integration with integrity

User's Manual

3.5" Embedded Board 3308370

Version 1.0

Copyrights

Trademarks

Introduction

Product Description

3308370 3.5” disk-size board incorporat es the Mobile Intel® Poulsbo Chipset for Embedded Computing, consisting of the Intel® single-chip system controller hub (SCH), an optimized integrated graphics solution with a 533MHz and 400MHz front-side bus. Dimensions of the board are 102mm x 147mm.

The integrated powerful 3D graphics engine, based on Intel® Graphics Media Accelerator) architecture 500, operates at core speeds of up to 200 MHz. It features a low-power design, With DDR2 533/400MHz one SO-DIMM socket on board, the board supports up to 1GB of DDR2 system memory.

Intel® Graphics supports a unique intelligent memory management scheme called Dynamic Video Memory Technology (DVMT). DVMT handles diverse applications by providing the availability of system memory for general computer usage, while supplying additional graphics memory when a 3D-intensive application requests it. The Intel graphics architecture also takes advantage of the high-performance Intel processor. Intel graphics supports Dual Independent Display technology.

The main features of the board are:

- Supports Intel® Atom Z530 (1.6GHz), Z510 (1.1GHz)
- Supports up to 533MHz FSB
- One DDR2 SDRAM SO-DIMM, Max. 1GB memory
- Onboard Two Intel Gigabit LAN
- Intel® VGA for CRT (ID398) / LVDS
- 1x SATA, 8x USB 2.0, 4x COM, Watchdog timer
- 1x PCI-104 Socket

Checklist

Your package should include the items listed below.

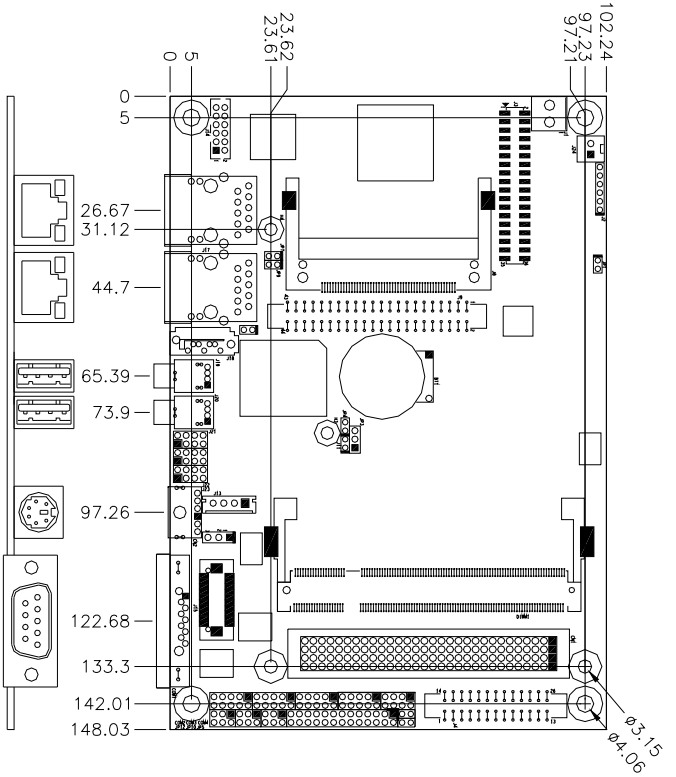
- 3308370
- This User's Manual
- 1 CD containing chipset drivers and flash memory utility
- Options:
  - Cable kit
  - Heatsink

Reminder:
When using the SATA hard disk to install the operating system, you need to install the SATA driver first. The path for the driver files, assuming D: is the CD/DVD ROM drive, would be D:\SATA\Jmicron\Windrv\Floppy32.

3308370 Specifications

Form Factor	3.5" Disk Size SBC
CPU Type	Intel Atom (Silverthorne) CPUs 400/533 MHz, - Ultra Low Voltage - 512KB On-die L2 Cache - Hyper- Threading Technology support
System Speed	1.1GHz / 1.6GHz
CPU FSB	400M/533MHz
Cache	512K/1M L2 cache
Green /APM	APM1.2
BIOS	Award BIOS: supports ACPI Function
Chipset	Intel Poulsbo SCH Chipset, 1249-pin BGA, 22X22 mm
Memory	1 x 200-pin DDRII, 400/533-MT/s SO-DIMM, Max. capacity 1GB (Single channel)
VGA	- Through SDVO port (ID398)
LAN	Poulsbo SCH offers two PCI express channel for expansion. Use one channel Thru PCIE switch (PLX pex8505) for two Gigabit LAN (Intel 82574Lx2)
USB	Poulsbo SCH built-in USB2. 0 host controller with 8 ports,
Audio	Poulsbo SCH built-in Audio controller ALC 888 5.1-Channel (Line-in, Line-out & MIC)
PATA-IDE	Poulsbo SCH built-in one channel Ultra DMA 33/66/100, for IDE (44-pin box-header x 1) & CF (TYPEII)
SATA-IDE	Thru PCIE switch (PLX pex8505) for SATAII x1 (JMICRON JMB362)
LPC I/O	Winbond 83627EHF: - COM1 (RS232), COM2 (RS232/422/485), - LPT1 - Hardware monitor (3 thermal inputs, 6 voltage monitor inputs) - KB/Mouse Connector (KB 1ST Priority)
2nd LPC I/O	Fintek F81216DG COM3 & COM4 (RS232)
RTC/CMOS	Poulsbo SCH built-in with on board Lithium Battery
Edge Connector	PS/2 connector x1 for PS/2 KB/Mouse DB15 x1 for VGA (ID398 VGA daughter board) DB9 x1 for COM1 RJ45 x2 for Gigabit LAN Dual USB stack connector x1 for USB1/2
Onboard Header / Connectors	DF13 Socket x 1 for LVDS 8 pins header x 3 for 6 USB ports 44-pin box header x1 for IDE 36-pin female connector x1 for SDVO port 12-pin header x1 for audio 10-pin header x3 for COM2, COM3, COM4 26-pin header for SD/SDIO/MMC
Digital I/O	4 in / 4 out
Expansion Slots	PCI104 , BY PCIE to PCI bridge (PLX pex8112)
Watchdog Timer	Yes (256 segments, 0, 1, 2…255. sec/min)
Power Connector	DC power jack x1 for +12V ~+19V DC-in
Board Size	102 x 147mm

Board Dimensions



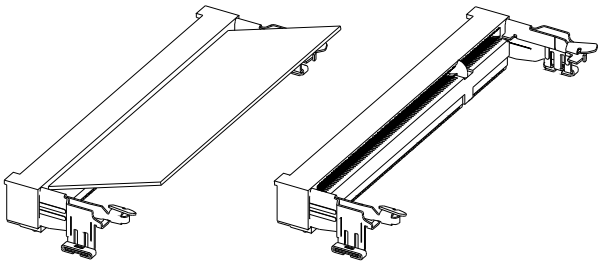
Installing the Memory

This board supports a DDR2 memory socket for a maximum total memory of 1GB in DDR2 533/400 memory type.

Installing and Removing Memory Modules

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

1. Hold the DDR2 module so that the keys of the DDR2 module align with those on the memory slot.
2. Gently push the DDR2 module in an angle as shown in the picture below until the clips of the sockets lock to hold the DDR2 module in place when the DDR2 module touches the bottom of the socket.
3. To remove the DDR2 module, press the clips with both hands.

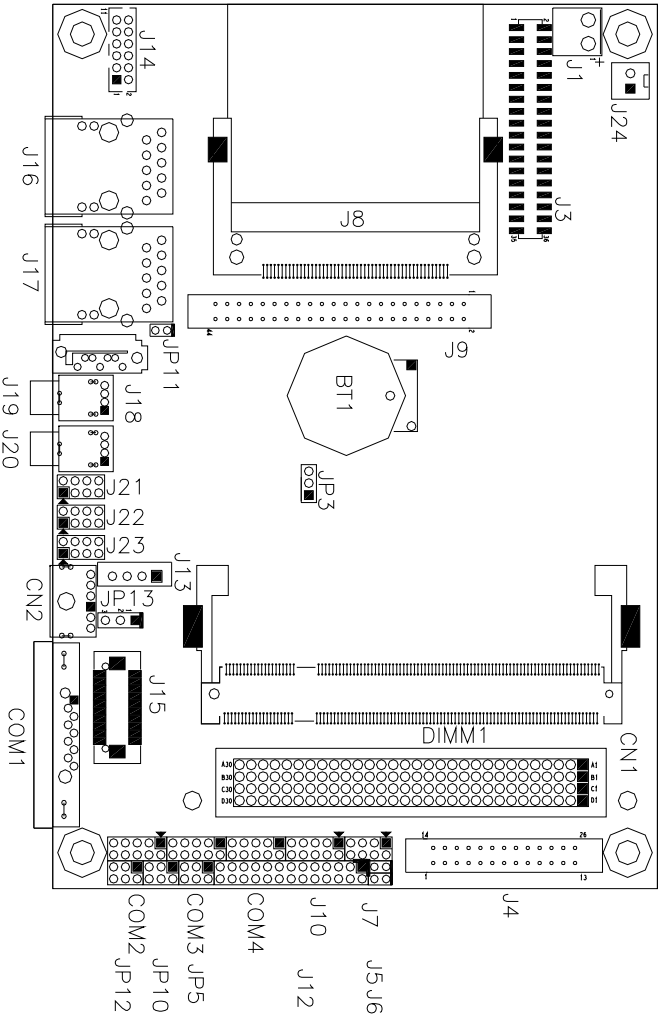


Setting the Jumpers

Jumpers are used on this 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.

Jumper Locations on 3308370
JP3: Clear CMOS Setting
JP11: CompactFlash Slave/Master Selection
JP5, JP10, JP12: RS232/422/485 (COM2) Selection
J5, J6: SD/IO Power Selection
JP13: LCD Panel Power Selection

Jumper Locations



JP3: Clear CMOS Setting

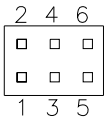
JP3	Setting
	Normal
	Clear CMOS

JP11: CompactFlash Slave/Master Selection

JP11	CF Setting
	Master
	Slave

JP5, JP10, JP12: RS232/422/485 (COM2) Selection

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



COM2 Function	RS-232	RS-422	RS-485
Jumper Setting (pin closed)	JP5: 1-2 JP10: 3-5 & 4-6 JP12: 3-5 & 4-6	JP5: 3-4 JP10: 1-3 & 2-4 JP12: 1-3 & 2-4	JP5: 5-6 JP10: 1-3 & 2-4 JP12: 1-3 & 2-4

J5, J6: SD/IO Power Selection

J5	SLOT2 Power
	Enable
	By MANUAL

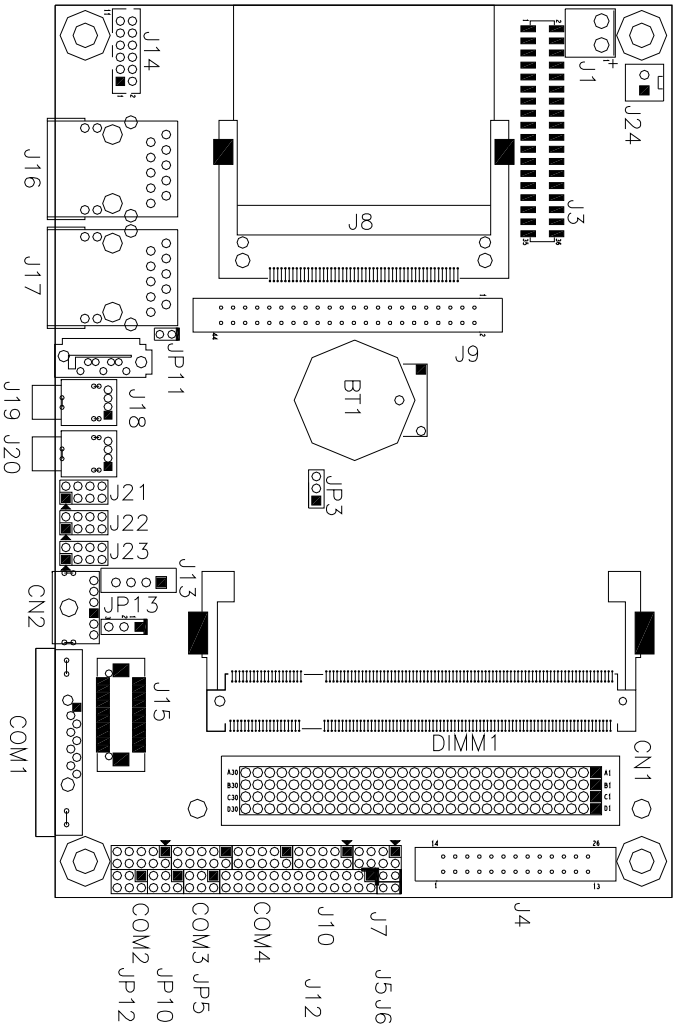
J6	SLOT1 Power
	Enable
	By MANUAL

JP13: LCD Panel Power Selection

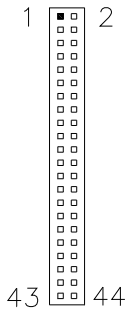
JP13	LCD Panel Power
	3.3V
	5V

Connectors

Connector Locations



IDE1: IDE Connector



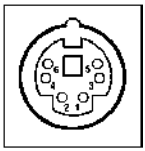
Signal Name	Pin #	Pin #	Signal Name
Reset IDE	1	2	Ground
Host data 7	3	4	Host data 8
Host data 6	5	6	Host data 9
Host data 5	7	8	Host data 10
Host data 4	9	10	Host data 11
Host data 3	11	12	Host data 12
Host data 2	13	14	Host data 13
Host data 1	15	16	Host data 14
Host data 0	17	18	Host data 15
Ground	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	Cable Detect
Address 0	35	36	Address 2
Chip select 0	37	38	Chip select 1
Activity	39	40	Ground
+5V	41	42	+5V
Ground	43	44	NC

J1: Board Input Power Connector

Pin #	Signal Name
1	+12V~+19V
2	GND

CN1: PCI-104 Connector

CN2: PS/2 Keyboard and PS/2 Mouse Connectors

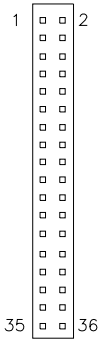


CN3

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

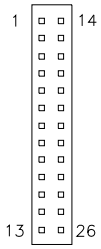
J2: CPLD Connector (factory use only)

J3: SDVO Connector



Signal Name	Pin #	Pin #	Signal Name
+12V	1	2	+3.3V
+12V	3	4	+3.3V
CTRL-DATA	5	6	+3.3V
GND	7	8	GND
CTRL-CLK	9	10	+2.5V
GND	11	12	+2.5V
RED+	13	14	GND
RED-	15	16	RESET
GND	17	18	GND
GREEN+	19	20	TV-CLK+
GREEN-	21	22	TV-CLK-
GND	23	24	GND
BLUE+	25	26	INIT+
BLUE-	27	28	INIT-
GND	29	30	GND
CLK+	31	32	STALL+
CLK-	33	34	STALL-
GND	35	36	GND

J4: SD/IO Connector

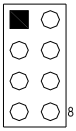


Signal Name	Pin #	Pin #	Signal Name
SLOT2-PWR(3.3V)	1	14	SLOT1-PWR(3.3V)
SLOT2-DATA0	2	15	SLOT1-DATA0
SLOT2-DATA1	3	16	SLOT1-DATA1
SLOT2-DATA2	4	17	SLOT1-DATA2
SLOT2-DATA3	5	18	SLOT1-DATA3
SLOT2-DATA4	6	19	SLOT1-CMD
SLOT2-DATA5	7	20	SLOT1-CLK
SLOT2-DATA6	8	21	SLOT1-CD#
SLOT2-DATA7	9	22	SLOT1-WP
SLOT2-CMD	10	23	N/A
SLOT2-CLK	11	24	N/A
SLOT2-CD#	12	25	Ground
SLOT2-WP	13	26	Ground

J7 (F_PANEL): System Function Connector

J7 provides connectors for system indicators that provide light indication of the computer activities and switches to change the computer status. J7 is an 8-pin header that provides interfaces for the following functions.

Power LED: Pins 1 – 2



Pin #	Signal Name
1	LED(-)
2	LED(+)

ATX Power ON Switch: Pins 3 and 4

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 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.

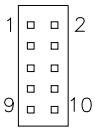
Hard Disk Drive LED Connector: Pins 7 and 8

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
7	LED(+)
8	LED(-)

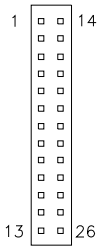
J8: Compact Flash Connector

J10: 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

J12: Parallel Port Connector

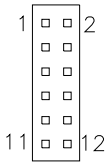


Signal Name	Pin #	Pin #	Signal Name
Line printer strobe	1	14	AutoFeed
PD0, parallel data 0	2	15	Error
PD1, parallel data 1	3	16	Initialize
PD2, parallel data 2	4	17	Select
PD3, parallel data 3	5	18	Ground
PD4, parallel data 4	6	19	Ground
PD5, parallel data 5	7	20	Ground
PD6, parallel data 6	8	21	Ground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

J13: LCD Backlight Connector

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

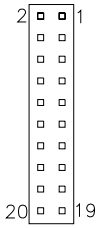
J14: Audio Connector



Signal Name	Pin #	Pin #	Signal Name
LINE-OUT L	1	2	LINE-OUT R
JD-OUT	3	4	Ground
LINE-IN L	5	6	LINE-IN R
JD-IN	7	8	Ground
Mic-In L	9	10	Mic-In R
JD-Mic	11	12	Ground

J15: LVDS Connectors

The LVDS connectors supports single-channel 18-bit or 24-bit displays.

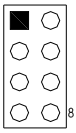


Signal Name	Pin #	Pin #	Signal Name
TX0-	2	1	TX0+
Ground	4	3	Ground
TX1-	6	5	TX1+
5V/3.3V	8	7	Ground
TX3-	10	9	TX3+
TX2-	12	11	TX2+
Ground	14	13	Ground
TXC-	16	15	TXC+
5V/3.3V	18	17	ENABKL
DDC_DATA	20	19	DDC_CLK

J16, J17: Gigabit LAN RJ-45 Connector

J18: SATA Connectors

J19~J23: USB Connectors



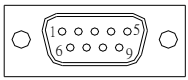
Signal Name	Pin	Pin	Signal Name
Vcc	1	5	Ground
D0-	2	6	D1+
D0+	3	7	D1-
Ground	4	8	Vcc

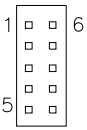
J24: +5V Output Power Connector

J24 is a 2-pin header for output only. (slim HDD)

Pin #	Signal Name
1	Ground
2	+5V

COM_PORT: COM1~COM4 Serial Ports Connector





Pin #	Signal Name (RS-232)
1	DCD, Data carrier detect
2	RXD, Receive data
3	TXD, Transmit data
4	DTR, Data terminal ready
5	Ground
6	DSR, Data set ready
7	RTS, Request to send
8	CTS, Clear to send
9	RI, Ring indicator
10	No Connect.

Appendix

A. I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses which 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
278 - 27F	Parallel Port #2(LPT2)
2F8h - 2FFh	Serial Port #2(COM2)
2B0 - 2DF	Graphics adapter Controller
378h - 3FFh	Parallel Port #1(LPT1)
360 - 36F	Network Ports
3B0 - 3BF	Monochrome & Printer adapter
3C0 - 3CF	EGA adapter
3D0 - 3DF	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
```

Any advice or comments about our products and service, or anything we can help you with please don't hesitate to contact 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