



integration with integrity

3308400 User's Manual

Full Size PICMG 1.0 SBC Socket LGA 775

Version 1.0

Copyrights

This document is copyrighted and all rights are reserved. It does not allow any non authorization in copied, photocopied, translated or reproduced to any electronic or machine readable form in whole or in part without prior written consent from the manufacturer.

In general, the manufacturer will not be liable for any direct, indirect, special, incidental or consequential damages arising from the use of inability to use the product or documentation, even if advised of the possibility of such damages. The manufacturer keeps the rights in the subject to change the contents of this document without prior notices in order to improve the function design, performance, quality and reliability. The author assumes no responsibility for any errors or omissions, which may appear in this document, nor does it make a commitment to update the information contained herein.

Trademarks

Intel is a registered trademark of Intel Corporation.
Award is a registered trademark of Award Software, Inc.

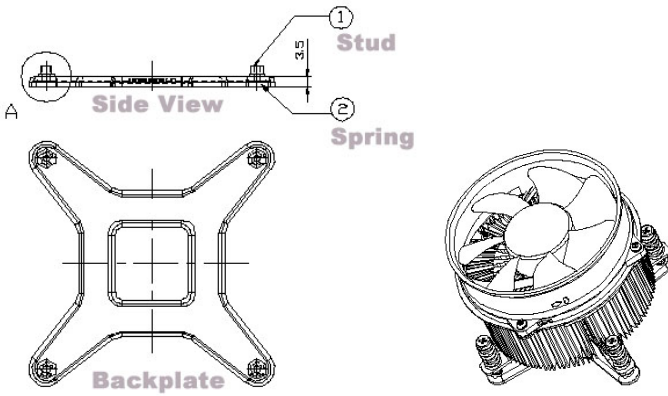
All other trademarks, products and or product's name mentioned herein are mentioned for identification purposes only, and may be trademarks and/or registered trademarks of their respective companies or owners.

Table of Contents

Introduction	1
Product Description.....	1
Checklist.....	2
Specifications	3
Board Dimensions	4
Installations	5
Installing the CPU	6
Installing the Memory	7
Setting the Jumpers	8
Connectors on 3308400.....	12
Appendix	25
A. I/O Port Address Map.....	25
B. Interrupt Request Lines (IRQ)	26
C. Watchdog Timer Configuration.....	27
D. Digital I/O Sample Code	31

ATTENTION:

It is strongly recommended that only heatsinks that have corresponding (metal) backplates 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

Product Description

The 3308400 CPU card is based on the Intel Q965 chipset and supports the Intel® Core™2 Duo processor with 1066/800/533 MHz System Bus. It supports the PCI Express x16 graphics interface for the latest high-performance graphics cards. The PCI Express x1 I/O ports offer up to 3.5X the bandwidth over traditional PCI architecture, delivering faster access to peripheral I/O devices.

The Intel GMA 3000 graphics on board supports Dual Independent Display and delivers richer visual color and picture clarity without the need for additional discrete graphics cards, and the integrated audio support enables premium digital sound and delivers advanced features such as multiple audio.

The CPU card also supports Dual-Channel DDR2 memory in four DIMM sockets and delivers up to 12.8 GB/s of bandwidth and 8 GB memory addressability for faster system responsiveness. Dimensions of the board are 338mm x 126mm.

The main features of the CPU card are:

- Intel® Q965 Express Chipset Based
- Support LGA775 Intel® Core™2 Duo Processors
- Support FSB 1066/800/533MHz
- Support up to 4GB DDRII 800/667/533 memory
- 1 x Mini PCI Express Slot
- Support one 10/100 or two Gigabit LAN on board
- 4 x SATA II, 1 x IDE, 1x Floppy, 6 x USB 2.0, 2 x COM, 1 x Parallel,
- 5.1Ch. Audio, 1 x TMDS
- Dual channel LVDS support 18/24 bit

Checklist

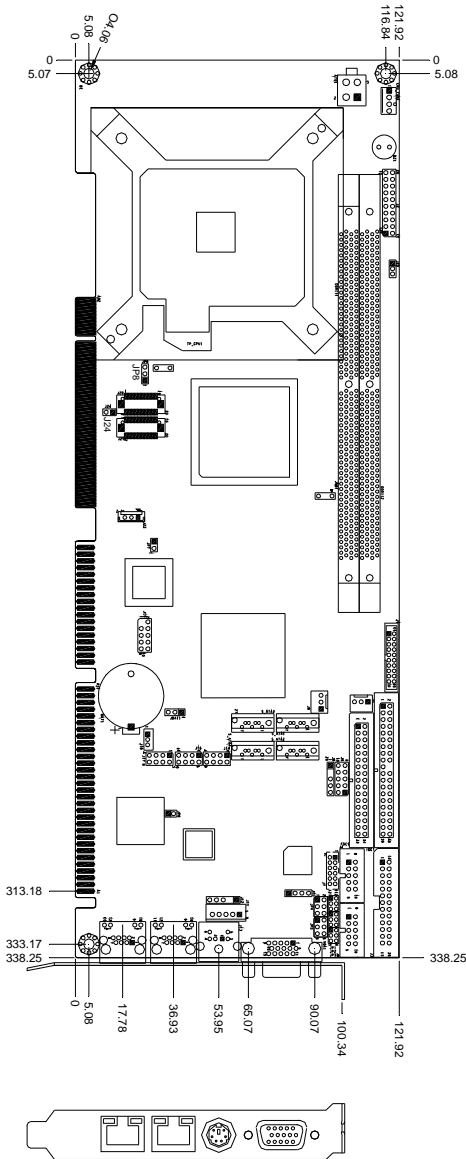
Your 3308400 Core 2 Duo CPU Card package should include the items listed below:

- The 3308400 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
- USB cable with bracket
- 1 CD containing the following:
 - Chipset Drivers
 - Flash Memory Utility
- Optional SATA Power Cord

Specifications

Form Factor	Full Size CPU Card (PICMG1.0)
Processor	Support for Intel Core 2 Duo processor (Conroe core) in LGA775 socket
FSB	533/800/1066 MHz
Chipset	Intel Broadwater (Q965) Chipset consisting of: <ul style="list-style-type: none"> • Intel Q965 Graphics Memory Controller Hub (GMCH) • Intel ICH8/ICH8DO I/O Controller Hub
BIOS	<ul style="list-style-type: none"> • Award BIOS: footprint for both SPI & LPC I/F • Support for ACPI, SMBIOS
Memory	2x 240-pin DDRII 533/667/800 DIMM sockets, supports single channel, max. 4 GB
Video	Intel Q965 integrated graphics subsystem GMA3000
DVI	Chrontel CH7307C x1 for DVI
LVDS	Chrontel CH7308B x1 for 24-bit single/dual channel LVDS
LAN	LAN1: dual footprint support option: <ul style="list-style-type: none"> • Intel 82566DM Nineveh 10/100/1000 (3308400B) • Intel 82562V Ekron-N 10/100 (3308400A) LAN2: Intel 82573L PCI-e gigabit LAN (3308400B)
Audio	Intel ICH8 built-in high definition audio w/ Realtek ALC888 Codec supports 5.1 CH audio (line-out, line-in & mic)
LPC I/O	Winbond W83627EHG: IrDA x1, Parallel x1, COM1 (RS232), COM2 (RS232/422/485), floppy, Hardware monitor (3 thermal inputs, 4 voltage monitor inputs & 2 fan headers)
USB	Intel ICH8 built-in two High Speed USB host controller, supports 6 USB 2.0 ports (support for USB port disable) w/ over-current protection
SATA II	Intel ICH8 built-in SATA II controller (3.0Gb/sec) w/ 4 ports
IDE	JMicron JM368 (PCI-e to PATA) x1 for 1 PATA channel
PCI-to-ISA bridge	ITE IT8888G x1 for high drive ISA bus
Expansion	Mini PCI-express socket x1 for Wireless LAN or other module
Edge Connector	<ul style="list-style-type: none"> • PS/2 connector x1 for PS/2 keyboard & Mouse, DB15 x1 for VGA, RJ45 x2 for LAN 1, 2
On Board Headers / Connectors	<ul style="list-style-type: none"> • Standard SATA (7-pin shrouded vertical) connector x4 • 40 pins, 0.1" pitch, standard box-header x1 for IDE • DF13-20 header x2 for LVDS • DF11-20 header x1 for DVI • 34 pins box-header x1 for floppy • 26 pins box-header x1 for printer • 5x2 pins box-header x2 for COM1-2 • 4 pins pin-header x1 for CPU fan • 3 pins pin-header x1 for system fan • 4x2 pins pin-header x3 for USB1-6
RTC	ICH8 built-in RTC with on-board lithium battery
Watchdog Timer	Yes (256 segments, 0, 1, 2...255 sec/min)
Digital IO	4 in and 4 out
System Voltage	+5V, +3.3V, +12V, -12V & 5VSB
Board Size	338 x 122mm

Board Dimensions



Installations

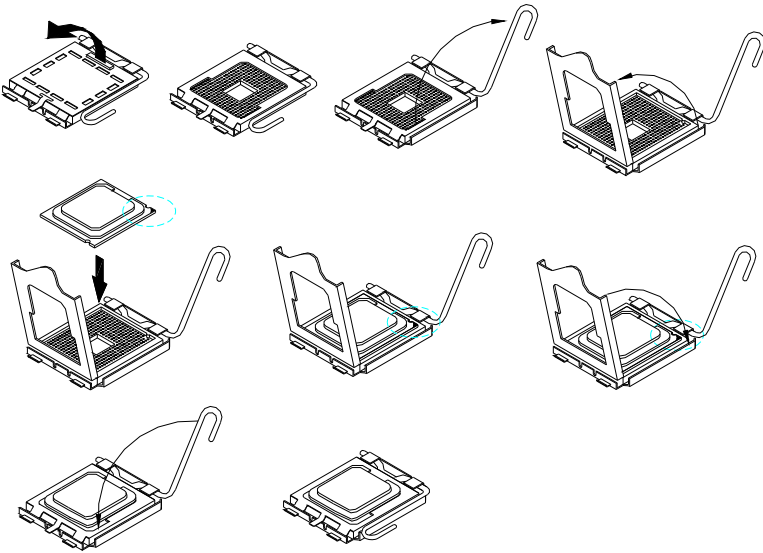
This section provides information on how to use the jumpers and connectors on the 3308400 in order to set up a workable system. The topics covered are:

Installing the CPU.....	6
Installing the Memory.....	7
Setting the Jumpers.....	8
Connectors on 3308400.....	12

Installing the CPU

The 3308400 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.***



Installing the Memory

The 3308400 CPU Card supports four DDR2 memory sockets for a maximum total memory of 4GB in DDR memory type. It supports DDR2 533/667/800.

Basically, the system memory interface has the following features:

- Supports two 64-bit wide DDR data channels

- Available bandwidth up to 6.4GB/s (DDR2 800) for single-channel mode.

- Supports 256Mb, 512Mb, 1Gb DDR2 technologies.

- Supports only x8, x16, DDR2 devices with four banks

- Supports only unbuffered DIMMs

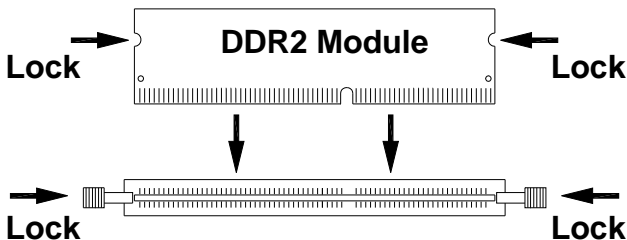
- Supports opportunistic refresh

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

Installing and Removing Memory Modules

To install the DDR2 modules, locate the memory slot on the board and perform the following steps:

1. Hold the DDR2 module so that the key of the DDR2 module align with those on the memory slot.
2. Gently push the DDR2 module in an upright position until the clips of the slot close to hold the DDR2 module in place when the DDR module touches the bottom of the slot.
3. To remove the DDR2 module, press the clips with both hands.

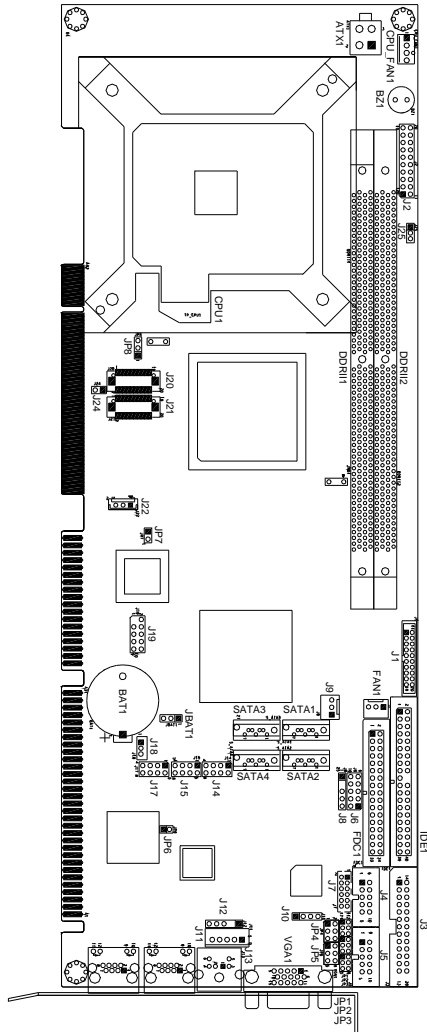


Setting the Jumpers

Jumpers are used on 3308400 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 3308400 and their respective functions.

Jumper Locations on 3308400.....	9
JBAT1: Clear CMOS Contents.....	10
JP1, JP2, JP3: RS232/422/485 (COM2) Selection	10
JP4, JP5: COM1 / COM2 RS232 +5V/+12V Power Setting	10
JP7: Processor Setting.....	11
JP8: LVDS Panel Power Select	11
J25: Power Supply Type Select	11



Jumper Locations on 3308400



Jumpers on 3308400.....	Page
JBAT1: Clear CMOS Contents.....	10
JP1, JP2, JP3: RS232/422/485 (COM2) Selection.....	10
JP4, JP5: COM1 / COM2 RS232 +5V/+12V Power Setting	10
JP7: Processor Setting.....	11
JP8: LVDS Panel Power Select	11
J25: Power Supply Type Select	11

JBAT1: Clear CMOS Contents

Use JBAT1, a 3-pin header, to clear the CMOS contents. *Note that the ATX-power connector should be disconnected from the CPU Card before clearing CMOS.*

JBAT1	Setting	Function
	Pin 1-2 Short/Closed	Normal
	Pin 2-3 Short/Closed	Clear CMOS

JP1, JP2, JP3: RS232/422/485 (COM2) Selection

COM1 is fixed for RS-232 use only.


COM2 is selectable for RS232, RS-422 and RS-485.

ID394: COM3 and COM4 are fixed for RS-232 use only. The following table describes the jumper settings for COM2 selection.





COM2 Function	RS-232	RS-422	RS-485
Jumper Setting (pin closed)	JP1: 1-2	JP1: 3-4	JP1: 5-6
	JP2: 3-5 & 4-6	JP2: 1-3 & 2-4	JP2: 1-3 & 2-4
	JP3: 3-5 & 4-6	JP3: 1-3 & 2-4	JP3: 1-3 & 2-4



JP4, JP5: COM1 / COM2 RS232 +5V/+12V Power Setting

JP4 / JP5	Setting	Function
	Pin 1-2 Short/Closed	+12V
	Pin 3-4 Short/Closed	Normal
	Pin 5-6 Short/Closed	+5V

JP7: Processor Setting



JP7	Setting	Processor Used
 Short	Pin 1-2 Short/Closed	Celeron D
 Open	Pin 1-2 Open	Core 2 Duo, Pentium D, Pentium 4 HT

JP8: LVDS Panel Power Select

JP8	Setting	Panel Voltage
 1 2 3	Pin 1-2 Short/Closed	3.3V (default)
 1 2 3	Pin 2-3 Short/Closed	5V

J25: Power Supply Type Select

Use J25, a 3-pin header, to select between AT and ATX power supply.

J25	Setting	Power Supply Type
 1 2 3	Pin 1-2 Short/Closed	ATX
 1 2 3	Pin 2-3 Short/Closed	AT

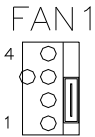
Connectors on 3308400

The connectors on 3308400 allows you to connect external devices such as keyboard, floppy disk drives, hard disk drives, printers, etc. The following table lists the connectors on 3308400 and their respective functions.

Jumper Locations on 3308400.....	13
CPU_FAN1: CPU Fan Power Connector.....	14
PWR_FAN1: SYSTEM Fan Power Connectors.....	14
VGA1: VGA CRT Connector.....	14
FDC1: Floppy Drive Connector	15
IDE1: Primary IDE Connectors	15
S_ATA1, S_ATA2, S_ATA3, S_ATA4: SATA HDD Connectors	16
ATX1: 12V/+12V Power Connector.....	16
J1: TMDS Panel Connector.....	16
J2: System Function Connector	17
J3: Parallel Port Connector	19
J4, J5: COM1 and COM2 Serial Ports Connector	19
J6: Digital I/O Connector (4 in, 4 out).....	20
J7: Audio Connector for 5.1 channel.....	20
J8: IrDA Connector.....	20
J9: External ATX Power Connector	20
J10: CD-In Audio Connector.....	21
J11, J12: External PS/2 Keyboard and Mouse Connector	21
J13: PS/2 Keyboard and Mouse Connector	21
J17, J15, J14: USB0/USB1 Connector	21
J16: Intel 82562V 10/100 or Intel 82566DC GbE RJ-45	22
<i>Note: 10/100 LAN for 3308400A; Gigabit LAN for 3308400B.....</i>	22
J18: Wake On LAN Connector.....	22
J19: SPI Flash Connector (factory use only)	22
J20, J21: LVDS Connectors (1st channel, 2nd channel)	22
J22: Panel Inverter Power Connector	23
J23: Intel PCI Express GbE RJ45 Connector	23
U46: x1 Mini PCI Express Slot	23

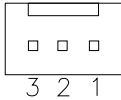
The connectors on 3308400 allows you to connect external devices such as keyboard, floppy disk drives, hard disk drives, printers, etc. The following table lists the connectors on 3308400 and their respective functions.

CPU_FAN1: CPU Fan Power Connector



Pin #	Signal Name
4	Control
3	Sense
2	+12V
1	Ground

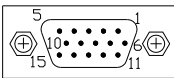
PWR_FAN1: SYSTEM Fan Power Connectors



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

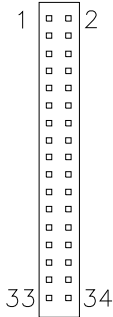
VGA1: VGA CRT Connector

VGA1 is a DB-15 VGA connector located beside the COM1 port. The following table shows the pin-out assignments of this connector.



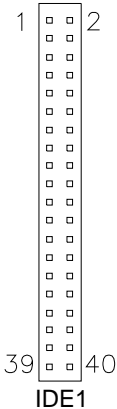
Signal Name	Pin #	Pin #	Signal Name
Red	1	2	Green
Blue	3	4	N.C.
GND	5	6	GND
GND	7	8	GND
VCC	9	10	GND
N.C.	11	12	DDCDATA
HSYNC	13	14	VSYNC
DDCCLK	15		

FDC1: Floppy Drive Connector



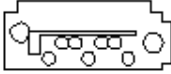
Signal Name	Pin #	Pin #	Signal Name
Ground	1	2	RM/LC
Ground	3	4	No connect
Ground	5	6	No connect
Ground	7	8	Index
Ground	9	10	Motor enable 0
Ground	11	12	Drive select 1
Ground	13	14	Drive select 0
Ground	15	16	Motor enable 1
Ground	17	18	Direction
Ground	19	20	Step
Ground	21	22	Write data
Ground	23	24	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

IDE1: Primary IDE Connectors



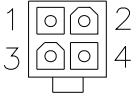
Signal Name	Pin #	Pin #	Signal Name
Reset IDE	1	2	Ground
Host data 7	3	4	Host data 8
Host data 6	5	6	Host data 9
Host data 5	7	8	Host data 10
Host data 4	9	10	Host data 11
Host data 3	11	12	Host data 12
Host data 2	13	14	Host data 13
Host data 1	15	16	Host data 14
Host data 0	17	18	Host data 15
Ground	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

S_ATA1, S_ATA2, S_ATA3, S_ATA4: SATA HDD Connectors



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

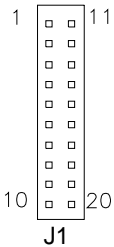
ATX1: 12V/+12V Power Connector



Pin #	Signal Name
1	Ground
2	Ground
3	+12V
4	+12V

J1: TMDS Panel Connector

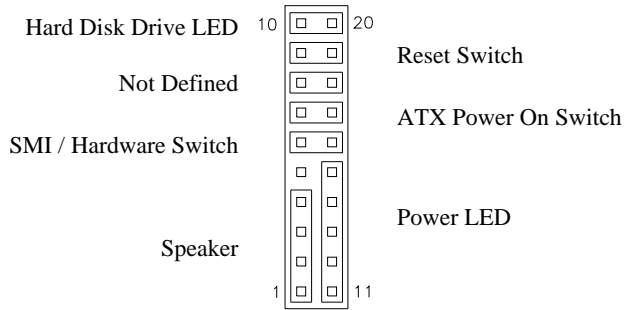
TMDS stands Transition Minimized Differential Signaling.



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

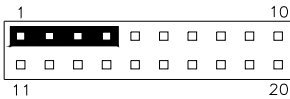
J2: System Function Connector

J2 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status. J1 is a 20-pin header that provides interfaces for the following functions.



Speaker: Pins 1 - 4

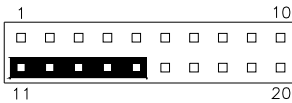
This connector provides an interface to a speaker for audio tone generation. An 8-ohm speaker is recommended.



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

Power LED: Pins 11 - 15

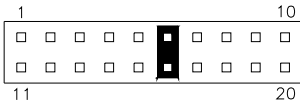
The power LED indicates the status of the main power switch.



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

SMI/Hardware Switch: Pins 6 and 16

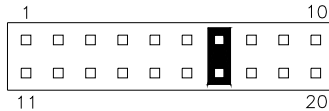
This connector supports the "Green Switch" on the control panel, which, when pressed, will force the system into the power-saving mode immediately.



Pin #	Signal Name
6	SMI
16	Ground

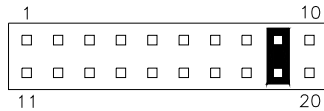
ATX Power ON Switch: Pins 7 and 17

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.



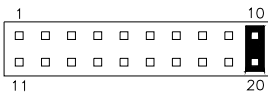
Reset Switch: Pins 9 and 19

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

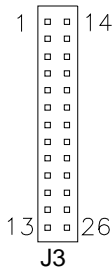
This connector connects to the hard drive activity LED on control panel. This LED will flash when the HDD is being accessed.



Pin #	Signal Name
10	HDD Active
20	5V

J3: Parallel Port Connector

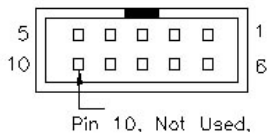
The following table describes the pin out assignments of this connector.



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

J4, J5: COM1 and COM2 Serial Ports Connector

J4 and J5 both 10-pin headers, are the onboard serial port connectors.



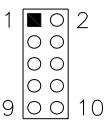
J4
Fixed as
RS-232

J5
Configurable
as RS-232/
RS-422/485
with jumpers
JP1/JP2/JP3

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

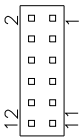
J6: Digital I/O Connector (4 in, 4 out)

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

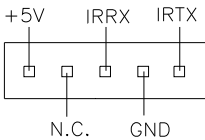
J7: Audio Connector for 5.1 channel



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

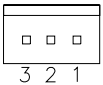
J8: IrDA Connector

J8 is used for an optional IrDA connector for wireless communication.



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

J9: External ATX Power Connector



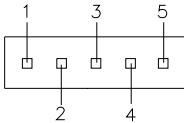
Pin #	Signal Name
1	Ground
2	PS-ON (soft on/off)
3	5VSB (Standby +5V)

J10: CD-In Audio Connector



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

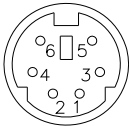
J11, J12: External PS/2 Keyboard and Mouse Connector



Pin #	J11	J12
1	KB clock	Mouse data
2	KB data	N.C.
3	N.C.	Ground
4	Ground	Vcc
5	Vcc	Mouse clock

J13: PS/2 Keyboard and Mouse Connector

J13 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

J17, J15, J14: USB0/USB1 Connector



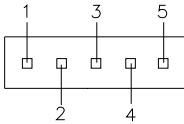
Signal Name	Pin	Pin	Signal Name
Vcc	1	2	Ground
USB0-	3	4	USB1+
USB0+	5	6	USB1-
Ground	7	8	Vcc

J10: CD-In Audio Connector



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

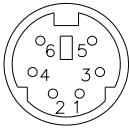
J11, J12: External PS/2 Keyboard and Mouse Connector



Pin #	J11	J12
1	KB clock	Mouse data
2	KB data	N.C.
3	N.C.	Ground
4	Ground	Vcc
5	Vcc	Mouse clock

J13: PS/2 Keyboard and Mouse Connector

J13 uses a Y-cable with dual D-connectors for a PS/2 keyboard and a PS/2 mouse.



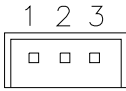
Pin #	Signal Name
1	Keyboard data
2	Mouse data
3	Ground
4	Vcc
5	Keyboard Clock
6	Mouse Clock

J17, J15, J14: USB0/USB1 Connector



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

J22: Panel Inverter Power Connector

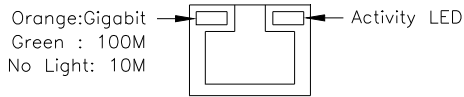


Pin #	Signal Name
1	+12V
2	ENABKL
3	Ground

Remarks: Maximum current is 1A.

J23: Intel PCI Express GbE RJ45 Connector

J23 is the Gigabit LAN RJ45 connector based on the Intel PCI Express GbE controller.



U46: x1 Mini PCI Express Slot

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
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 <stdio.h>
#include <stdlib.h>
#include "W627EHF.H"
//=====
====
int main (int argc, char *argv[]);
void copyright(void);
void EnableWDT(int);
void DisableWDT(void);
//=====
====
int main (int argc, char *argv[])
{
    unsigned char bBuf;
    unsigned char bTime;
    char **endptr;

    copyright();

    if (argc != 2)
    {
        printf(" Parameter incorrect!!\n");
        return 1;
    }

    if (Init_W627EHF() == 0)
    {
        printf(" Winbond 83627HF is not detected, program abort.\n");
        return 1;
    }
    bTime = strtol (argv[1], endptr, 10);
}
```

```

    printf("System will reset after %d seconds\n", bTime);

    EnableWDT(bTime);

    return 0;
}
//=====
void copyright(void)
{
    printf("\n===== Winbond 83627EHF Watch Timer Tester (AUTO DETECT)
=====\\n\\
        \" Usage : W627E_WD reset_time\\n\\
        \" Ex : W627E_WD 3 => reset system after 3 second\\n\\
        \" W627E_WD 0 => disable watch dog timer\\n\\");
}
//=====
void EnableWDT(int interval)
{
    unsigned char bBuf;

    bBuf = Get_W627EHF_Reg( 0x2D);
    bBuf &= (!0x01);
    Set_W627EHF_Reg( 0x2D, bBuf); //Enable WDTO

    Set_W627EHF_LD( 0x08); //switch to logic device 8
    Set_W627EHF_Reg( 0x30, 0x01); //enable timer

    bBuf = Get_W627EHF_Reg( 0xF5);
    bBuf &= (!0x08);
    Set_W627EHF_Reg( 0xF5, bBuf); //count mode is second

    Set_W627EHF_Reg( 0xF6, interval); //set timer
}
//=====
void DisableWDT(void)
{
    Set_W627EHF_LD(0x08); //switch to logic device 8
    Set_W627EHF_Reg(0xF6, 0x00); //clear watchdog timer
    Set_W627EHF_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 "W627EHF.H"
#include <dos.h>
//=====
====
unsigned int W627EHF_BASE;
void Unlock_W627EHF (void);
void Lock_W627EHF (void);
//=====
unsigned int Init_W627EHF(void)
{
    unsigned int result;
    unsigned char ucDid;

    W627EHF_BASE = 0x2E;
    result = W627EHF_BASE;

    ucDid = Get_W627EHF_Reg(0x20);
    if (ucDid == 0x88)
    {   goto Init_Finish;   }

    W627EHF_BASE = 0x4E;
    result = W627EHF_BASE;
    ucDid = Get_W627EHF_Reg(0x20);
    if (ucDid == 0x88)
    {   goto Init_Finish;   }

    W627EHF_BASE = 0x00;
    result = W627EHF_BASE;

Init_Finish:
    return (result);
}
//=====
void Unlock_W627EHF (void)
{
    outportb(W627EHF_INDEX_PORT, W627EHF_UNLOCK);
    outportb(W627EHF_INDEX_PORT, W627EHF_UNLOCK);
}
//=====
====
void Lock_W627EHF (void)
{
    outportb(W627EHF_INDEX_PORT, W627EHF_LOCK);
}
//=====
void Set_W627EHF_LD (unsigned char LD)
{
    Unlock_W627EHF();
    outportb(W627EHF_INDEX_PORT, W627EHF_REG_LD);
    outportb(W627EHF_DATA_PORT, LD);
}

```

```

    Lock_W627EHF();
}
//=====
void Set_W627EHF_Reg( unsigned char REG, unsigned char DATA)
{
    Unlock_W627EHF();
    outportb(W627EHF_INDEX_PORT, REG);
    outportb(W627EHF_DATA_PORT, DATA);
    Lock_W627EHF();
}
//=====
unsigned char Get_W627EHF_Reg(unsigned char REG)
{
    unsigned char Result;
    Unlock_W627EHF();
    outportb(W627EHF_INDEX_PORT, REG);
    Result = inportb(W627EHF_DATA_PORT);
    Lock_W627EHF();
    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 __W627EHF_H
#define __W627EHF_H        1
//=====
#define    W627EHF_INDEX_PORT        (W627EHF_BASE)
#define    W627EHF_DATA_PORT        (W627EHF_BASE+1)
//=====
#define    W627EHF_REG_LD            0x07
//=====
#define W627EHF_UNLOCK            0x87
#define    W627EHF_LOCK            0xAA
//=====
unsigned int Init_W627EHF(void);
void Set_W627EHF_LD( unsigned char);
void Set_W627EHF_Reg( unsigned char, unsigned char);
unsigned char Get_W627EHF_Reg( unsigned char);
//=====
#endif    //__W627EHF_H

```

D. Digital I/O Sample Code

```
Filename: W627hf.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 __W627HF_H
#define __W627HF_H                1
//=====================================================
#define W627HF_INDEX_PORT        (W627HF_BASE)
#define W627HF_DATA_PORT         (W627HF_BASE+1)
//=====================================================
#define W627HF_REG_LD             0x07
//=====================================================
#define W627HF_UNLOCK             0x87
#define W627HF_LOCK               0xAA
//=====================================================
unsigned int Init_W627HF(void);
void Set_W627HF_LD( unsigned char);
void Set_W627HF_Reg( unsigned char, unsigned char);
unsigned char Get_W627HF_Reg( unsigned char);
//=====================================================
#endif    // __W627HF_H
```

Filename: W627hf.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.
//
//=====
unsigned int W627HF_BASE;
void Unlock_W627HF (void);
void Lock_W627HF (void);
//=====
unsigned int Init_W627HF(void)
{
    unsigned int result;    //0=NA, 1=627HF, 2=627THF, 4=627EHF

//    W627HF_BASE = 0x2E;
//    W627HF_BASE = 0x4E;
    result = Get_W627HF_Reg(0x20);
    if (result == 0x52)
    {
        result = 1;
        goto Init_Finish;
    }
    else if (result == 0x82)
    {
        result = 2;
        goto Init_Finish;
    }
    else if (result == 0x88)
    {
        result = 4;
        goto Init_Finish;
    }

//    W627HF_BASE = 0x4E;
//    W627HF_BASE = 0x2E;

    result = Get_W627HF_Reg(0x20);
    if (result == 0x52)
    {
        result = 1;
        goto Init_Finish;
    }
    else if (result == 0x82)
    {
        result = 2;
        goto Init_Finish;
    }
    else if (result == 0x88)
    {
        result = 4;
        goto Init_Finish;
    }
}
```

```

        W627HF_BASE = 0x00;
        result = 0;

Init_Finish:
        return (result);
    }
//=====
void Unlock_W627HF (void)
{
    outportb(W627HF_INDEX_PORT, W627HF_UNLOCK);
    outportb(W627HF_INDEX_PORT, W627HF_UNLOCK);
}
//=====
void Lock_W627HF (void)
{
    outportb(W627HF_INDEX_PORT, W627HF_LOCK);
}
//=====
void Set_W627HF_LD( unsigned char LD)
{
    Unlock_W627HF();
    outportb(W627HF_INDEX_PORT, W627HF_REG_LD);
    outportb(W627HF_DATA_PORT, LD);
    Lock_W627HF();
}
//=====
void Set_W627HF_Reg( unsigned char REG, unsigned char DATA)
{
    Unlock_W627HF();
    outportb(W627HF_INDEX_PORT, REG);
    outportb(W627HF_DATA_PORT, DATA);
    Lock_W627HF();
}
//=====
unsigned char Get_W627HF_Reg(unsigned char REG)
{
    unsigned char Result;
    Unlock_W627HF();
    outportb(W627HF_INDEX_PORT, REG);
    Result = inportb(W627HF_DATA_PORT);
    Lock_W627HF();
    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 "W627HF.H"
//-----
void ClrKbBuf(void);
int main (void);
unsigned char GetDI(unsigned char);
void SetDo(unsigned char, unsigned char);
//-----
int main (void)
{
    unsigned char ucDO = 0;                //data for digital output
    unsigned char ucDI;                    //data for digital input
    unsigned char ucBuf;
    char SIO;

    SIO = Init_W627HF();
    if (SIO == 0)
    {
        printf("Can not detect Winbond 83627HF/83627THF/83627EHF, program abort.\n");
        return(1);
    }
    switch (SIO)
    {
        //-----
        case 1:
            printf("Winbond 83627HF is detected.\n");
            break;
        //-----
        case 2:
            printf("Winbond 83627THF is detected.\n");
            break;
        //-----
        case 4:
            printf("Winbond 83627EHF is detected.\n");
            break;
        //-----
    }

    //bit 0..3 = input signal
    //bit 4..7 = output signal

    ucDI = GetDI(0x0F);                    //get current DI status
    SetDo(ucDO, 0xF0);                    //set current DO status
    return 0;
}
//-----
unsigned char GetDI(unsigned char Mask)
{
```

```
    unsigned char result;

    Set_W627HF_LD(0x07);
    Set_W627HF_Reg(0xF0, Mask);
    result = Get_W627HF_Reg(0xF1) & Mask;
    return (result);
}
//-----
void SetDo(unsigned char NewData, unsigned char Mask)
{
    Set_W627HF_LD(0x07);
    Set_W627HF_Reg(0xF0, ~Mask);
    Set_W627HF_Reg(0xF1, NewData & Mask);
}
//-----
void ClrKbBuf(void)
{
    while(kbhit())
    {   getch();   }
}
//-----
```

Any advice or comments about our products and service, or anything we can help you with please don't hesitate to contact with us. We will do our best to support your products, projects and business.



Address: Global American, Inc.
17 Hampshire Drive
Hudson, NH 03051

Telephone: Toll Free U.S. Only (800) 833-8999
(603) 886-3900

FAX: (603) 886-4545

Website: <http://www.globalamericaninc.com>

Support: Technical Support at Global American