

**TS-3300 User's Manual**





Technologic Systems, Incorporated

16525 East Laser Drive  
Fountain Hills, AZ 85268

480-837-5200  
FAX 837-5300

[info@embeddedx86.com](mailto:info@embeddedx86.com)

<http://www.embeddedx86.com/>

This revision of the manual is dated

May 21, 2009

All modifications from previous versions are listed in the appendix.

Copyright © 1998-2008 by Technologic Systems, Inc. All rights reserved.

## Limited Warranty

---

Technologic Systems warrants this product to be free of defects in material and workmanship for a period of one year from date of purchase. During this warranty period Technologic Systems will repair or replace the defective unit in accordance with the following instructions:

- Contact Technologic Systems and obtain a Return Material Authorization (RMA) number and a copy of the RMA form.
- Fill out the RMA form completely and include it and dated proof of purchase with the defective unit being returned. Clearly print the RMA number on the outside of the package.

This limited warranty does not cover damages resulting from lightning or other power surges, misuse, abuse, abnormal conditions of operation, or attempts to alter or modify the function of the product.

**This warranty is limited to the repair or replacement of the defective unit. In no event shall Technologic Systems be liable or responsible for any loss or damages, including but not limited to any lost profits, incidental or consequential damages, loss of business, or anticipatory profits arising from the use or inability to use this product.**

Repairs made after the expiration of the warranty period are subject to a flat rate repair charge and the cost of return shipping. Please contact Technologic Systems to arrange for any repair service.

## Table Of Contents

---

LIMITED WARRANTY .....	ii
1 INTRODUCTION .....	5
2 PC COMPATIBILITY .....	5
2.1 Operating Systems .....	5
3 POWER .....	6
4 MEMORY .....	6
4.1 SDRAM .....	6
4.2 Flash .....	6
4.3 Compact Flash cards and DiskOnChip modules .....	6
4.4 Using the SanDisk USB Compact Flash Card Reader .....	7
4.5 Battery-Backed SRAM .....	7
5 SERIAL PORTS .....	7
5.1 Serial Port Configuration Registers .....	7
5.2 Serial Port Hardware .....	8
5.3 RS-485 Support .....	8
5.4 Adding Serial Ports .....	9
6 DIGITAL I/O .....	9
6.1 DIO1 Header .....	9
6.2 DIO2 Header .....	10
6.3 Using LCD Port as Digital I/O .....	10
6.4 DIO on the PC/104 bus .....	11
7 LCD INTERFACE .....	11
8 MATRIX KEYPAD SUPPORT .....	12
9 A/D CONVERTER .....	12
9.1 Single Sample Acquisition Procedure .....	13
9.2 A/D Converter BIOS Call .....	13
10 THE 10 BASE-T ETHERNET PORT .....	13
11 REAL TIME CLOCK .....	15
12 WATCHDOG TIMER AND SOFTWARE RESET .....	16
13 LED AND JUMPERS .....	17
14 PC/104 BUS EXPANSION .....	18
15 LOADING OR TRANSFERRING FILES .....	19
15.1 Developing with the Technologic Systems TS-9500 .....	19
15.2 Transferring files with Compact Flash .....	19
15.3 Zmodem Downloads .....	19
15.4 Manufacturing Mode .....	20
16 DEBUGGING .....	20
16.1 Integrated BIOS Debugger .....	20
16.2 Using other debuggers .....	21
17 VIDEO, KEYBOARD, AND CONSOLE REDIRECTION .....	21
18 SYSTEM BIOS SETUP SCREENS .....	22
19 FEEDBACK AND UPDATES TO THE MANUAL .....	24
APPENDIX A - BOARD DIAGRAM AND DIMENSIONS .....	24
APPENDIX B - OPERATING CONDITIONS .....	24
APPENDIX C - SYSTEM MEMORY MAP .....	25
APPENDIX D - SYSTEM I/O MAP .....	26
APPENDIX E - BIOS INTERRUPT FUNCTIONS .....	27
Int 15h / Function B000h- Technologic Systems BIOS information .....	27
Int 15h / Function B010h- LED Control .....	27
Int 15h / Function B042h- Alphanumeric LCD Support .....	28

Int 15h / Function B040h– Matrix Keypad Support.....28

Int 15h / Function A1h– Console I/O Redirection.....28

Int 15h / Function B021h– JP5 Status .....29

Int 15h / Function B020h- Jumper Pin Status .....29

APPENDIX F - DIRECT CONTROL OF THE 386EXDIO PINS.....30

APPENDIX G - USING A 16.5 MHZ BAUD CLOCK .....31

APPENDIX H - FURTHER REFERENCES.....32

APPENDIX I - MANUAL REVISIONS .....32

## 1 Introduction

---

The model TS-3300 is a compact, full-featured PC compatible Single Board Computer based on the Intel 386EX processor. At 33 MHz, it is approximately 40% faster than our 2000 series of 25 MHz 386EX based products. PC compatibility allows for rapid development since you can use standard PC development tools such as Turbo C or Power Basic or Linux based tools as well. If you have done work in the PC world in the past, you will find you can now build applications for a very small target that does not require a keyboard, video, floppy disks, or hard drives.

By adding the optional TS-9500 daughter board, you can compile and debug directly on the TS-3300 with standard VGA video and keyboard interfaces. Alternatively, you can typically write and debug code on a host PC using standard development tools for the PC platform, then simply copy it to and run it on the TS-3300 without modification. If additional peripherals are required, the PC/104 expansion bus allows for many standard functions available off-the-shelf. It is also very simple to create a custom PC/104 daughter board for those special features that differentiate your product. Technologic Systems can provide technical support as well as a free quotation for any custom hardware, software, or BIOS modifications you may require.

This manual is fairly short. This is because for the most part, the TS-3300 is a standard x86-based PC compatible computer, and there are hundreds of books about writing software for the PC platform. The primary purpose of this manual is documenting where the TS-3300 differs from a standard PC.

## 2 PC Compatibility

---

PC compatibility requires much more than just an x86 processor. It requires PC compatible memory and I/O maps as well as a PC compatible BIOS. The General Software EMBEDDED BIOS offers a high degree of compatibility with past and present BIOS standards allowing it to run off-the shelf operating systems and application software.

The EMBEDDED BIOS has been tested with all major versions of DOS, including MS-DOS, DR-DOS, and Embedded DOS 6-XL; all major versions of OS/2, including MS-OS/2 and IBM OS/2; MS-Windows 3.1, Windows-95, Windows NT, and NetWare 386.

### 2.1 Operating Systems

Technologic Systems Embedded PCs are compatible with a wide variety of x86-based operating systems. A partial list OSes currently used with our boards by customers includes:

- TNT Embedded Toolsuite, Phar Lap Software
- UCos II
- RTKernel, On Time Software
- RTEMS, On-Line Applications Research Corporation
- DOS with WATTCP, public domain TCP/IP source code for DOS
- Linux

The TS-3300 is shipped, free of charge, with Embedded DOS ROM by General Software.

The TS-3300 can be shipped upon request with Linux pre-installed for a nominal fee. The Linux file system and kernel is also freely available on the web should you wish to install it yourself. Typically, the Linux OS requires a 16MB or larger Compact Flash or an M-System's DiskOnChip.

## 3 Power

---

The TS-3300 requires **regulated** 5VDC at 800 mA (typical). A quick release screw-down terminal block for the 5V power and power GND connections is provided for easy connection to an external power supply.

When power is first supplied to the TS-3300, the board mounted LED is immediately turned on under hardware control. Once the processor begins execution, the LED is turned off. The LED then turns on then off to provide a characteristic blink during execution of POST. If the LED does not turn on at all, the most likely problem is the power supply. Check that the +5V and GND connections are not reversed. A diode protects the board against damage in such a situation, but it will not run.

Please note that supply voltages over 6 VDC may damage the TS-3300. Be sure to use a **regulated** 5 VDC power supply.

## 4 Memory

---

### 4.1 SDRAM

The TS-3300 has a total of 8 Megabytes (MB) of high-speed SDRAM providing 640 Kilobytes (KB) of base memory, 7 MB of extended memory, and 128 KB of shadow RAM for the BIOS. This is identical to a standard PC memory map. The TS-3300 can be ordered with 16MB or 32MB of SDRAM, but it is not field upgradeable.

The TS-3300 SDRAM chips are soldered directly to the board. By not using a SIMM socket, the TS-3300 is much more reliable in high-vibration environments.

### 4.2 Flash

There is a total of 1 or 2 MB of Flash memory on the TS-3300 with 128 KB reserved for the BIOS. During POST, this 128 KB area is copied from Flash into SDRAM at addresses E0000h through FFFFFh for improved performance (a standard technique known as BIOS Shadowing). The remainder of the Flash memory (896KB or 1920 KB) is configured as two solid-state disk (SSD) drives appearing as drive A and drive B. Drive A is always present and uses 896 KB of Flash memory while drive B uses the remaining 1024 KB of Flash memory if the 2MB option is present. Both drives are fully supported by the BIOS as INT 13h drives.

The physical Flash memory is accessed by the BIOS in protected mode at memory address 52M.

The Flash memory is guaranteed capable of a minimum of 100,000 write/erase cycles. This means that if you completely erase and rewrite the SSD drive 10 times a day you have over 27 years before any problems would occur. Reading the SSD produces no wear at all.

Flash drive A is **read-only** when JP3 is not installed. Write protecting the drive can be useful if there are critical files in the final product that must be very secure.

The Flash memory is guaranteed capable of a minimum of 100,000 write/erase cycles. This means that if you completely erase and rewrite the SSD drive 10 times a day you have over 27 years before any problems would occur. Reading the SSD produces no wear at all.

### 4.3 Compact Flash cards and DiskOnChip modules

If 2MB of Flash is insufficient for your application, additional non-volatile storage can be added with a Compact Flash card or an M-Systems DiskOnChip module. Either of these products can supply additional storage that will behave much as a hard drive does in a typical PC with sizes ranging from 8MB to 512MB. These products are inherently more rugged than a hard drive since they are completely solid-state with no moving parts.

The Compact Flash card has the added advantage of being removable media. A SanDisk USB Compact Flash reader/writer (which is included in the TS-3300 Developer's Kit) is recommended for the host PC for file transfers. This results in the ability to quickly move files from a host PC to the TS-3300

using a Compact Flash card as the removable media. Since the Compact Flash card appears as a standard IDE drive on the TS-3300, it uses no additional RAM for drivers. While a USB Compact Flash reader allows for hot swapping of the Compact Flash card on the host PC, the Compact Flash interface on the TS-3300 is not hot swappable, the TS-3300 must be rebooted after removing or installing a Compact Flash card.

The DiskOnChip module can be installed into the 32-pin socket near the center of the board. DiskOnChip modules are available from Technologic Systems as well as other distributors. It is compatible with DOS as shipped, and drivers for other operating systems (such as Linux) are available. If a DiskOnChip is installed, it will simply appear as drive C. The DiskOnChip is accessed through an 8 KB range of memory at D8000h through D9FFFh. If you are installing a PC/104 daughter card that uses memory mapped I/O, it must not conflict with this address range if the DiskOnChip is installed. Additionally, in a DOS environment the DiskOnChip firmware uses approximately 30 KB of user RAM (below 640 KB).

#### **4.4 Using the SanDisk USB Compact Flash Card Reader**

This device allows for a very fast and reliable method of moving files between the host PC and target SBC (TS-3300). For best results, we have noticed that it is best to boot the host PC with a Compact Flash card installed in the SanDisk USB Reader. The Compact Flash card can then be hot swapped (inserted or removed without rebooting the host PC).

#### **4.5 Battery-Backed SRAM**

The 32-pin socket can also optionally hold 32 KB of battery-backed CMOS SRAM memory. This or the DiskOnChip may be installed, but not both.

Battery backed SRAM provides non-volatile memory with unlimited write cycles and no write time degradation, unlike Flash memory. The SRAM uses an additional 32 KB range of D8000h through DFFFFh. If the SRAM is installed, PC/104 daughter card that uses memory mapped I/O must not conflict with this address range.

I/O location 75h, bit 0 can be read to determine whether the SRAM option is installed; a '1' in bit 0 indicates that it is installed, a '0' that it is not.

## **5 Serial Ports**

---

The two PC compatible asynchronous serial ports (COM1 and COM2) provide a means to communicate with external serial devices such as printers, modems, etc. Each is independently configured as a standard PC COM port that is compatible with the National Semiconductor NS16C450. COM1 appears in the I/O space at 3F8h – 3FFh and uses IRQ4. COM2 is located at 2F8h – 2FFh and uses IRQ3.

By changing an internal configuration register in the 386EX, the serial clock to the COM ports can be switched to 16.50 MHz. This feature allows baud rates higher than 115 Kbaud, as well as non-standard lower baud rates (such as 24 Kbaud). See Appendix G for further information.

The COM ports may also be configured to use a DMA channel, which is handy when very high baud rates are being used. When enabled, a DMA request is issued any time a serial port's receive buffer is full or its transmit buffer is empty. This allows higher speed operation with much lower CPU overhead. See the Intel 386EX User's Manual for further details.

### **5.1 Serial Port Configuration Registers**

Because both serial ports are 100% PC compatible, software written for the PC that accesses serial ports directly or through standard BIOS calls will work without modification on the TS-3300. The details

of the COM port internal registers are available in most PC documentation books or the data sheet for the National Semiconductor NS16C450 may be consulted.

### 5.2 Serial Port Hardware

Each serial port has 4 lines buffered: the Rx and Tx data lines and the CTS / RTS handshake pair. This is quite sufficient to interface with the vast majority of serial devices. The serial lines are routed to 10-pin headers labeled COM1 and COM2. A serial adapter cable can be plugged into the header to convert this into a standard DB9 male connector. The pin-out for the 10-pin header and DB9 male connector are listed below. The RTS signal also drives the DTR pin on the serial ports; DTR is always the same state as RTS. In addition, RTS is also used to enable the RS-485 transmitter (see below for more details).

5V Power	10	5	GND	
	NC	9	4	DTR (RTS) [out]
[in]	CTS	8	3	TX data [out]
[out]	RTS	7	2	RX data [in]
	NC	6	1	NC

Figure 1 - Serial Port Header and DB9 Pin-out [signal direction is in brackets]

**PLEASE NOTE:** The serial port headers use a non-standard numbering scheme. This was done so the header pins would have the same numbering as the corresponding DB-9 pin; i.e. pin 8 (CTS) on the header connects to pin 8 on the DB -9

### 5.3 RS-485 Support

An option is available to add support to COM1 for half duplex or full duplex RS-485. RS-485 drivers allow communications between multiple nodes up to 4000 feet (1200 meters) via twisted pair cable. Half-duplex RS-485 requires one twisted pair plus a Ground connection, while full duplex requires two twisted pair plus a Ground.

For half-duplex operation, a single twisted pair is used for transmitting and receiving. The serial port's RTS signal controls the RS-485 transmitter/receiver. When RTS is asserted true (bit 1 of the modem control register = 1), the RS-485 transmitter is enabled and the receiver disabled. When RTS is de-asserted the transmitter is tri-stated (disabled) and the receiver is enabled. Since the transmitter and receiver are never both enabled, the serial port UART does not receive the data transmitted.

#### RS-485 Quick start procedure:

1. The RS-485 option must be installed
2. Install JP6 for full-duplex or JP7 for half-duplex RS-485 operation
3. Attach the RS-485 cable to the 3-pin or 5-pin terminal strip connector.
4. Set the COM1 UART serial parameters (baud rate, data, parity, and stop bits, interrupts, etc).
5. To transmit data, assert RTS and write the data to the UART
6. To receive data, deassert RTS and read the data from the UART

Position (with terminal strip facing you)	Left		Center		Right
Silk-screen Label	TX+	TX-	GND	RX-	RX+
Half-duplex Usage	TX+ / RX+	TX- / RX-	GND	Not Used	Not Used
Full-duplex Usage	TX+	TX-	GND	RX-	RX+

Table 1 - RS-485 Connector Signals

The transmitter and receiver share a single pair of signals that are available in a screw-down terminal block. See the Table 1 for pin-outs.

**Note:** the correct jumper (JP6 or JP7) must be installed. See the Table 2 for details.

Fail-safe bias resistors are used to bias the TX+, TX- and RX+, RX- lines to the correct state when these lines are not being actively driven. This is an important consideration, since in a typical RS-485 installation, the drivers are frequently tri-stated. If fail-safe bias resistors are not present, the 485 bus may be floating and very small amounts of noise can cause spurious characters at the receivers. 4.7KΩ

resistors are used to pull the TX+ and RX+ signals to 5V and also to bias the TX- and RX- signals to ground. Termination resistors may be required for reliable operation when running long distances at high baud rates. Termination resistors should only be installed at each end of an RS-485 transmission line. In a multi-drop application where there are several drivers and/or receivers attached, only the devices at each end of the transmission line pair should have termination resistors.

Jumper	Receiver Source
JP6	Full-Duplex RS-485
JP7	Half-Duplex RS-485
neither	RS-232

Table 2 - COM1 Receiver Source

When neither **JP6** nor **JP7** is installed, COM1 will function normally as an RS-232 serial port.

A read at I/O location 75h bit 1 will return a "1" when the RS-485 option is installed.

### 5.4 Adding Serial Ports

If your project requires more than two serial ports, additional ports may be added via the PC/104 expansion bus. Technologic Systems offers a 2 serial / 1 parallel port card (TS-SER2), and other manufacturers sell cards with up to four additional serial ports. Typically these would be configured as COM3 or COM4 or be assigned other non-standard I/O locations. Because DOS only directly supports four serial ports, any additional ports beyond four will require software drivers.

The PC/104 bus has IRQ 3, 4, 5, 6, 7 or 9 available for additional serial ports. IRQ 5, 6, 7, or 9 can be used for additional serial port interrupts with no system changes required. If IRQ3 must be used on a PC/104 expansion card, then care must be taken since IRQ3 defaults to COM2 usage. In order to use IRQ3 for COM4, a configuration register in the UART must be changed to disable COM2 usage of this interrupt. (COM2 can still be used, but it cannot use interrupts). Contact us for details.

## 6 Digital I/O

There are 40 Digital Input/Output (DIO) lines available on the TS-3300. These are available on 3 headers labeled DIO1, DIO2, LCD and a single DIO input is on the PC/104 bus connector. In addition to the DIO signals, each header also has 5 Volt power and Ground available. The header labeled LCD can be used as 11 DIO lines or as an alphanumeric LCD interface (See Section 7). 24 of the DIO lines are arranged as three byte-wide ports that can be programmed as either inputs or outputs in groups of 4-bits. 8 more of the DIO lines can also be programmed as either inputs or outputs (in groups of 4-bits also). The remaining 8 lines have a fixed configuration of 7 inputs and 1 output. Three of the DIO lines can be programmed to cause interrupts.

### 6.1 DIO1 Header

The DIO1 port provides +5V, GND, and 14 digital I/O lines that may be used to interface the TS-3300 with a wide range of external devices. DIO lines DIO1\_0 thru DIO1\_7 are a byte-wide port accessed at I/O location Hex 7B, while the 6 other DIO lines DIO1\_8 thru DIO1\_13 are accessed in the lower 6 bits of I/O location Hex 7C. I/O location Hex 7A is a control port for DIO1. The direction of DIO lines DIO1\_0 thru DIO1\_3 is controlled by bit 0 of I/O location Hex 7A, and the direction of DIO1\_4 thru DIO1\_7 is controlled by bit 1 of I/O location Hex 7A. The direction of DIO1\_8 thru DIO1\_11 is controlled by bit 5 of I/O location Hex 7A, while DIO1\_12 and DIO1\_13 are always inputs. In all cases, when a control bit is a "1", it is setting the corresponding DIO lines to be Outputs, while a "0" sets them to be Inputs. All control bits at I/O location Hex 7A are initialized at reset to be "0". When bit 7 of I/O location Hex 7A is a "1", DIO1\_13 is connected to IRQ7 allowing this port to trigger an interrupt.

5 V	16	15	DIO1_7
DIO1_13	14	13	DIO1_6
DIO1_12	12	11	DIO1_5
DIO1_11	10	9	DIO1_4
DIO1_10	8	7	DIO1_3
DIO1_9	6	5	DIO1_2
DIO1_8	4	3	DIO1_1
GND	2	1	DIO1_0

Figure 2 – DIO1 Header Pinout

All digital outputs on this port can source 4 mA or sink 8 mA and the digital inputs have standard TTL level thresholds and must not be driven below 0 Volts or above 5.0 Volts. DIO lines DIO1\_0 thru DIO1\_7 have 4.7KΩ pull-up resistors biasing these signals to a logic"1".

### 6.2 DIO2 Header

The DIO2 port provides +5V, GND, and 14 digital I/O lines. DIO lines DIO2\_0 thru DIO2\_7 are a byte-wide port accessed at I/O location Hex 7E, while the 6 other DIO lines DIO2\_8 thru DIO2\_13 are accessed in the lower 6 bits of I/O location Hex 7F. I/O location Hex 7D is a control port for DIO2. The direction of DIO lines DIO2\_0 thru DIO2\_3 is controlled by bit 0 of I/O location Hex 7D, and the direction of DIO2\_4 thru DIO2\_7 is controlled by bit 1 of I/O location Hex 7D. The direction of DIO2\_8 thru DIO2\_11 is controlled by bit 5 of I/O location Hex 7D, while DIO2\_12 and DIO2\_13 are always inputs. In all cases, when a control bit is a "1", it is setting the corresponding DIO lines to be Outputs, while a "0" sets them to be Inputs. All control bits at I/O location Hex 7D are initialized at reset to be "0". When bit 7 of I/O location Hex 7D is a "1", DIO2\_13 is connected to IRQ5 allowing this port to trigger an interrupt.

5 V	16	15	DIO2_7
DIO2_13	14	13	DIO2_6
DIO2_12	12	11	DIO2_5
DIO2_11	10	9	DIO2_4
DIO2_10	8	7	DIO2_3
DIO2_9	6	5	DIO2_2
DIO2_8	4	3	DIO2_1
GND	2	1	DIO2_0

Figure 4 – DIO2 Header Pinout

All digital outputs on this port can source 4 mA or sink 8 mA and the digital inputs have standard TTL level thresholds and must not be driven below 0 Volts or above 5.0 Volts. DIO lines DIO2\_0 thru DIO2\_3 have 4.7KΩ pull-up resistors biasing these signals to a logic"1".

DIO2\_8 can be programmed to indicate the state of the TS-3300 LED. When bit 0 of I/O location Hex 79 is set, DIO2\_8 will be a logic "1" when the LED is on. Setting bit 0 of I/O location Hex 79, forces DIO2\_8 to be an output regardless of the state of bit 5 at I/O location Hex 7D.

### 6.3 Using LCD Port as Digital I/O

The LCD Port can be used as 11 additional digital I/O lines or it can be used to interface to a standard alphanumeric LCD display. At system reset, the port defaults to DIO mode. If using an LCD display this port can be switched to LCD mode by writing a "1" into bit 4 at I/O location Hex 7D, or the BIOS call to enable the LCD also sets bit 4 at I/O location Hex 7D (See Section 7 for LCD mode).

LCD_6	14	13	LCD_7
LCD_4	12	11	LCD_5
LCD_2	10	9	LCD_3
LCD_0	8	7	LCD_1
LCD_WR	6	5	LCD_EN
Bias	4	3	LCD_RS
GND	2	1	5V

Figure 5 – Pinout for LCD header when used as DIO

When the LCD port is in DIO mode, pins LCD\_RS and LCD\_WR are digital inputs, LCD\_EN is a digital output, and LCD\_0 thru LCD\_7 are programmable as either inputs or outputs.

LCD\_RS and LCD\_WR can be read at I/O location 73h bits 7 and 6, respectively. The state of LCD\_EN is controlled by writing to I/O location 73h bit 0.

LCD\_0 thru LCD\_7 can be read or written at I/O location 72h. The direction of this byte-wide port (pins 7 – 14) is determined by bits 2 and 3 at I/O location 7Dh. If bit 2 is a zero, then the lower 4 bits (pins 7 – 10) are inputs. If bit 2 is logic 1, then pins 7 – 10 are outputs. Bit 3

at location 7Dh controls the upper 4 bits, pins 11 – 14 in a like manner.

When bit 6 of I/O location Hex 7D is a "1", LCD\_RS is connected to IRQ6 allowing this port to trigger an interrupt.

All digital outputs on this port can source 4 mA or sink 8 mA and the digital inputs have standard TTL level thresholds and must not be driven below 0 Volts or above 5.0 Volts. LCD\_7 and LCD\_RS have 4.7KΩ pull-up resistors biasing these signals to a logic"1".

### 6.4 DIO on the PC/104 bus

Pin A1 on the PC/104 bus can be used as Digital I/O if it is in "DIO mode". This pin defaults to "IRQ1" and is used by the TS-9500 keyboard controller for keyboard interrupts. Clearing bit 2 at I/O location F824h will switch this pin to "DIO mode" and can then be programmed as either an Input or an Output based on bit 2 at I/O location F874h (zero = Output). It can then be read (as an Input) on bit 2 at I/O location F870h or written (as an Output) at I/O location F872h. Care must be taken when writing to these registers to ensure that no other bits are modified. Since the TS-9500 requires this pin for the IRQ1 function, changing this pin to "DIO mode" will prevent keyboard usage on the TS-9500.

## 7 LCD Interface

A 14-pin LCD connector is provided on the TS-3300 for interfacing with standard alphanumeric LCD displays. These displays use a common controller, the Hitachi HD44780 or equivalent. While software written for the HD44780 will work with all displays using the controller, the cable needed is dependent on the display used. For most displays, a straight-through type ribbon cable can be used. The connector on the LCD display is typically mounted on the backside of the display. **Warning** – using an incorrect cable or mounting the LCD connector on the front-side can result in a reverse power polarity and can damage the LCD display. Please refer to your LCD data sheets for in-depth information.

Pin	Function	Comments
1	LCD 5V	
2	LCD GND	
3	LCD_RS	Register Select (Buffered A0)
4	Bias	680 Ohm to GND
5	LCD_EN	Active high Enable
6	LCD_WR#	Active low Write
7	LCD_D1	
8	LCD_D0	
9	LCD_D3	
10	LCD_D2	D0 – D7: Buffered
11	LCD_D5	bi-directional data bus
12	LCD_D4	
13	LCD_D7	
14	LCD_D6	

Table 3 - LCD Header Signals

The TS-3300 BIOS incorporates a fairly complete set of INT10h video routines that work with the LCD. Once the LCD has been enabled (**INT15h/Func B042h** – see Appendix E below)

The LCD can be written to as the standard I/O device. This means that software can be

2	4	6	8	10	12	14
1	3	5	7	9	11	13

Figure 6 - LCD Header Pinout

developed and debugged using standard I/O calls, and the executable will work with LCD, VGA video, or redirected COM port. See the section 16 for more information.

I/O addresses **72h** and **73h** are used to access the LCD. Figure 6 shows the header pin-out, while Table 3 lists the LCD signals. The section below will briefly describe the LCD interface signals. The LCD can be controlled directly by software at these addresses.

The register select signal is simply the buffered A0 address line. Thus, reads and writes to **72h** cause register select to be low, and those to **73h** cause it to be high. Generally the LCD uses this line to separate data bytes from command bytes. See your LCD data sheet for details.

The LCD Write# signal is an active low write enable line.

LCD Enable is an active high signal, raised whenever the LCD addresses are being read or written.

D0 – D7 are bi-directional, buffered copies of the data bus and carry all data and commands to the LCD.

Table 3 is not the standard pin-outs given for LCD displays. But this pin-out allows a standard ribbon cable to be used when the ribbon cable is attached to the **backside** of the LCD.

Example LCD code is available at: <ftp://ftp.embeddedx86.com/old/downloads/util.zip>

## 8 Matrix Keypad Support

The DIO2 port, signals DIO2\_0 through DIO2\_7, may be configured to support a 4 x 4 matrix keypad. When enabled, BIOS firmware performs all the work, making the matrix keypad appear as a simple 16-key keyboard to software. This allows the use of standard keyboard access routines. The default set of keys translated by the BIOS consists of 0 – 9, A – D, \*, and #. The # key is returned as an ASCII Carriage Return character (Hex 0D). Because the user is writing the software, this set of keys is usually sufficient. However, a custom translation table can be loaded, allowing the use of function keys, arrow keys, or any other key on the keyboard.

Matrix keypad support is enabled or disabled using **INT15h, Function B040h** (see Appendix F). Once enabled, standard keyboard BIOS functions are enabled. Note that console redirection and the matrix keypad support are mutually exclusive – console redirection must be disabled to use the keypad.

A matrix keypad can be used with console redirection only if the keypad is not used as stdin, but read from as a hardware device. This is useful if one wishes to use console redirection or to use a keyboard as stdin. See section 16 for an explanation of console I/O and redirection. See the Technologic Systems download page for example code.

<ftp://ftp.embeddedx86.com/old/downloads/KEYPAD.ZIP>

When using a matrix keypad, the DIO2 signals DIO2\_0 through DIO2\_7 are not available as general I/O.

## 9 A/D Converter

The TS-3300 supports an optional eight channel, 12-bit A/D converter (ADC) capable of 60,000 samples per second. Each channel is independently software programmable for a variety of analog input ranges: -10V to +10V, -5V to +5V, 0V to +10V, or 0V to +5V. This allows an effective dynamic range of 14 bits. Each channel is overvoltage tolerant from -16V to + 16V; a fault condition on any channel will not affect the conversion result of the selected channel. This is all accomplished with a 5V only power supply; no negative supply voltage is required. The Maxim MAX197 chip can be replaced with a MAX199 chip if a lower range of analog input levels is required (-4V to +4V, -2V to +2V, 0V to 4V, and 0V to 2V).

I/O Address	Action
196h Write	Initiate A/D Conversion
196h Read	LSB of Conversion
197h Read	MSB of Conversion
75h Read	Bit 2 = 1 if A/D option installed
F860h Read	Bit 5 = 0 when Conversion completed

Table 3 – A/D registers

Bit	Description	Details
0-2	Analog channel select	Channels 0-7
3	Unipolar / Bipolar	0 = Unipolar (i.e. 0 to +5V) 1 = Bipolar (i.e. -5 to +5V)
4	Range select	0 = 5V range 1 = 10V range
5-7		Set to zero

Table 6 – A/D Control Register (196h Write)

### 9.1 Single Sample Acquisition Procedure

An acquisition is initiated by writing to I/O location 196h. The value written to I/O location 196h determines the channel to convert (bits 0-2) and selects one of four input ranges (bits 3,4). Bits 5-7 should be set to zero. After the write cycle to I/O location 196h, the MAX197 completes the A/D conversion in 11 μS. Bit 5 at I/O location F860h may be polled to determine when the conversion is complete. The conversion result is now available at locations 196h (LSB) and 197h (MSB). A single word read at I/O 196h can also be used. When using unipolar modes, the result is in binary format with the upper 4 bits of the MSB equal to zero. When a bipolar mode is used, the result is in twos-complement binary with the upper 4 bits (Bits 12-15) equal to bit 11 (sign extended).

If more details on the A/D converter specifications are required, the Maxim web site is listed in Appendix H.

### 9.2 A/D Converter BIOS Call

An A/D acquisition can also be obtained through BIOS call int 15h, function B050h. By using a BIOS call, your code will operate safely even when running on a development machine without the ADC, because the function call will not "hang" if there is a hardware fault (MAX197 not populated). If the ADC completion bit is not true after 50 μS, the routine exits with an error condition.

Int 15h / Function B050h –

ENTRY:

AX = B050h

BL = Value to write into A/D Control register (See Table 6)

EXIT:

CY = 0 (no error)

AH = 0 – No Error

1 – bad subfunction

2 – bad input registers (i.e. if BL bit 5 set)

3 – ADC option not present (I/O 7Dh bit 0 = 0)

4 – Hardware error (A/D timeout)

BX = A/D Conversion value

Pin	Description
1	Channel 0
3	Channel 1
5	Channel 2
7	Channel 3
9	Channel 4
11	Channel 5
13	Channel 6
15	Channel 7
Even	Analog GND

Table 7  
ADC Header

2	4	6	8	10	12	14	16
1	3	5	7	9	11	13	15

Figure 3 - ADC Header Pinout

## 10 The 10 Base-T Ethernet Port

The TS-3300 has full-function IEEE 802.3 Ethernet capability (10 Mbit/sec) provided by a Cirrus Logic CS8900A Ethernet controller. The CS8900A is a single-chip, 16-bit Ethernet controller that includes such features as full-duplex operation, power saving shutdown modes, and LED indicators for link status and activity. The physical interface is 10Base-T (RJ45 connector).

The TS-3300 has both a LINK LED and a LAN LED built into the RJ-45 connector that indicates the current ethernet status. The LINK LED (right side of connector) is active when valid ethernet link pulses are detected. This LED should be ON whenever the TS-3300 is powered on and properly connected to a 10BaseT Ethernet network. The LAN LED (left side of connector) should pulse ON briefly when any network traffic is detected. This includes all traffic, not just that sent to or from the TS-3300. Both of these LEDs are controlled by the CS8900A and do not require initialization. Additionally, the LEDs can be placed under software control, allowing the customer application use of the LEDs for feedback. Please see the CS8900A User Manual, Appendix H, for further details.

The hardware settings for the CS8900A are stored in a non-volatile EEPROM chip, programmed before shipment. The settings are – interrupt IRQ12, I/O address range 300h – 30Fh, and I/O mapped operation. The hardware MAC address is also stored in this chip.

## DOS TCP/IP configuration

A standard packet driver for DOS is installed on the board as shipped, along with sample network applications written with the public domain Waterloo TCP/IP software (WATTCP). WATTCP is a freely available package (including source code) that provides TCP/IP connectivity for programs written for the DOS environment. See the Technologic Systems download page

<ftp://ftp.embeddedx86.com/old/downloads/wat2001t.zip>

In addition, we have written a simple DOS HTTP web server using WATTCP that is included on the TS-3300 utility disk. The simple web server uses CGI calls to control a DIO pin from a web browser. Full source code is included, and you are free to modify and extend the code for your own use on Technologic Systems Single Board Computers.

The DOS packet driver (EPKTISA.COM) is loaded by AUTOEXEC.BAT once DOS starts, hardware settings are read from the EEPROM chip and used by the packet driver to initialize the CS8900.

The TCP/IP settings for the WATTCP code are stored in the WATTCP.CFG configuration file in the A:\ETHERNET directory, this file must be modified for the network environment where the TS-3300 will be installed.

## WATTCP.CFG configuration file

```
my_ip=192.168.0.20           // IP address of this Ethernet interface.
hostname="epc.embeddedx86.com" // Host name of this computer.
netmask=255.255.255.0       // Used to determine which IP's are local.
gateway=192.168.0.1         // Gateway for internet access.
nameserver=192.168.0.1      // Name server for domain name lookups.
```

With the WATTCP.CFG file properly setup and the 10 base-T cable connected, you should be able to ping other nodes on the network.

Ping example:

```
[A:\]ping www.embeddedx86.com
Technologic Systems Example Configuration
Pinging 'www.embeddedx86.com' [209.130.84.83]
sent PING # 1 , PING receipt # 1 : response time 0.00 seconds
```

```
Ping Statistics
Sent          : 1
Received     : 1
Success      : 100 %
Average RTT  : 0.35 seconds
[A:\]
```

Other WATTCP examples include: serial to telnet redirector, http file download, telnet server, and finger. Many more can be downloaded from the internet as freeware.

## @BI L`H7 D#D`Vc bZ[ i fUjcb`

When using the TS-3300 with Linux, the CS8900 driver can be included in the kernel or loaded as a kernel module. If the Technologic Systems kernel is used, the CS8900 driver is built into the kernel. The settings stored in EPROM on the TS-3300 are used to configure the CS8900.

The TCP/IP settings for the 2800 are configured in the file '/etc/config/ifcfg-eth0', here is a listing:

```
DEVICE=eth0           # Name of Ethernet interface
IPADDR=192.168.0.50   # IP address of this Ethernet interface.
NETMASK=255.255.255.0 # Used with NETWORK to determine local IP's.
NETWORK=192.168.0.0   # Used with NETMASK to determine local
IP's.
BROADCAST=192.168.0.255 # Broadcast IP for system wide messages.
```

The TCP/IP network settings are configured in the file '/etc/config/network', here is a listing:

```
NETWORKING=yes
FORWARD_IPV4=yes
DEFRAG_IPV4=no
GATEWAY="192.168.0.1" # Gateway for internet access.
HOSTNAME=epc          # Host name of this computer.
```

The TCP/IP name resolution server is configured in the file '/etc/resolv.conf', here is a listing:

```
Nameserver 192.168.0.1 # Name server for domain name lookups.
```

To access the web server, open a web browser and enter "192.168.0.50" as the address. This should display the sample web page which demonstrates some of the functionality of Apache with PHP.

Use the "ifconfig" command at the bash prompt to display the status of the CS8900.

## %%FYU`H]a Y'7`cW`

The Dallas Semiconductor DS12887 is used for the PC compatible battery-backed real-time clock. It is a completely self-contained module that includes a Motorola 146818 compatible clock chip, the 32.768 kHz crystal, the lithium battery, and 114 bytes of battery-backed CMOS RAM. It is guaranteed to maintain clock operation for a minimum of 10 years in the absence of power. It is located at the standard PC I/O addresses of Hex 070 and 071. The top 32 bytes (index 60h through 7Fh) are not used by the BIOS and are available for user applications.

The RTC is capable of generating a square wave output function with a period of 500 mSec to 122 uSec. The square wave output pin is connected to IRQ8 on the processor, and can be used to generate periodic interrupts. The keypad example code uses this function to generate interrupts at a 256 Hz rate.

<ftp://ftp.embeddedx86.com/old/downloads/keypad.zip>

## 12 Watchdog Timer and Software Reset

The Intel 386EX contains a 32-bit watchdog timer (WDT) unit that can be used as a watchdog timer or as a software reset function. A system reset is asserted when the WDT times out preventing a system “hanging” due to a software bug. To prevent a WDT timeout, the application must periodically “feed” the WDT by writing to a specific I/O location (WDTCLR). The value loaded into the 32-bit down-counter (WDTRLDH and WDTRLDL) allows timeout values as high as 128 seconds (for a 33 MHz clock). This value will be referred to as the “Reload Constant”.

The 32-bit “Reload Constant” determines the maximum time allowed between each Watch Dog Feed. After the WDT has been started, it cannot be turned off and the “Reload Constant “ cannot be changed. This makes it impossible for a crashed program to cause a system to “hang”.

The following steps outline how to start the Watch Dog Timer:

- 1) Write to the upper 16 bits (WDTRLDH at 0F4C0h) of the reload value (the number of clock cycles before the watchdog timer times out) followed by a write to the lower 16 bits (WDTRLDL at 0F4C2h) of the reload value.
- 2) Perform a *Watch Dog Feed*. Write 2 sequential words: 0F01Eh followed by 0FE1h to the watchdog timer clear register (WDTCLR). The WDTCLR is located at 0F4C8h.
- 3) Software must periodically perform step two (the Watch Dog Feed) before the watchdog timer times out to prevent a system reset.

Each increment in the value of the upper 16-bits (WDTRDH) of 32-bit Reload Register adds 1.98 mS to the Watch Dog Timeout Value. The lower 16 bits (WDTRDL) are insignificant and can be loaded with zero. This fact allows the following simple method of calculating a “Reload Constant” (assuming a 33MHz 386EX clock):

$$\frac{\text{Desired Time-out (in mS)}}{1.98} = \text{Upper 16-bit Reload Value}$$

For example: if a 2 second time-out is desired, then

$$\frac{2000 \text{ (mS)}}{1.98} = 1010 \text{ (decimal), or } 03F2 \text{ (hex)}$$

So WDTRLDH = 03F2h and WDTRLDL = 0000

When the WDT times out, it causes a complete hardware reset to the entire TS-3200 and also asserts RESETDRV on the PC/104 Bus.

The following code illustrates using the WDT to implement a software reset routine. The WDT is initialized with a very low “Reload Constant”.

```

Void main ()
{
const int WDTCLR      = 0xF4C8;
const int WDTRLDH = 0xF4C0;
const int WDTRLDL    = 0xF4C2;
int n = 0;

    outport(WDTRLDH, 0x0001); // Reload Constant = 1.98 mS
    outport(WDTRLDL, 0x0000); //
    outport(WDTCLR, 0xF01E);      // Watch Dog Feed word 1
    outport(WDTCLR, 0x0FE1);      // Watch Dog Feed word 2
    for (n =0; n = 1; )
        { //endless loop...system will
            //be reset in about 2 milliseconds anyway
        }
return -1;
}
    
```

For more information, please see Intel's 386EX manual, chapter 17, for more information.

### 13 LED and Jumpers

Jumper	Function	I/O Location (Hexadecimal)	Bit Position and State
JP1	Manufacturing Mode	F860	Bit 7, "0" = true
JP2	Console Redirection	F860	Bit 5, "0" = true
JP3	Write Enable Drive A	77	Bit 1, "1" = true
JP4	Reduce 386 clock to 16.5 MHz (half-rate)	77	Bit 2, "1" = true
JP5	User Jumper	F860	Bit 0, "0" = true
JP6	Enable Full-Duplex RS-485 on COM1		
JP7	Enable Half-Duplex RS-485 on COM1		
JP8	Write Enable Drive B	77	Bit 0, "1" = true
JP9	User Jumper	F860	Bit 3, "0" = true

Table 7 - Jumper Listing

The TS-3300 has an LED available for user software. The user LED can be used for diagnostics, status messages, and simple output. This signal is also available as a digital output on the DIO2 port. When power is first supplied to the TS-3300, the user LED is immediately turned on under hardware control. Once the processor begins execution, the LED is turned off, then flashed on and off again briefly. After boot, this LED is reserved solely for user applications.

There are also two LEDs on the RJ-45 Ethernet connector that are controlled by the CS8900A Ethernet Controller, and provide feedback on the status of the network. See Section 10 above for details.

If the user LED does not turn on at all when power is applied, the most likely problem is the power supply. Check that the +5V and GND

connections are not reversed. A diode protects the board against damage in such a situation.

It is preferred to use BIOS interrupt functions to interface software with the user LED and option jumpers. Please see Appendix F for further details and the utility disk for example code.

The LED is controlled by bit 6 at I/O location F862h, with a "0" turning ON the LED.

## 14 PC/104 Bus Expansion

The PC/104 is a compact implementation of the PC/AT ISA bus ideal for embedded applications. Designers benefit from using an already-developed standard, rather than creating their own. Further, the presence of a compact form-factor PC compatible standard has encouraged the development of a broad array of off-the-shelf products, allowing a very quick time to market for new products.

Pin #	Signal Name
A1	DIO_A1 †
A2 - A9	D7 through D0
A10	IOCHRDY
A11	AEN
A12 - A31	A19 through A0
A32	GND
B1	GND
B2	RESETDRV
B3	+5V
B4	IRQ9
B11	SMEMW#
B12	SMEMR#
B13	IOW#
B14	IOR#
B15	Reserved †
B16	Reserved †
B19	User Chip Select# †
B20	SYSClk (8.25 MHz)
B21	IRQ7
B22	IRQ6
B23	IRQ5
B24	IRQ4
B25	IRQ3
B28	BALE
B29	+5V
B30	OSC
B31	GND
B32	GND

Table 8 - **Supported** PC/104 Signals on the 8-bit Bus

† these signals have a non-standard usage.

The electrical specification for the PC/104 expansion bus is identical to the PC ISA bus. The mechanical specification allows for the very compact implementation of the ISA bus tailor made for embedded systems. The full PC/104 specification is available from the IEEE Standards Office under # IEEE P996.1 (see Appendix E for further information). Basically, this bus allows multiple daughter boards in a 3.6 inch by 3.8 inch form factor to be added in a self-stacking bus. Since the electrical specs are identical (except for drive levels) to a standard PC ISA bus, standard peripherals such as COM ports, Ethernet, video, LCD drivers, and Flash drives may be easily added using standard drivers.

The TS-3300 implements a sub-set of the 8-bit or optionally the 16-bit version of the PC/104 bus. We have found this allows the support of the vast majority of PC/104 boards including all of the above mentioned examples. The one feature missing is DMA, which few PC/104 boards use.

Pin B19 (normally not used) has been reassigned to be a User Chip Select. This active low signal decodes the I/O address range 140h through 15Fh, to allow for simple low-cost daughter board designs. The User Chip Select uses the 386EX CS0# signal. This can be easily reprogrammed to be at any I/O location or memory range. All Technologic Systems 3000 and 5000 series products support the User Chip Select feature.

Pin #	Signal Name
B5	-5V
B6	DRQ2
B7	-12V
B8	ENDXFR#
B9	+12V
B17	DACK1#
B18	DRQ1
B26	DACK2#
B27	TC
D8	DACK0#
D9	DRQ0

Table 9 - **Unsupported** PC/104 Signals

Pin #	Signal Name
C0	GND
C1	SBHE#
C2 - C8	LA23 - LA17
C9	MEMR#
C10	MEMW#
C11-18	D8 - D15
D0	GND
D1	MEM16#
D2	IO16#
D16	+5V
D18, D19	GND

Table 10 - Supported signals on the optional 16-bit PC/104

## 15 Loading or Transferring Files

---

Three methods are available for transferring files between a desktop PC and your TS-3300: Compact Flash, Zmodem downloads, and Manufacturing Mode. Full descriptions of each are detailed below. Other programs that use serial ports to transfer should work as well (for example, FastLynx).

### 15.1 Developing with the Technologic Systems TS-9500

The TS-9500 daughter board provides full VGA video, keyboard, and mouse interfaces necessary to work within most environments. It also has a Compact Flash card that can be used even if the TS-3300 is using a Compact Flash card as well. You can transfer your development tools (e.g. Turbo C) onto the TS-9500 Compact Flash card. Now you can compile, debug, and execute right on the TS-3300 target.

### 15.2 Transferring files with Compact Flash

The TS-3300 removable Compact Flash card can be used to easily move files from a host system. We suggest using a low-cost SanDisk™ USB compact flash card interface for your host system. This is included in the TS-3300 Developer's Kit.

For best results, we have noticed that it is best to boot the host PC with a Compact Flash card installed in the SanDisk USB Reader. The Compact Flash card can then be hot swapped (inserted or removed without rebooting the host PC).

There are two other options for connecting a Compact Flash to your host computer. A Compact Flash to PCMCIA adapter, which can map a Compact Flash as a hard drive on your laptop host computer. Or, a Compact Flash to IDE adapter, which connects to the 40 pin IDE interface on your host computer. The IDE adapter will allow you can boot to CF, run fdisk, format, etc.

### 15.3 Zmodem Downloads

Using the Zmodem protocol to send files to and from the TS-3300 is simple and straightforward. The only requirement is a terminal emulation program that supports Zmodem, and virtually all do. If you are using Windows 95 or later for your development work, the HyperTerminal accessory works well.

To download a file to the TS-3300 from your host PC, execute **DL.BAT** at the DOS command line on the TS-3300 (while using console-redirectation from within your terminal emulator) and begin the transfer with your terminal emulator. In HyperTerminal, this is 'Send File...' from the 'Transfer' menu.

To upload a file from the TS-3300 to your host PC, execute **UL.BAT <FILENAME>** at the DOS command line on the TS-3300 and start the transfer in your terminal emulator. Many emulators, HyperTerminal among them, will automatically begin the transfer themselves.

Occasionally there may be errors in transmission due to background solid-state disk operations. This is not a problem -- Zmodem uses very accurate CRC checks to detect errors and simply resends bad data. Once the file transfer is complete the file is completely error free.

For best results when using HyperTerminal, the hardware handshaking must be enabled in HyperTerminal. To increase the download speed you could also use a ram disk on the TS-3300 to receive the download file, and then copy to the flash drive.

To declare a ram disk insert this line into the config.sys file: `device=\dos\vdisk.sys /kbtouse=1024`

Please note that the utility used to perform Zmodem file transfers on the TS-3300 side is called DSZ, produced by Omen Technologies. **DSZ is shareware -- it is not free.** If you decide to use it, you are legally obligated to pay Omen Technologies. Currently the cost is \$20. Further info is available in the DSZ zip file located on the utility disk, and contact info for Omen Technologies is in Appendix G.

## 15.4 Manufacturing Mode

The TS-3300 has a special feature called 'Manufacturing Mode' which makes the on-board Flash SSD appear as just another drive on your desktop computer using a DOS device driver and a serial cable.

First, connect a null modem cable between COM2 on the TS-3300 and COM1 or COM2 of your desktop computer. Next, the TS-3300 must be placed in Manufacturing Mode. To do so, install jumper **JP1** and power cycle the unit, or enter CMOS setup and select "Start RS232 Manufacturing Link". Manufacturing Mode will automatically start once the POST routines have been executed. At this point, the TS-3300 will simply sit and wait for serial packets to arrive from a host.

Now install the Manufacturing Mode driver on your desktop computer. To do so, simply copy the `MFGDRV.SYS` device driver from the utility disk to anywhere on your desktop machine's hard drive. Then insert the following line in your `CONFIG.SYS` file and reboot:

```
DEVICE=<PATH>\MFGDRV.SYS /UNIT=0 /BAUD=56K /PORT=COMx
```

Where `<PATH>` is the full path to the location where you copied the `MFGDRV.SYS` driver, and `x` is the port on your host PC that the null modem cable is connected to (1 or 2).

The Flash SSD drive should now appear on the next free drive letter on your desktop computer (usually the D: or E: drive). Simply copy your program onto the drive, and that's it!

You can create directories, edit files, and even execute programs on your desktop computer over the Manufacturing Mode link just the way you would with a regular disk drive, just a bit more slowly.

When you are finished, turn off the TS-3300, remove the jumper, and turn it back on. Your program (if loaded into the autoexec) will now execute every time the TS-3300 is turned on.

While Manufacturing Mode is in operation, the board LED provides feedback. While idle, the LED will cycle on and off at approximately 1/2 Hertz. While data is being transferred, it will cycle much more rapidly (anywhere from 5 to 1000 Hertz)

**NOTE:** The Manufacturing Mode driver currently does **not** work correctly with Windows 95 or later, MS-DOS 6.22 or earlier is required. Please use the Zmodem transfer if you are using a Windows environment. You may need to boot directly to DOS using the utility diskette supplied with your unit if you wish to use the Manufacturing Mode method. The utility diskette will boot to DOS ROM 4.04 with the manufacturing mode driver installed. The flash drive can now be accessed and restored using the floppy.

## 16 Debugging

There are two main methods for debugging on the TS-3300: using the integrated BIOS debugger (INT3) and/or using your development tool's debugger (this usually requires the TS-9500 video and keyboard).

### 16.1 Integrated BIOS Debugger

To provide simple, direct access to the TS-3300 hardware, the system BIOS has an integrated debugger that can perform standard low-level debugger functions. The debugger allows you to perform operations such as disassemble code, display and alter the contents of memory, write to and read from I/O ports, and single-step through or breakpoint code. The debugger is not intended for use as the only debugging tool for applications, but it can be a real lifesaver when you need interactive, direct access to hardware.

The BIOS debugger can be entered by any of several methods:

- The debugger hooks the CPU exception vectors in case a divide by zero occurs, an invalid opcode is executed, or an INT 3 instruction is executed, for example. By placing an INT 3 instruction in your application code the debugger will automatically be invoked. To resume, type the 'G' command to "GO", or continue on with the rest of initialization.

- From DOS-ROM by typing 'INT3' at the command prompt. If the full command.com interpreter is running, this is an internal command. If only mini-command.com is running, this will execute a small utility that simply contains an 'INT 3' instruction.
- From the BIOS Setup main menu (started by typing 'ctrl-C' during the BIOS POST), the **9BH9F** **GMGH9A 6-CG896I ; ; 9F** selection will enter the debugger. After use, typing the 'G' (go) command will return to the SETUP screens.
- As a boot action, as a last-ditch effort if the operating system cannot be booted from the appropriate drives or out of ROM.

A complete discussion of debugger commands is available in the *Integrated BIOS Debugger Reference Manual*, included on the TS-3200 Utility Disk and also available from the Technologic Systems web site. Entering '3' will list all available commands, and a '[' (go) will return execution to the point where the debugger was called.

## 16.2 Using other debuggers

For example, the Turbo C debugger can be used on the target board using the TS-9500 video and keyboard interfaces. Any other standard PC software debugger of your choice should work just as well.

---

### %& J JXYcž? YmVcUfXžUbX'7 cbgc`Y'F YX]f YW]cb`

The TS-3300 has no video controller or keyboard interface. This was done to keep the board size small and the cost low. For applications that require it, a PC/104 video board can be added to the system easily. Technologic Systems recommends the TS-9500, a PC/104 daughter board with video, keyboard, mouse, and compact flash. This is extremely useful in speeding up the development phase and shortening the time to market. With the TS-9500 installed, developers can develop right on the target board with any compiler or debugger of their choice (such as the Turbo C compiler or debugger).

Without a video board in the system, the TS-3300 can redirect all console activity to the COM2 serial port. Simply connect an ANSI terminal (or emulator) to COM2 with a null modem cable, using serial parameters of 9600 baud, 8 data bits, no parity, 1 stop bit, and make sure jumper **>D&** is installed. All text information that would normally be displayed on a video screen is now displayed in your terminal window, and any serial data sent to the TS-3300 is seen as standard keyboard input by programs.

Please note that the console redirection support is limited by the fact that there is no actual video or keyboard hardware on the TS-3300. Programs must use the standard BIOS routines for display and keyboard input, which are rerouted to the serial port. Any program that accesses the video or keyboard hardware directly will not work. Keyboard redirection is limited simply because most of the extended keys on the keyboard (function keys and Alt key in particular) are not sent by the terminal emulator. For these reasons, the console redirection feature is meant more for system development, testing, and field repair, rather than as the primary user interface for a finished product.

If your application uses COM2, removing the jumper **>D&** easily disables console redirection.

If you wish to use a different serial port and / or baud rate for the console, the `CONSOLE.EXE` utility allows these modifications to be made. Please see the appropriate application notes for further details, available on the utility disk.

If a video board is installed on the PC/104 bus, the video BIOS on the graphics card will automatically replace the standard video routines (INT10h), disabling both the LCD display and the console redirection to COM2, regardless of the state of jumper **>D&**. If a video board is present, all console input is disabled and the TS-3300 will only accept input from a standard PC keyboard.

If LCD and/or Matrix keypad is enabled the console must be directed to standard I/O. If a TS-9500 is present, with video/keyboard enabled the standard output will be directed to video, even if the LCD is enabled, if the Matrix keypad is enabled the PC keyboard and the Matrix keypad will work at the same time.



The "Basic CMOS Configuration" screen allows setup of the Compact Flash or IDE drives, Drive Assignment mapping and Boot Order, as well as Time, Date and misc. keyboard and memory test settings. The default settings for the TS-3300 are show below.

```

+-----+
|                                     |
|               System Bios Setup - Basic CMOS Configuration               |
|               (C) 2000 General Software, Inc. All rights reserved         |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| DRIVE ASSIGNMENT ORDER: | Date:>Dec 14, 2001 | Typematic Delay   : 250 ms |
| Drive A: 5300 Flash A:  | Time: 09 : 56 : 21 | Typematic Rate    : 30 cps |
| Drive B: 5300 Flash B:  | NumLock: Disabled | Seek at Boot      : None   |
| Drive C: Ide 0/Pri Master +-----+-----+ Show "Hit Del"    : Enabled |
| Drive D: (None)         | BOOT ORDER:       | Config Box        : Enabled |
| Drive E: (None)         | Boot 1st: Drive C: | F1 Error Wait     : Enabled |
| Drive F: (None)         | Boot 2nd: Drive A: | Parity Checking   : (Unused) |
| Drive G: (None)         | Boot 3rd: DOS-ROM  | Memory Test Tick  : Enabled |
| Drive H: (None)         | Boot 4th: (None)   | Test Above 1 MB  : Enabled |
| Drive I: (None)         | Boot 5th: (None)   | Debug Breakpoints: Enabled |
| Drive J: (None)         | Boot 6th: (None)   | Splash Screen     : (Unused) |
| Drive K: (None)         +-----+-----+
| Boot Method: Boot Sector | IDE DRIVE GEOMETRY: Sect Hds Cyls | Memory |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Floppy Drive Types: | Ide 0: 2 = AUTOCONFIG, PHYSICAL | Base: |
| Floppy 0: Not installed | Ide 1: Not installed | 640KB |
| Floppy 1: Not installed | Ide 2: Not installed | Ext: |
|                         | Ide 3: Not installed | 31MB |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |
|               ^E/^X/<Tab> to select or +/- to modify                       |
|               <Esc> to return to main menu                               |
|                                     |

```

The factory defaults shown will first attempt to boot from Compact Flash as Drive C:. If no CF is installed the BIOS will then boot from Drive A:. If a CF is installed, but it is not desired to boot from this drive, change the "Boot 1<sup>st</sup>" setting to "Drive A:".

In order for the Compact Flash to function properly as an IDE hard drive, the IDE DRIVE GEOMETRY must be set to "AUTOCONFIG, PHYSICAL" and the DRIVE ASSIGNMENT ORDER must be set to "Ide 0/Pri Master" or "Ide 2/Sec Master".

In this example the Compact Flash socket on the TS-3300 (Ide 0) is set as "AUTOCONFIG, PHYSICAL" and mapped so that drive C: is assigned to "Ide 0/Pri Master". This drive is also set as the 1<sup>st</sup> boot device.

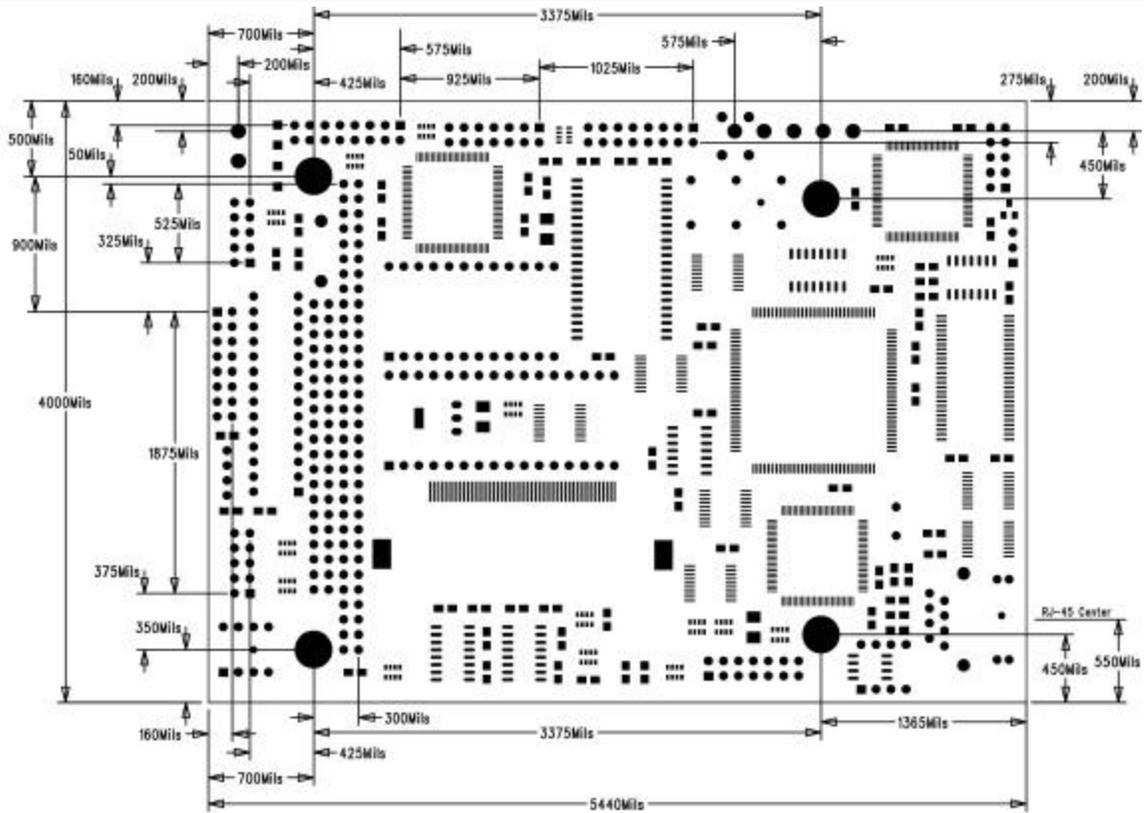
A TS-9500 may be added to the system, this board has its' own Compact Flash socket which may be set to Primary Master or Secondary Master using the DIP switches. If the TS-9500 CF is set to Primary Master (DIP switch 5 on), then no CF may be installed in the TS-3300s' socket and the CMOS settings should be exactly as shown above. The TS-9500s' CF will be mapped as Drive C: and will be the boot drive. If the TS-9500s' CF is set to Secondary Master (Ide 2), then the TS-3300s' socket may have a CF installed as Primary Master (Ide 0). To configure the TS-9500 as Secondary Master (DIP switch 5 off), change the IDE DRIVE GEOMETRY for "Ide 2" to "AUTOCONFIG, PHYSICAL" and change the DRIVE ASSIGNMENT for Drive D: to "Ide 2, Sec Master".

## 19 Feedback and Updates to the Manual

To help our customers make the most of our products, we are continually making additional and updated resources available on the Technologic Systems web site ([www.embeddedx86.com](http://www.embeddedx86.com)). These include manuals, application notes, programming examples, and updated software and firmware. Check in periodically to see what's new!

When we are prioritizing work on these updated resources, feedback from customers (and prospective customers) is the number one influence. If you have questions, comments, or concerns about your TS-3300 Embedded PC, *please let us know*. Details for contacting us are listed in the front of this manual.

## Appendix A - Board Diagram and Dimensions



ts-3300.pcb - Fri Jul 05 12:18:54 2002

## Appendix B - Operating Conditions

<b>Operating Temperature:</b>	0 to 70° C Extended temperature range is optional
<b>Operating Humidity:</b>	0 to 90% relative humidity (non-condensing)
<b>Supply Voltage:</b>	5 VDC, 5% tolerance (4.75 to 5.25 Volts)

## Appendix C - System Memory Map

Resource	Starting Address		Size
	hex	decimal	
Linear Flash Chip	340000h	52M	2M
Unused			
Extended Memory (RAM)	100000h	1M	7M or 15M or 31M
BIOS (Shadow RAM)	E0000h	896k	128k
PC/104 Bus	DA000h	872k	24k
DiskOnChip or SRAM	D8000h	864k	8k
PC/104 Bus	C8000h	800k	64k
PC/104 (typically video BIOS)	C0000h	768k	32k
PC/104 (typically video memory)	A0000h	640k	128k
Lower Memory (RAM)	00000h	00000	640k

Figure 4 – TS-3300 Memory Map

## Appendix D - System I/O Map

The following table lists the I/O addresses used by the system. This is all a very standard PC compatible I/O map. All other I/O locations from 100h through 3FFh are available on the PC/104 expansion bus.

Hex Address	Resource
500h - FFFFh	PC/104 Bus (Not recommended for use)
400h - 4FFh	PC/104 Bus, User daughter board
3F8h - 3FFh	COM1
3F6h - 3F7h	Primary IDE (Compact Flash)
3E8h - 3EFh	Reserved for COM3
3B4h - 3DFh	Video (TS-9500)
376h - 377h	Secondary IDE (TS-9500)
300h - 30Fh	CS8900A Ethernet controller
2F8h - 2FFh	COM2
2E8h - 2EFh	Reserved for COM4
278h - 27Ah	Reserved for LPT1
1F0h - 1F7h	Primary IDE (Compact Flash)
198h - 19Fh	Registers on TS-9500
196h - 197h	A/D Converter
190h - 195h	Reserved
170h - 177h	Secondary IDE (TS-9500)
140h - 15Fh	User Chip Select (PC/104 Bus)
080h - 0FFh	Internal 386EX peripherals
074h - 07Fh	DIO and Control registers
072h - 073h	LCD port
070h - 071h	RTC and CMOS memory
060h - 064h	Keyboard Controller (TS-9500)
000h - 05Fh	Internal 386EX peripherals

I/O Address	R / W	Resource
74h	Read	Product Code xxxx010 = Product Code for TS-3300
75h	Read	Bit 0 = SRAM Option
	Read	Bit 1 = RS-485 Option
	Read	Bit 2 = A/D Converter Option
76h		Reserved
77h	Read	Bit 0 = JP8
	Read	Bit 1 = JP3
	Read	Bit 2 = JP4
78h		Reserved
79h	R/W	Bit 0 = Enable DIO2_8 to track LED
7Ah	R/W	<b>Control Register for DIO1</b> Bit 0 = Direction of DIO1_0 - DIO1_3 Bit 1 = Direction of DIO1_4 - DIO1_7 Bit 5 = Direction of DIO1_8 - DIO1_11 Bit 7 = Enable DIO1_13 to drive IRQ7
7Bh	R/W	DIO1_0 thru DIO1_7 (Bits 07)
7Ch	R/W	DIO1_8 thru DIO1_13 (Bits 05)
7Dh	R/W	<b>Control Register for DIO2 and LCD</b> Bit 0 = Direction of DIO2_0 - DIO2_3 Bit 1 = Direction of DIO2_4 - DIO2_7 Bit 2 = Direction of LCD_0 - LCD_3 Bit 3 = Direction of LCD_4 - LCD_7 Bit 4 = Enable LCD Mode Bit 5 = Direction of DIO2_8 - DIO2_11 Bit 6 = Enable LCD_RS to drive IRQ6 Bit 7 = Enable DIO2_13 to drive IRQ5
7Eh	R/W	DIO2_0 thru DIO2_7 (Bits 07)
7Fh	R/W	DIO2_8 thru DIO2_13 (Bits 05)

Table 12 - TS-3300 DIO and Control Registers

The I/O ranges of 100h - 13Fh or 200h - 26Fh or 400h - 4FFh are excellent choices for PC/104 daughter board I/O usage. When designing a custom PC/104 daughter board, a User Chip Select is available on PC/104 Bus pin B19 that decodes the I/O range 140h - 15Fh. This can make for a simpler and more cost-effective design.

## 5 ddYbXJl '9 '! '6 =CG' bhYffi dh: i bWjcbg'

We have extended the standard BIOS interrupts with several functions that simplify interfacing with the TS-3300 hardware.

Many books are available with detailed information on using interrupts with just about any language. Example code is also available on the utility disk.

You can download the complete "User's Manual with BIOS Interrupt Reference" from the Technologic Systems website: <ftp://ftp.embeddedarm.com/old/saved-downloads-manuals/EBIOS-UM.PDF>

### **Int 15h / Function B000h - Technologic Systems BIOS information**

This function is mostly for our own internal use, but may be useful for user programs as well. For example, your program could have debugging code that executes on your desktop machine, but does not when the program is executing on the TS-3300.

ENTRY:

AX = B000h

EXIT:

CY = 0 (carry flag)

AH = 0

AL = SP\_VERSION

For standard versions of the BIOS, this is 0. An 'SP number' is assigned when custom modifications are made to the BIOS for a client, and it is returned in this register. Contact us for further information. (00h for standard products)

BH = BIOS Version, Major Number.

E.g. If the current BIOS version is 2.40, the register will contain 02h.

BL = BIOS Version, Minor Number.

E.g. If the current BIOS version is 2.40, the register will contain 28h (28h = 40 decimal).

CH = Base Flash Memory Size / 512kB

The TS-3300 is available with 1024kB of base Flash memory standard, but this can be increased to 2048 KB for custom orders. These would be returned as 02h or 04h, respectively.

CL = 50h

This is the hardware model number (50h = 80 decimal).

DX = Always returns: 'TS' (5453h)

### **Int 15h / Function B010h - LED Control**

This function is used to turn the board LED on and off. You can also invert the LED, i.e. if the LED is off, it will be turned on, and if it is on it will be turned off.

ENTRY:

AX = B010h

BH = 00 - LED off.

01 - LED on.

81 - LED invert.

EXIT:

CY = 0 (carry flag)

AH = 00

**Int 15h / Function B042h – Alphanumeric LCD Support**

This routine enables and disables the console on the LCD interface. It supports all LCD displays sized 2 x 40 or smaller using a Hitachi HD44780 controller.

## ENTRY:

AX = B042h  
 BH = 00 - Disable LCD  
       01 - Enable LCD  
       02 - Return current LCD status

## EXIT:

CY = Set if error, else clear if success (carry flag)  
 AH = 00 - No Error / LCD initialized OK  
 BH = 00 - LCD currently disabled / Not present  
       01 - LCD currently enabled and running.

**Int 15h / Function B040h – Matrix Keypad Support**

This function is used to enable and disable support for the matrix keypad. When enabled, the RTC periodic interrupt (IRQ8 / INT70h) is enabled at 256 Hz and hooked to the keypad scanning routine. Disabling the keypad reverses this process – IRQ8 is turned off, and INT70h is returned to the original standard BIOS function. The default translation table consists of the following: 0 - 9, A - D, \*, and # (# returns the scan-code for Carriage Return). A custom translation table is 16 words long, where each word is a scan code / ASCII pair for a key. Information on scan codes can be found in books on IBM PC hardware. Example programs are included on the utility disk. See Section 8 for further details.

## ENTRY:

AX = B040h  
 BH = 00 - Disable matrix keypad  
       - Enable matrix keypad  
       - Enable matrix keypad with a custom translation table  
 CX = Translation table segment (if BH = 11)  
 DX = Translation table offset (if BH = 11)

## EXIT:

CY = 0 (carry flag)  
 AH = 00

**Int 15h / Function A1h – Console I/O Redirection**

This function specifies the device that will be used by the BIOS to direct console input (INT 16h) and console output (INT 10h). If the LCD or Matrix keypad is to be used as the console, you must also call the LCD and/or Matrix keypad BIOS function to enable them.

## ENTRY:

AX = A1h  
 BH = 00 - Console is directed to Standard I/O. (VGA/keyboard or LCD/Matrix)  
       01 - Console is redirected to COM1  
       02 - Console is redirected to COM2  
       03 - Console is redirected to COM3  
       04 - Console is redirected to COM4

## EXIT:

CY = 0 if success, set if failure (carry flag)

AH = 00 if no error, 86h if no supported by BIOS

**Int 15h / Function B021h – JP5 Status**

This function is for backward compatibility with those who used the TS-2350 board-mounted push switch. JP5 emulates the push-switch. With JP5 installed, this function will behave as if the push switch was installed. With JP5 not installed, it behaves as if the push switch is not pressed.

ENTRY:

AX = B021h

EXIT:

CY = 0 (carry flag)

AH = 00

BX = 00 - JP5 not installed (De-asserted)

01 - JP5 installed (Asserted)

**Int 15h / Function B020h - Jumper Pin Status**

This function returns the status of the option jumpers in register BX according to the list below.

ENTRY:

AX = B020h

EXIT:

CY = 0 (carry flag)

AX = 0

BX = Jumper Pin Status. If Bit X = 1, the jumper is installed.

BX Bit	Jumper	Description
1	JP1	Manufacturing Mode
2	JP2	Console Redirection
3	JP3	Write Enable Flash A:
4	JP4	Reduce 386 clock 50%
5	JP5	User definable
7	JP8	Write Enable Flash B:
?	JP9	User definable

Table 13 - Int15h/Function B020h register BX return values

## Appendix F - Direct Control of the 386EX DIO Pins

The Intel386 EX processor has three 8-bit bi-directional I/O ports, all of which are functionally identical (Figure 16-1). Each port has three control registers and a status register. All three ports share pins with internal peripherals. Several of these pins are routed to the DIO ports. If your design does not require a pin's peripheral function, you can configure that pin for use as an I/O port. For example, if you don't need IRQ6 for PC/104, you can use the associated pin (386EX P3.4) as a DIO on JP header pin 10. Each pin can operate either in I/O mode or in peripheral mode. In I/O mode, a pin has three possible configurations:

- high-impedance input
- open-drain output (requires an external pull-up resistor)
- complementary output

In I/O mode, register bits control the direction (input or output) of each pin and the value of each output pin. In peripheral mode, the internal peripheral controls the operation (input or output) of the pin.

Each port has three control registers and a status register associated with it (). The control registers (PnCFG, PnDIR, and PnLTC) can be both read and written. The status register (PnPIN) can only be read. All four registers reside in I/O address space.

Register	I/O Address	Description
P1CFG P3CFG (read/write)	0F820h 0F824h	Port n Mode Configuration: Each bit controls the mode of the associated pin. 0 = Selects I/O mode. 1 = Selects peripheral mode.
P1DIR P3DIR (read/write)	0F864h 0F874h	Port n Direction: Each bit controls the direction of a pin that is in I/O mode. If a pin is in peripheral mode, this value is ignored. 0 = Configures a pin as a complementary output. 1 = Configures a pin as either an input or an open -drain output.
P1LTC P3LTC (read/write)	0F862h 0F872h	Port n Data Latch: Each bit contains data to be driven on to an output pin that is in I/O mode. Write the desired pin state value to this register. If a pin is in peripheral mode, this value is ignored.  Writing a value to a PL bit causes that value to be driven onto the corresponding pin.  For a complementary output, write the desired pin value to its PL bit. This value is actively driven high or low onto the pin.  For an open-drain output, a zero results in an actively driven low on the pin, a one results in a high-impedance (input) state at the pin.  To configure a pin as an input, write a one to the corresponding PL bit. A one results in a high-impedance state at the pin, allowing external hardware to drive it.  Reading this register returns the value in the register – not the actual pin state.
P1PIN P3PIN (read only)	0F860h 0F870h	Port n Pin State: Each bit of this read-only register reflects the state of the associated pin. Reading this register returns the current pin state value, regardless of the pin's mode and direction.

Table 4 – 386EX I/O Port Registers

In the default configuration, P1.5, P1.0 and P3.6 are all initialized as inputs, while P3.0, P3.1, and P3.3 are initialized as "peripherals" (IRQ4, IRQ3, and IRQ5)

**Warning: When changing these registers, always use read/modify/write procedures so that other port pins (used by on-board peripherals) are not affected.**

For example, let's say you want to use 386EX Port 3.1 (DIO pin 11) as an output rather than IRQ3 (the default configuration).

1. Read P3CFG (I/O F824h), AND it with 0FDh, write it back to F824h. (changes from peripheral to I/O pin)
2. Read P3DIR (I/O F874h), AND it with 0FDh, write it back to F874h. (this made it a complementary output)
3. To set this pin to a "1", Read P3LTC (I/O F872h), OR it with 02h, write it back.

Note: Because the serial ports are internal to the 386EX, COM1 and COM2 can still use IRQ4 and IRQ3 even when the associated pins P3.1 and P3.0 are configured as DIO pins DIO1.11 and DIO1.12 – the interrupts are simply no longer available on the PC/104 bus.

## Appendix G - Using A 16.5 MHz Baud Clock

---

Each serial port baud rate generator clock can be independently switched between either the standard 1.85 MHz clock or a 16.5 MHz clock (the internal processor clock divided by 2).

Changing the clock to 16.5 MHz allows baud rates higher than 115 kbaud (up to 1031 kbaud), as well as low error, non-standard lower baud rates (such as 24 kbaud).

The baud rate clock is controlled by I/O location 0F836h:

Bit 0 controls COM1.  
Bit 1 controls COM2.

Setting a bit to 0 uses the standard 1.85 MHz clock, and setting a bit to 1 uses the 16.5 MHz processor clock. **NOTE:** You **must not** modify the other bits of this register. You **must** use a read-modify-write procedure to change these bits. The following example in assembly illustrates this:

```
MOV  DX, 0F836          ; load the I/O address
IN   AL, DX            ; read the configuration register
OR   AL, 00000001b     ; switch COM1 to the 12.5 MHz clock
OUT  DX, AL            ; write the register
```

## Appendix H - Further References

---

### Technologic Systems Web Site

<http://www.embeddedx86.com/>

Crystal CS8900A Ethernet Controller Overview

<http://www.cirrus.com/en/products/pro/detail/P46.html>

Dallas Semiconductor DS12887 Data Sheet

<ftp://ftp.embeddedarm.com/old/downloads/Components/12887.pdf>

Intel 386EX User's Guide

<http://developer.intel.com/design/intarch/manuals/272485.htm>

Maxim Integrated Products

<http://www.maxim-ic.com/>

Omen Technologies

<http://www.omen.com/>

PC/104 Consortium Web Site

<http://www.pc104.org/>

Waterloo TCP/IP Software (WATTCP)

<http://www.wattcp.com/>

## Appendix I - Manual Revisions

---

02-08-02	Initial release
10-31-03	Added mechanical dwg.
07-24-08	Fixed broken web links
05-21-09	Updated mailing address, fixed broken web link