integration with integrity

2808040 User's Manual

Mini-ITX Motherboard with Socket LGA 775

Version 1.0

Table of Contents

This page is intentionally left blank.

Introduction

Checklist

Your 2808040 Core 2 Duo motherboard package should include the items listed below:

- The 2808040 motherboard
- This User's manual
- 1 x I/O shield
- 1 x IDE cable
- 1 x SATA cable
- 1 CD containing the following:
  - Chipset Drivers
  - Flash Memory Utility

Product Description

The 2808040 Mini-ITX motherboard is designed for either the Intel® Core™2 Duo or Core™2 Quad processors of up to 1333MHz FSB. It is based on the Intel's Q35 Express chipset and it comes with two single-channel DDR2 memory slots and 4GB memory capacity for faster system responsiveness and support of 64-bit computing. The new GAI motherboards are aimed for high performance PCs in the digital, communications and industrial sector.

On board is one PCI Express x16 slot that offers up to 3.5X the bandwidth over traditional PCI architecture to support the latest high-performance graphics cards. Dual independent display comes to life with the onboard Intel® Q35 integrated graphics for CRT and an optional SDVO card supporting either an LVDS or DVI display interface. LAN functionality is supported with a 10/100 Ethernet controller or with two Gigabit Ethernet controllers. 2808040 is expandable, with the use of an adaptor card, 1008097, to support 2 or 4 serial ports, or 1008098 to support TPM 1.2 security function. Other useful features on the board include two SATA II ports, one eSATA port, eight USB 2.0 interface, watchdog timer, digital I/O and two serial ports. Board dimensions are 170mm by 170mm. (Note: CPU power consumption – under 95watt recommended.)

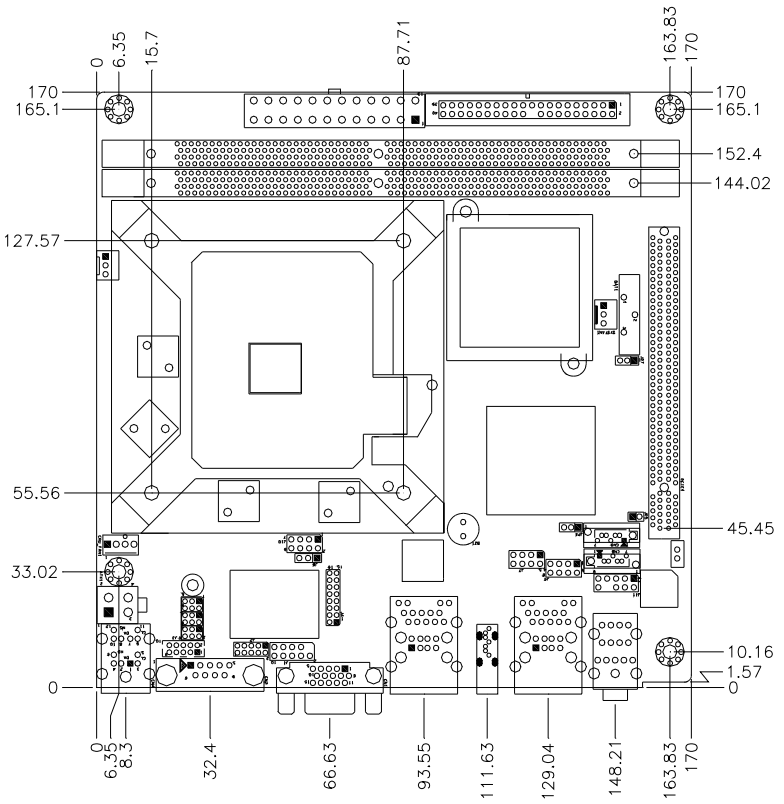
2808040 FEATURES

- Intel® Q35 Express Chipset Based
- Support LGA775 Intel® Core™2 Duo/Quad CPU
- Support up to 1333MHz FSB
- Support up to 4GB DDRII 800/667 memory
- 1 x PCI Express (x16)
- Support one 10/100 or two Gigabit LAN on board
- 2x SATA II, 1x eSATA, 1x IDE, 8x USB 2.0, 2x COM, 7.1Ch.HD Audio

Specifications

Form Factor	Mini ITX (for performance desktop market)
Processor	Socket LGA775, Supports the Intel Core 2 Duo and Intel Core2 Quad processors, and Intel Celeron 400 (Conroe-L) Sequence processor.
FSB	800/1066/1333 MHz
Chipset	Intel Q35 Chipset: • Intel Q35 Graphic Memory Controller Hub (GMCH) • Intel ICH9/ICH9R I/O Controller Hub
BIOS	• Award BIOS: footprint from SPI, supports ACPI, SMBIOS
Memory	• **2** x 240-pin DDRII 667/800 DIMM sockets, support one channels, • Supports max. **4 GB** system memory
Video	Intel Q35 integrated graphic subsystem (GMA3100), dual independent display available through on-board VGA and PCI-e x16 expansion adapter (DVI or LVDS) or VGA card
LAN	LAN1: dual Footprint support option: •Intel 82566DM Nineveh 10/100/1000 LAN •Intel 82562V Ekron-N 10/100 LAN2: Marvell 88E8053 PCI-express Gigabit LAN controller x1
USB	Intel ICH9/ICH9R built-in USB 2.0 host controller, supports 8 ports: •4 ports in the rear I/O region •4 ports with on-board headers
SATA II	Intel ICH9 built-in SATA II controller (3.0Gb/sec) w/ 2 ports. ICh9R Built-in raid 0,1 or AHCI (for eSATA)
IDE	Jmicron JM368 (PCI-e to PATA) x1 for 1 PATA channel
Audio	Intel ICH9/ICH9R built-in high definition audio w/ Realtek ALC888 Codec
LPC I/O	Winbond W83627EHG: COM1 (RS232), COM2 (RS232/422/485) & Hardware monitor
Hardware Monitor	• Two fan connectors with tachometer support • CPU fan connector supports 4-wire fan with PWM control • Supports three thermal diodes (CPU die + 2 on-board) • Voltage monitoring for VCC (processor), 3.3V, 5V, and 12V
Edge Connectors	• Mini-DIN x 1 for PS/2 KB & MS • Esata connectorx1(for ICH9R) • DB9 + DB15 stack connector x 1 for COM1 & VGA • RJ45 + dual USB stack connector x2 for LAN1~2 and USB1~4 • Triple (3x1) phone jack connector x1 for High-Definition Audio
On Board Headers / Connectors	• Standard SATA (7-pin shrouded vertical) connector x2 • 4x2 pins pin-header x2 for USB 5-6,7-8. • 5x2 pins x1 for COM2 (RS232/422/485) • 5x2 pins pin-header x1 for Digital I/O • 5X2 pins pin-headerx1 for audio front. • 40 pins box-header x1 for IDE • 4 pins pin--header x1 for CPU fan & system fan • 3 pins pin--header x2 for system fan
Expansion	PCI-express (x16) slot x1 8x2 pins pin header x1 for adaptor card: - 1008097 (2 or 4 serial ports) - 1008098 (TPM function)
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
Digital IO	4 in and 4 Out
Other	LAN Wakeup
Power Connector	24 pins ATX main power + 4 pins 12V
System Voltage	+5V, +3.3V, +12V, -12V & 5VSB
Board Size	170 x 170mm

Board Dimensions



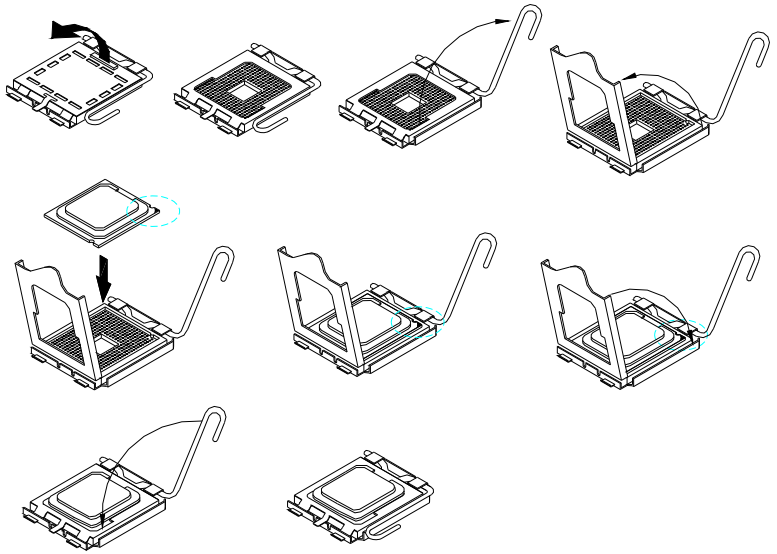
Installations

This section provides information on how to use the jumpers and connectors on the 2808040 in order to set up a workable system. The topics covered are:

Installing the CPU

The 2808040 motherboard supports an LGA 775 processor socket for Intel® Core 2 Duo processors.

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



ATX Power Installation

The system power is provided to the motherboard with the ATX2 and ATX1 power connectors. ATX2 is a 24-pin power connector and ATX1 is a 4-pin 12V power connector.

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

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

Installing the Memory

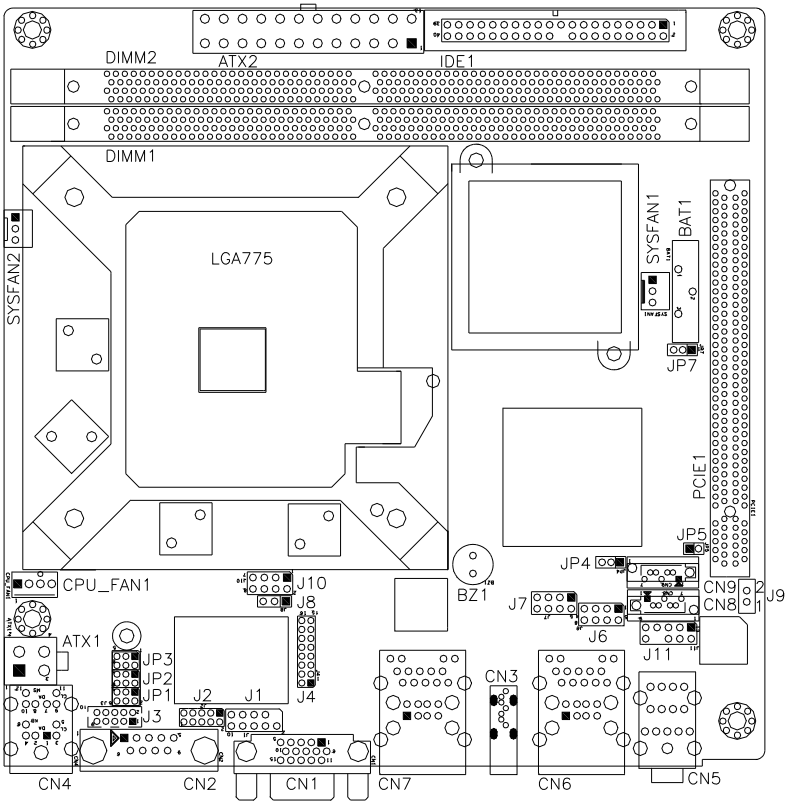
The 2808040 motherboard supports four DDR2 memory sockets for a maximum total memory of 4GB in DDR memory type. It supports DDR2 667/800MHz.

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 two-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)

Setting the Jumpers

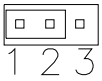
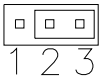
Jumpers are used on the motherboard are used to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors and their respective functions.



Jumper Locations on 2808040

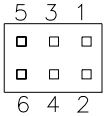
JP7: Clear CMOS Contents

Use JP7, a 3-pin header, to clear the CMOS contents. *Note that the ATX-power connector should be disconnected from the motherboard before clearing CMOS.*

JP7	Setting	Function
	Pin 1-2 Short/Closed	Normal
	Pin 2-3 Short/Closed	Clear CMOS

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

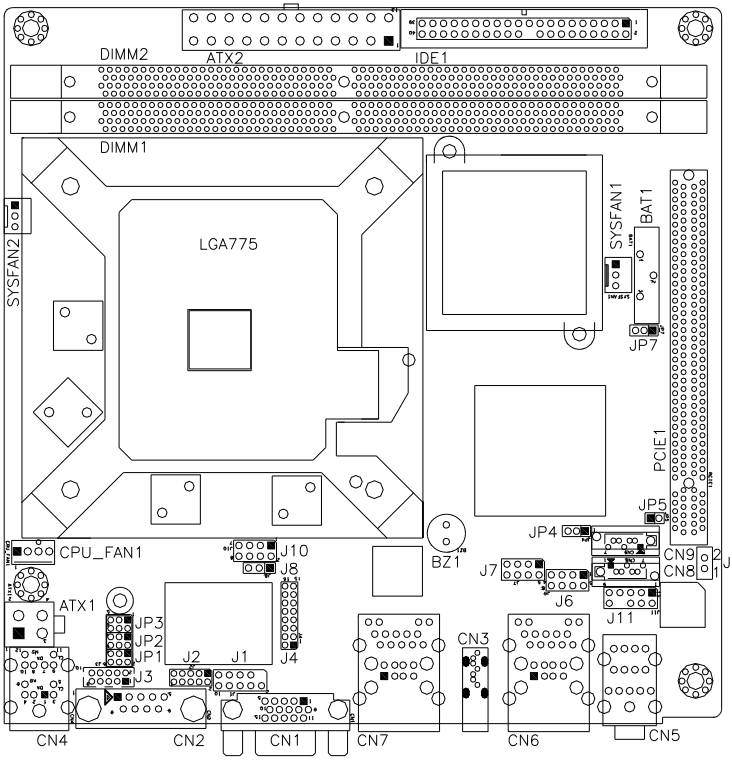
COM1 is fixed for RS-232 use only.
COM2 is selectable for RS232, RS-422 and RS-485.
1008097: COM3 and COM4 are fixed for RS-232 use only. The following table describes the jumper settings for COM2 selection.



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

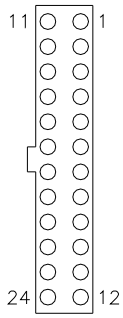
Connectors on 2808040

The connectors on 2808040 allows you to connect external devices such as keyboard, floppy disk drives, hard disk drives, printers, etc. The following table lists the connectors on 2808040 and their respective functions.



Connector Locations on 2808040

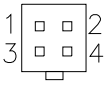
ATX2: 24-pin ATX Power Connector



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

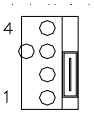
ATX1: ATX 12V Power Connector

This connector supplies the CPU operation voltage



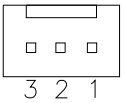
Pin #	Signal Name
1	Ground
2	Ground
3	+12V
4	+12V

CPU_FAN1: CPU Fan Power Connector

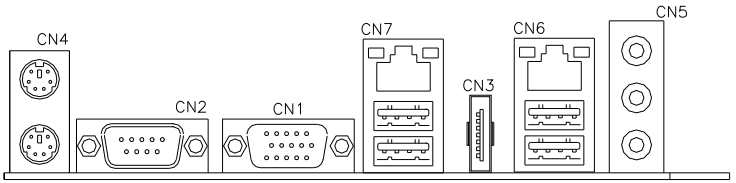


Pin #	Signal Name
1	Ground
2	+12V
3	Sense
4	Control

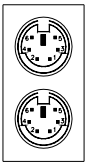
SYS FAN1, 2: System Fan Power Connectors



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



CN4: PS/2 Keyboard and PS/2 Mouse Connectors



Mouse (top)
Keyboard (bottom)

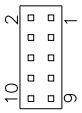
Keyboard Signal	Pin #	Mouse Signal
Keyboard data	1	Mouse data
N.C.	2	N.C.
GND	3	GND
5V	4	5V
Keyboard clock	5	Mouse clock
N.C.	6	N.C.

CN2, J3: COM1/2 Serial Ports

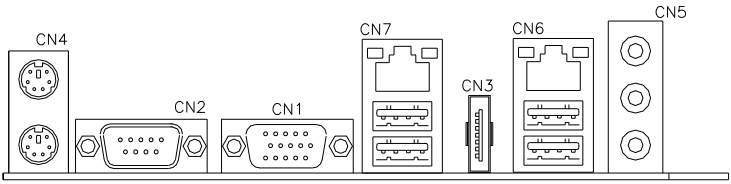
CN2 (COM1) is a DB-9 connector, while J3 is a COM pin-header connector.

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

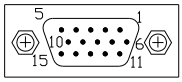
J3: COM2 is jumper selectable for RS-232, RS-422 and RS-485.



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



CN1: VGA CRT Connector



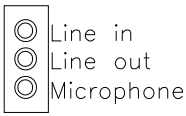
Signal Name	Pin #	Pin #	Signal Name
Red	1	2	Green
Blue	3	4	N.C.
GND	5	6	GND
GND	7	8	GND
VCC	9	10	GND
N.C.	11	12	DDCDATA
HSYNC	13	14	VSYNC
DDCCLK	15		

CN6: Marvell 88E8053 PCI-express Gigabit LAN and USB2/3 Connector

CN7: Intel 82562V 10/100 or Intel 82566DM GbE RJ-45 and USB0/1 Connector

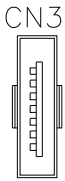
Note: 10/100 LAN for 2808040A; Dual Gigabit LAN for 2808040B, C

CN5: Audio Connector



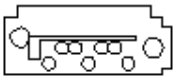
CN4 is a 3-jack audio connector

CN3: eSATA HDD Connector



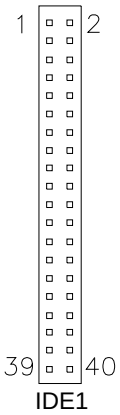
Pin #	Signal Name
1	Ground
2	TX+
3	TX-
4	Ground
5	RX-
6	RX+
7	Ground

CN9, CN8: SATA HDD Connectors



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

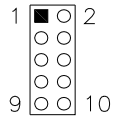
IDE1: Primary IDE Connectors



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

J2: 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.

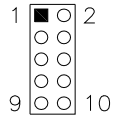


Signal Name	Pin #	Pin #	Signal Name
Ground	1	2	+5V
Out3	3	4	Out1
Out2	5	6	Out0
IN3	7	8	IN1
IN2	9	10	IN0

J4: For LPC I/F Adaptor Card

Supports 1008097 with Fintek F81216, 2 or 4 serial ports

J5: Audio Front Header



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

J6, J7: USB4/5, USB6/7 Connectors

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

J8: Power LED

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

Pin #	Signal Name
1	Power LED
2	No connect
3	Ground

J9: SPDIF Out Connector

Pin #	Signal Name
1	SPDIF out
2	Ground

J10: System Function Connector

ATX Power ON Switch: Pins 1 and 2
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.

Hard Disk Drive LED Connector: Pins 3 and 4
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
4	HDD Active
3	5V

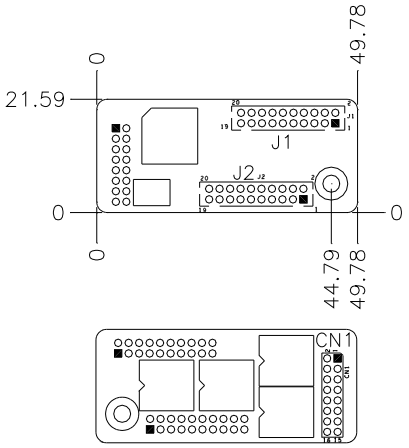
Reset Switch: Pins 5 and 6
The reset switch allows the user to reset the system without turning the main power switch off and then on again. Orientation is not required when making a connection to this header.

+5V and 5VSB Signals: Pins 7 and 8

Pin #	Signal Name
7	+5V
8	+5VSB

PCIE_1: x16 PCI Express Slot

1008097 LPC Serial Ports Adapter (option)



J1 J2: COM3/4/5/6 Serial Ports

J1 - COM3/4 pin-header connector.

PIN1~PIN10 COM3

PIN11~PIN20 COM4

J2 - COM5/6 pin-header connector.

PIN1~PIN10 COM5

PIN11~PIN20 COM6

Signal Name	Pin #	Pin #	Signal Name
DCD, Data carrier detect	1	2	DSR, Data set ready
RXD, Receive data	3	4	RTS, Request to send
TXD, Transmit data	5	6	CTS, Clear to send
DTR, Data terminal ready	7	8	RI, Ring indicator
GND, ground	9	10	Not Used
DCD, Data carrier detect	11	12	DSR, Data set ready
RXD, Receive data	13	14	RTS, Request to send
TXD, Transmit data	15	16	CTS, Clear to send
DTR, Data terminal ready	17	18	RI, Ring indicator
GND, ground	19	20	Not Used

CN1: LPC I/F connectors

Package list:

COM port cable x 2

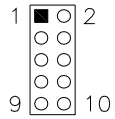
Nylon Nut x 1

Screw x 2

Washer x2

J2: 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.

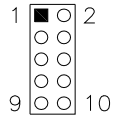


Signal Name	Pin #	Pin #	Signal Name
Ground	1	2	+5V
Out3	3	4	Out1
Out2	5	6	Out0
IN3	7	8	IN1
IN2	9	10	IN0

J4: For LPC I/F Adaptor Card

Supports 1008097 with Fintek F81216, 2 or 4 serial ports

J5: Audio Front Header



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

J6, J7: USB4/5, USB6/7 Connectors

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

J8: Power LED

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

Pin #	Signal Name
1	Power LED
2	No connect
3	Ground

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
```

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 (int argc, char *argv[]);
//====================================================================
int main (int argc, char *argv[])
{
	unsigned char ucDO = 0;					//data for digital output
	unsigned char ucDI;					//data for digital input
	unsigned char ucBuf;

	Set_W627HF_LD( 0x07);					//switch to logic device 7

	Set_W627HF_Reg(0xF1, 0x00);				//clear
	ucDI = Get_W627HF_Reg(0xF1) & 0x0F;

	ClrKbBuf();
	while(1)
	{
		ucDO++;
		Set_W627HF_Reg(0xF1, ((ucDO & 0x0F) << 4));
		ucBuf = Get_W627HF_Reg(0xF1) & 0x0F;
		if (ucBuf != ucDI)
		{
			ucDI = ucBuf;
			printf("Digital I/O Input Changed. Current Data is 0x%X\n",ucDI);
		}

		if (kbhit())
		{
			getch();
			break;
		}
		delay(500);
	}
	return 0;
}
//====================================================================
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.

Global AMERICAN

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