled**
Configure SATA as	[IDE]	**Compatible**
		Enhanced
► Primary IDE Master	: [Not Detected]	
► Primary IDE Slave	: [Not Detected]	
► Secondary IDE Master	: [Not Detected]	
► Secondary IDE Slave	: [Not Detected]	**<-** **Select Screen**
► Third IDE Master	: [Not Detected]	**↑↓** **Select Item**
► Third IDE Slave	: [Not Detected]	**+-** **Change Option**
► Fourth IDE Master	: [Not Detected]	**F1** **General Help**
► Fourth IDE Slave	: [Not Detected]	**F10** **Save and Exit**
		ESC **Exit**
Hard Disk Write Protect	[Disabled]	
IDE Detect Time Out (Sec)	[35]	
ATA(PI) 80Pin Cable Detection	[Host & Device]	

The IDE Configuration menu is used to change and/or set the configuration of the IDE devices installed in the system.

ATA/IDE Configuration

(1) Disabled.
(2) Compatible.
(3) Enhanced

Configure SATA as

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

BIOS SETUP UTILITY

Advanced		
Configure Win627UHG Super IO Chipset		**Allows BIOS to Select**
		Serial Port Base
Serial Port1 Address	**[3F8/IRQ4]**	**Addresses**
Serial Port2 Address	[2F8/IRQ3]	
Serial Port3 Address	[3E8]	
Serial PortC IRQ	[IRQ11]	**<-** **Select Screen**
Serial Port4 Address	[2E8]	**↑↓** **Select Item**
Serial PortD IRQ	[IRQ10]	**+-** **Change Option**
Restore on AC Power Loss	[Power Off]	**F1** **General Help**
COM1 RS485 Auto Flow Contorl	[Disabled]	**F10** **Save and Exit**
		ESC **Exit**

Onboard Serial Port

The default values are:

Serial Port 1:	3F8/IRQ4
Serial Port 2:	2F8/IRQ3
Serial Port 3:	3E8/IRQ11
Serial Port 4:	2E8/IRQ10

Restore on AC Power Loss

This field sets the system power status whether *Power On or Power Off* when power returns to the system from a power failure situation.

BIOS SETUP UTILITY

Advanced

Hardware Health Configuration		Options
System Temperature	:42°C/107°F	Disabled
CPU Temperature	:37°C/98°F	70°C/158°F
CPU FAN Speed	:6750 RPM	75°C/167°F
		80°C/176°F
		85°C/185°F
Vcore	:1.136 V	90°C/194°F
+5VS	:5.304 V	95°C/203°F
+3VS	:3.200 V	
12 V	:11.904 V	
3.3V	:3.408 V	
1.5V	:1.504V	<- Select Screen
VBAT	: 3.536V	↑↓ Select Item
CPU Smart Fan	[Disabled]	+- Change Option
ACPI Shutdown Temperature	[Disabled]	Tab Select Field
		F1 General Help
		F10 Save and Exit
		ESC Exit

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

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

Advanced	
ACPI Settings	**General ACPI Configuration settings**
►**General ACPI Configuration** ►Advanced ACPI Configuration ►Chipset ACPI Configuration	**<- Select Screen** **↑↓ Select Item** **Enter Go to Sub Screen** **F1 General Help** **F10 Save and Exit** **ESC Exit**

BIOS SETUP UTILITY

Advanced	
General ACPI Configuration	**Select the ACPI state used for System Suspend.**
Suspend mode **[S1 (POS)]**	**<- Select Screen** **↑↓ Select Item** **+- Change Option** **F1 General Help** **F10 Save and Exit** **ESC Exit**

Suspend Mode

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

BIOS SETUP UTILITY

Advanced	
Advance ACPI Configuration	**Enable RSDP pointers to 64-bit Fixed System Description Tables.** **Different ACPI version Has some addition**
ACPI Version Features **[ACPI v1.0]** ACPI APIC support [Enabled]	**<- Select Screen** **↑↓ Select Item** **+- Change Option** **F1 General Help** **F10 Save and Exit** **ESC Exit**

BIOS SETUP UTILITY

Advanced	
South Bridge ACPI Configuration	**Options**
Energy Lake Feature **[Disabled]**	**Enabled**
APIC ACPI SCI IRQ [Disabled]	**Disabled**
USB Device Wakeup From S3/S4 [Disabled]	
	<- Select Screen
	↑↓ Select Item
	+- Change Option
	F1 General Help
	F10 Save and Exit
	ESC Exit

BIOS SETUP UTILITY

Advanced	
AHCI Settings	**While entering setup, BIOS auto detect the presence of IDE device. This displays the status of auto detection of IDE devices.**
AHCI Port0 **[Not Detected]**	
AHCI Port1 [Not Detected]	
AHCI Port2 [Not Detected]	
	<- Select Screen
	↑↓ Select Item
	Enter Go to Sub Screen
	F1 General Help
	F10 Save and Exit
	ESC Exit

BIOS SETUP UTILITY

Advanced	
APM Configuration	**Enable or disable APM.**
Power Management/APM **[Enabled]**	
Power Button Mode [On/Off]	
Resume On Ring Disabled	**<- Select Screen**
Resume On PME# Disabled	**↑↓ Select Item**
Resume On RTC Alarm Disabled	**+- Change Option**
	F1 General Help
	F10 Save and Exit
	ESC Exit

Power Management/APM
By default, this field is set to *Enabled*.

Power Button Mode
Go into On/Off, or Suspend when power button is pressed.

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 management event) 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

Advanced	
USB Configuration	**Enables support for legacy USB. AUTO option disables legacy support if no USB devices are connected.**
USB Devices Enabled: None	
Legacy USB Support **[Enabled]** USB 2.0 Controller Mode [HiSpeed] BIOS EHCI Hand-Off [Enabled] Legacy USB1.1 HC Support [Enabled]	**<- Select Screen** **↑↓ Select Item** **+- Change Option** **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

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

USB 2.0 Controller Mode

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

BIOS EHCI Hand-Off

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

Legacy USB1.1 HC Support

Support USB1.1 HC.

PCIPnP Settings

This option configures the PCI/PnP settings.

BIOS SETUP UTILITY

Main Advanced PCIPnP Boot Security Chipset Exit

Advanced PCI/PnP Settings

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

Plug & Play O/S	**[No]**
Allocate IRQ to PCI VGA	[Yes]
IRQ3	[Available]
IRQ4	[Available]
IRQ5	[Available]
IRQ7	[Available]
IRQ9	[Available]
IRQ10	[Available]
IRQ11	[Available]
IRQ14	[Available]
IRQ15	[Available]
DMA Channel 0	[Available]
DMA Channel 1	[Available]
DMA Channel 3	[Available]
DMA Channel 5	[Available]
DMA Channel 6	[Available]
DMA Channel 7	[Available]
Reserved Memory Size	[Disabled]

NO: lets the BIOS Configure all the Devices in the system. YES: lets the operating system configure Plug and Play (PnP) devices not required for boot if your system has a Plug and Play operating system.

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

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.

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.

IRQ#

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

Boot Settings

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Boot Settings	Configure Settings during System Boot.
►Boot Settings Configuration	<- Select Screen ↑↓ Select Item Enter Go to Sub Screen F1 General Help F10 Save and Exit ESC Exit

BIOS SETUP UTILITY

Boot

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

Quick Boot

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

Quite Boot

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

AddOn ROM Display Mode

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

Bootup Num-Lock

This select the power-on state for numlock.

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

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

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

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

Security Settings

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

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

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	**Security**	Chipset	Exit

Security Settings	**Install or Change the Password.**
Supervisor Password : Not Installed	
User Password : Not Installed	
Change Supervisor Password	**<- Select Screen**
Change User Password	**↑↓ Select Item**
	Enter Change
Boot Sector Virus Protection [Disabled]	**F1 General Help**
	F10 Save and Exit
	ESC Exit

Advanced Chipset Settings

This setting configures the north bridge and south bridge settings. WARNING! Setting the wrong values may cause the system to malfunction.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Advanced Chipset Settings	Configure North Bridge features.
WARNING: Setting wrong values in below sections may cause system to malfunction.	
► North Bridge Configuration	<- Select Screen
► South Bridge Configuration	↑↓ Select Item
	Enter Go to Sub Screen
	F1 General Help
	F10 Save and Exit
	ESC Exit

BIOS SETUP UTILITY

	Chipset
North Bridge Chipset Configuration	Options
PCI MMIO Allocation: 4GB To 3072MB	Enabled
DRAM Frequency [Auto]	Disabled
Configure DRAM Timing by SPD [Enabled]	
Initiate Graphics Adapter [IGD]	
Internal Graphics Mode Select [Enabled, 8MB]	<- Select Screen
PEG Port Configuration	↑↓ Select Item
	+- Change Option
► Video Function Configuration	F1 General Help
	F10 Save and Exit
	ESC Exit

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.

Initiate Graphic Adapter

Select which graphics controller to use as the primary boot device. This option, by default, is set to IGD.

Internal Graphics Mode Select

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

BIOS SETUP UTILITY

Chipset	
Video Function Configuration	**Options**
DVMT Mode Select **[DVMT Mode]**	**Fixed Mode**
DVMT/FIXED Memory [256MB]	**DVMT Mode**
Boot Display Device [CRT]	
Flat Panel Type [1024x768]	
Spread Spectrum Clock [Disabled]	**<- Select Screen**
	↑↓ Select Item
	+- Change Option
	F1 General Help
	F10 Save and Exit
	ESC Exit

DVMT Mode Select

Select the control mode of memory built-in graphics capabilities. This option, by default, is set to DVMT Mode.

DVMT/FIXED Memory

Sets the maximum memory size assigned to the integrated graphics capabilities. This option, by default, is set to 256MB.

Boot Display Device

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

Flat Panel Type

This option is used to select the type of flat panel connected to the system. Options include: 640x480 / 800x600 / 1024x768 / 800x480 / 1280x768 / 1280x800 / 1280x600.

Spread Spectrum Clock

By default, this field is set to *Disabled*.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	**Chipset**	Exit

South Bridge Chipset Configuration

USB Function	**[10 USB Ports]**
USB 2.0 Controller	[Enabled]
HAD Controller	[Enabled]
SMBUS Controller	[Enabled]
PCIE Ports Configuration	
PCIE Port 0	[Auto]
PCIE Port 1	[Auto]
PCIE Port 2	[Auto]
PCIE Port 3	[Auto]
PCIE Port 4	[Auto]
PCIE Port 5	[Auto]
PCIE High Priority Port	[Disabled]
PCIE Port 0 IOxAPIC Enable	[Disabled]
PCIE Port 1 IOxAPIC Enable	[Disabled]
PCIE Port 2 IOxAPIC Enable	[Disabled]
PCIE Port 3 IOxAPIC Enable	[Disabled]
PCIE Port 4 IOxAPIC Enable	[Disabled]
PCIE Port 5 IOxAPIC Enable	[Disabled]
Enable Onboard PCI option ROM	[Disabled]

Options

Disabled
2 USB Ports
4 USB Ports
6 USB Ports
8 USB Ports
10 USB Ports

<-	**Select Screen**
↑↓	**Select Item**
+-	**Change Option**
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.

HDA Controller

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

SMBUS Controller

This option is enabled by default.

Clock-Gen Spectrum

This option is disabled by default.

Enable Onboard PCI option ROM

This option is disabled by default.

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	**Exit system setup after saving the changes.**
Discard Changes and Exit	
Discard Changes	**F10 key can be used for this operation**
Load Optimal Defaults	
Load Failsafe Defaults	**<- Select Screen**
	↑↓ Select Item
	Enter Go to Sub Screen
	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.

This page is intentionally left blank.

Drivers Installation

This section describes the installation procedures for software and drivers under the Windows XP, Windows Vista and Windows 7. The software and drivers are included with the motherboard. If you find the items missing, please contact the vendor where you made the purchase. The contents of this section include the following:

Intel Chipset Software Installation Utility..... 38
Intel Pineview Chipset Family Graphics Driver Installation..... 40
Realtek High Definition Codec Audio Driver Installation 42
Intel 82583V LAN Drivers Installation..... 43

IMPORTANT NOTE:
After installing your Windows operating system (Windows XP/ Vista/ 7), 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/7.

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

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

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

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

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

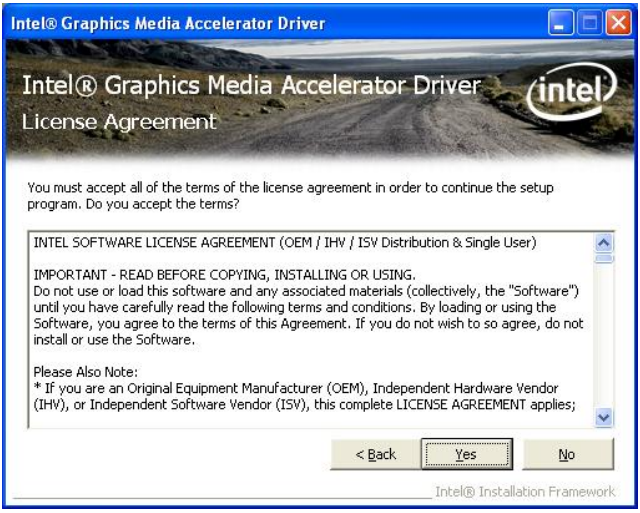
Intel Pineview Chipset Family Graphics Driver Installation

To install the VGA 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) Pineview Chipset Drivers***. Click ***Intel(R) Pineview Chipset Family Graphics Driver***.

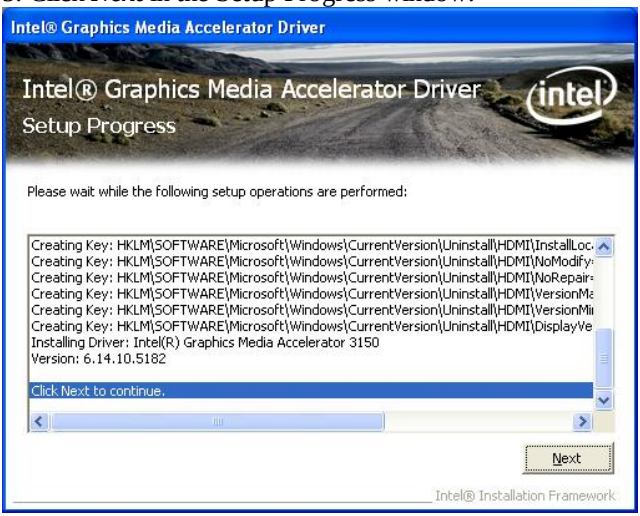
2. When the welcome screen of the Intel(R) Graphics Media Accelerator Driver appears, click ***Next*** to continue.

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



4. Click ***Next*** in the Readme File Information window.

5. Click ***Next*** in the Setup Progress window.



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

Realtek High Definition Codec Audio Driver Installation

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

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

2.When the welcome screen to InstallShield Wizard for ***Realtek High Definition Audio Driver*** appears, click ***Next*** to start the installation.

3.When the InstallShieldWizard has finished performing maintenance operations on Realtek High Definition Codec Audio Audio Driver, click ***Finish*** to restart the computer.

Intel 82583V LAN Drivers Installation

Follow the steps below to install Intel 82583V LAN Drivers.

1. Insert the drivers DVD into the DVD drive. Click ***LAN Card*** and then ***Intel LAN Controller Drivers***. Click ***Intel (R)Gigabit Ethernet Drivers.***

2. On the next screen, click ***Install Drivers*** to start the drivers installation.

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

4. In the License Agreement screen, click ***I accept the terms in license agreement*** and ***Next*** to accept the software license agreement and proceed with the installation process.

5. When the Setup Options appears, click ***Install Drivers only*** and ***Next*** to continue.

6. When the Ready to Install the Program screen appears, click ***Install*** to continue.

7. The Setup process is now complete (InstallShield Wizard Completed). Click ***Finish*** to restart the computer and for changes to take effect.

This page is intentionally left blank.

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
0000h-0CF7h	PCI bus
0000h-0CF7h	Direct memory access controller
0010h-001Fh	Motherboard resources
0020h-0021h	Programmable interrupt controller
0022h-003Fh	Motherboard resources
0040h-0043h	System timer
0044h-005Fh	Motherboard resources
0060h-0060h	Motherboard resources
0061h-0061h	System speaker
0062h-0063h	Motherboard resources
0064h-0064h	Motherboard resources
0065h-006Fh	Motherboard resources
0070h-0071h	System CMOS/real time clock
0072h-007Fh	Motherboard resources
0080h-0080h	Motherboard resources
0081h-0083h	Direct memory access controller
0084h-0086h	Motherboard resources
0087h-0087h	Direct memory access controller
0088h-0088h	Motherboard resources
0089h-008Bh	Direct memory access controller
008Ch-008Eh	Motherboard resources
008Fh-008Fh	Direct memory access controller
0090h-009Fh	Motherboard resources
00A0h-00A1h	Programmable interrupt controller
00A2h-00BFh	Motherboard resources
00C0h-00DFh	Direct memory access controller
00E0h-00EFh	Motherboard resources
00F0h-00FFh	Numeric data processor
01F0h-01F7h	Primary IDE Channel
0274h-0277h	ISAPNP Read Data Port
0279h-0279h	ISAPNP Read Data Port

Address	Device Description
02E8h-02EFh	Communications Port (COM4)
02F8h-02FFh	Communications Port (COM2)
03B0h-03BBh	Intel(R) Graphics Media Accelerator 3150
03C0h-03DFh	Intel(R) Graphics Media Accelerator 3150
03E8h-03EFh	Communications Port (COM3)
03F6h-03F6h	Primary IDE Channel
03F8h-03FFh	Communications Port (COM1)
0400h-041Fh	Intel(R) ICH8 Family SMBus Controller - 283E
04D0h-04D1h	Motherboard resources
0500h-053Fh	Motherboard resources
0800h-087Fh	Motherboard resources
0A00h-0A0Fh	Motherboard resources
0A10h-0A1Fh	Motherboard resources
0A79h-0A79h	ISAPNP Read Data Port
0D00h-FFFFh	PCI bus
B480h-B48Fh	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
B800h-B80Fh	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
B880h-B883h	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
BC00h-BC07h	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
C000h-C003h	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
C080h-C087h	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
C400h-C41Fh	Standard Universal PCI to USB Host Controller
C480h-C49Fh	Intel(R) ICH8 Family USB Universal Host Controller - 2832
C800h-C81Fh	Intel(R) ICH8 Family USB Universal Host Controller - 2831
C880h-C89Fh	Intel(R) ICH8 Family USB Universal Host Controller - 2830

Address	Device Description
CC00h-CC07h	Intel(R) Graphics Media Accelerator 3150
D000h-DFFFh	Intel(R) ICH8 Family PCI Express Root Port 5 - 2847
D800h-D8FFh	Realtek PCIe GBE Family Controller
E000h-EFFFh	Intel(R) ICH8 Family PCI Express Root Port 6 - 2849
E800h-E8FFh	Realtek PCIe GBE Family Controller #2
FFA0h-FFAFh	Intel(R) ICH8M Ultra ATA Storage Controllers - 2850

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
IRQ1	PS/2 Keyboard
IRQ3	Communications Port (COM2)
IRQ4	Communications Port (COM1)
IRQ5	Intel(R) ICH8 Family SMBus Controller - 283E
IRQ8	System CMOS/real time clock
IRQ9	Microsoft ACPI-Compliant System
IRQ10	Communications Port (COM4)
IRQ11	Communications Port (COM3)
IRQ12	PS/2 Mouse
IRQ13	Numeric data processor
IRQ14	Primary IDE Channel
IRQ16	Intel(R) Graphics Media Accelerator 3150
IRQ16	Realtek PCIe GBE Family Controller
IRQ16	Standard Universal PCI to USB Host Controller
IRQ17	Realtek PCIe GBE Family Controller #2
IRQ18	Intel(R) ICH8 Family USB Universal Host Controller - 2832
IRQ18	Intel(R) ICH8M 3 port Serial ATA Storage Controller - 2828
IRQ19	Intel(R) ICH8 Family USB Universal Host Controller - 2831
IRQ21	Microsoft UAA Bus Driver for High Definition Audio
IRQ22	Intel(R) ICH8 Family PCI Express Root Port 1 - 283F
IRQ22	Intel(R) ICH8 Family PCI Express Root Port 5 - 2847
IRQ23	Intel(R) ICH8 Family PCI Express Root Port 6 - 2849
IRQ23	Intel(R) ICH8 Family USB Universal Host Controller - 2830
IRQ23	Intel(R) ICH8 Family USB2 Enhanced Host Controller - 2836

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);
}
//-----------------------------------------------------------------------
```