TS-5400 User's Manual

Technologic Systems, Incorporated

16525 East Laser Drive
Fountain Hills, AZ 85268

480-837-5200
FAX 837-5300

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 lighting 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 Non-Volatile SRAM 7
5 SERIAL PORTS 7
5.1 Serial Port Configuration Registers 8
5.2 Serial Port Hardware 8
5.3 RS-485 Support 8
5.4 Automatic RS-485 TX Enable 9
5.5 Adding Serial Ports 9
6 DIGITAL I/O 9
6.1 DIO Header 10
6.2 Using LCD Port as Digital I/O 10
7 LCD INTERFACE 10
8 MATRIX KEYPAD SUPPORT 11
9 THE 10/100 BASE-T ETHERNET PORT 12
10 REAL TIME CLOCK 15
11 WATCHDOG TIMER 16
12 LED, JUMPERS AND RESET BUTTON 17
13 PC/104 BUS EXPANSION 18
14 LOADING OR TRANSFERRING FILES 19
14.1 Developing with the Technologic Systems TS-9500 19
14.2 Transferring files with Compact Flash 19
14.3 Zmodem Downloads 19
15 DEBUGGING 20
15.1 Integrated BIOS Debugger 20
15.2 Using other debuggers 20
16 VIDEO, KEYBOARD, AND CONSOLE REDIRECTION 21
17 SYSTEM BIOS SETUP SCREENS 22
17.1 Main CMOS Configuration Screen 22
17.2 Basic CMOS Configuration 23
17.3 Adding a TS-9400 or TS-9500 with Compact Flash 23
17.4 Custom CMOS Configuration 25
17.5 Shadow Configuration 26
18 FEEDBACK AND UPDATES TO THE MANUAL 26
APPENDIX A - BOARD DIAGRAM AND DIMENSIONS 27
APPENDIX B - OPERATING CONDITIONS 27
APPENDIX C - SYSTEM MEMORY MAP 28
APPENDIX D - SYSTEM I/O MAP 29
APPENDIX E - BIOS INTERRUPT FUNCTIONS 30
Int 15h / Function B000h - Technologic Systems BIOS information 30
Int 15h / Function B010h - LED Control 30
Int 15h / Function B042h – Alphanumeric LCD Support 31

Int 15h / Function B040h – Matrix Keypad Support....31
Int 15h / Function A1h – Console I/O Redirection....31
Int 15h / Function B020h - Jumper Pin Status....32
APPENDIX F - USING A HIGHER RATE (10X) BAUD CLOCK....33
APPENDIX G - FURTHER REFERENCES....34
APPENDIX H - MANUAL REVISIONS....34

1 Introduction

The model TS-5400 is a compact, full-featured PC compatible Single Board Computer based on the AMD Elan520 processor. At 133 MHz, it is approximately 10 times faster than our other 386EX based products for only a small additional cost. 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-5400 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-5400 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-5400 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-5400 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-5400 is shipped, free of charge, with Embedded DOS ROM by General Software.

The TS-5400 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-5400 requires **regulated** 5VDC at 750 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-5400, 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-5400.

Be sure to use a **regulated** 5 VDC power supply.

4 Memory

4.1 SDRAM

The TS-5400 has a total of 16 Megabytes (MB) of high-speed SDRAM providing 640 Kilobytes (KB) of base memory, 15 MB of extended memory, and 128 KB of shadow RAM for the BIOS. This is identical to a standard PC memory map. The TS-5400 can be ordered with 32MB or 64MB of SDRAM, but it is not field upgradeable.

The TS-5400 SDRAM chips are soldered directly to the board. By not using a SIMM socket, the TS-5400 is much more reliable in high-vibration environments.

4.2 Flash

There is a total of 2 MB of Flash memory on the TS-5400 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 (1920 KB) is configured as two solid-state disk (SSD) drives appearing as drive A and drive B. Drive A uses 896 KB of Flash memory while drive B uses the remaining 1024 KB of Flash memory. 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 148M.

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.

Power failure during flash writes can cause corruption of flash drive FAT tables (A: or B:). Therefore applications writing frequently should use DiskOnChip or Compact Flash card drives which are more tolerant of power failure during write cycles.

Flash drive A is **read-only** when JP3 is not installed. Removing JP3 also makes the 128 kbyte BIOS area of the Flash write protected as well. Write protecting the A: drive can be useful if there are critical files in the final product that must be very secure.

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-5400 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-5400 using a Compact Flash card as the removable media. Since the Compact Flash card appears as a standard IDE drive on the TS-5400, 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-5400 is not hot swappable, the TS-5400 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 D0000h through D1FFFh. 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-5400). The Compact Flash (CF) card can then be hot swapped (inserted or removed without rebooting the host PC). Sometimes it is necessary to unplug the USB cable and reinstall it after swapping CF cards (at least Windows ME seems to prefer this).

Note: The TS-5400 always needs to be powered off before swapping CF cards.

4.5 Non-Volatile SRAM

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

Non-volatile 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 D0000h through D7FFFh. 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. Alternatively, these ports can be changed to the 16C550 mode with 16 byte FIFOs in both the receive and transmit UART channels. 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 Elan520, the serial clock to the COM ports can be switched to a 10 times rate (18.432 MHz). This feature allows baud rates higher than 115 Kbaud (such as 230K baud or 576K baud), as well as non-standard lower baud rates (such as 24 Kbaud). See Appendix G for further information.

The COM1 and COM2 ports may also be configured to use a DMA channel, which may be useful when very high baud rates are being used.

See the AMD Elan520 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-5400. The details of the COM port internal registers are available in most PC documentation books or the data sheet for the National Semiconductor NS16C550 may be consulted.

5.2 Serial Port Hardware

	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 – **COM2** 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 corre - sponding DB-9 pin; i.e. pin 8 (CTS) on the header connects to pin 8 on the DB -9

			5	GND	
	Rx -	9	4	Rx +	
[in]	CTS	8	3	TX data	[out]
[out]	RTS	7	2	RX data	[in]
	Tx -	6	1	Tx +	

Figure 2 – **COM1** 9-pin Sub D Pin-out [signal direction is in brackets]

PLEASE NOTE: COM1 has connections for the RS -232 port and the RS -485 port on the same 9 pin connector, only one of the two functions will be used at a time.

Each RS-232 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. If additional handshake lines are required, it will be necessary to add a TS-SER daughter board. The TS-5400 serial signals are routed to 10-pin header labeled COM2 and a standard 9-pin Sub-D male labeled COM2. A serial adapter cable can be plugged into the COM2 header to convert this into a standard DB9 male connector. The pin-outs for the 10-pin header and DB9 male connector are listed above. The RTS signal also drives the DTR pin on COM2, DTR is always the same state as RTS.

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. Bit 6 at I/O location 75h must be set to enable RTS mode or bit 7 can be set to enable Automatic mode. In RTS mode, the serial port RTS signal controls the RS-485 transmitter/receiver (See Automatic mode below). When RTS is asserted true, 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.

For full-duplex operation, two twisted pairs are used and the transmitter can typically be left on all the time. Simply use RTS mode, and set RTS true.

See Figure 2 above for connector pin-outs.

RS-485 Quick start procedure:

1. The RS -485 option must be installed
2. Install FD jumper for full -duplex or HD for half-duplex RS -485 operation
3. Attach the RS -485 cable to the 9-pin Sub-D connector.
4. Set the COM1 UART serial parameters (baud rate, data, parity, and stop bits, interrupts, etc).
5. Run Auto485.exe utility (configures bits 6 and 7 at I/O 75h) (and initializes Timer2)
6. For Half-Duplex RTS mode: To transmit data, assert RTS and write the data to the UART. To receive data, deassert RTS and read the data from the UART
7. For Half-Duplex Automatic mode: just read or write data to the UART

Note: the correct jumper (FD or HD) must be installed. See the Table 2 for details.

Jumper	Receiver Source
FD	Full-Duplex RS-485
HD	Half-Duplex RS-485
232	RS-232

Table 1 - COM1 Receiver Source

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.

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

5.4 Automatic RS-485 TX Enable

TS-5400 boards support fully automatic TX enable control. This simplifies the design of half-duplex systems since turning off the transmitter via the RTS signal is typically difficult to implement. The COM1 UART transmit holding register and the transmit shift register both must be polled until empty before deasserting RTS when using the RTS mode. The design gets more difficult when using the TX FIFO or when using a multi-tasking OS such as Linux.

In Automatic mode, Timer2 and a Xilinx PLD keep track of the bits shifting out the COM1 UART. This circuit automatically turns on/off the RS-485 transceiver at the correct times. This only requires the TIMER2 to be initialized once based on baud rate and data format, and bit 7 at I/O location 75 must be set. A utility called AUTO485.exe is included in the AUTOEXEC.bat that simplifies this task.

5.5 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 three different daughter boards (TS-SER1, TS-SER2, and TS-SER4) that add 1, 2, or 4 extra COM ports respectively. Typically these would be configured as COM3 or COM4 or be assigned other higher COM I/O locations. Because DOS only directly supports four serial ports, any additional ports beyond four will require DOS software drivers.

The TS-5400 PC/104 bus has IRQ 5, 6, 7, 9, 12, or 15 available for additional serial ports. Typically each serial port has a dedicated interrupt, but the TS-SER4 allows all four extra serial ports to share a single interrupt. This is very helpful in systems with a large number of serial ports since there are a limited number of IRQ lines available.

6 Digital I/O

There are 24 Digital Input/Output (DIO) lines available on the TS-5400. These are available on 2 headers labeled DIO and LCD. In addition to the DIO signals, each header also has 5 Volt power and Ground available, while the DIO header has an external reset available on pin 12. The header labeled LCD can be used as 11 DIO lines or as an alphanumeric LCD interface (See Section 7). 16 of the DIO lines are arranged as two byte-wide ports that can be programmed as either inputs or outputs in groups of 4-bits. An additional 4 of the DIO lines can also be programmed as either inputs or outputs, while the remaining 4 have a fixed configuration of 3 inputs and 1 output. One of the fixed input lines can be programmed to cause an interrupt.

6.1 DIO Header

The DIO port provides +5V, GND, and 13 digital I/O lines that may be used to interface the TS-5400 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 7E, while the DIO lines DIO1_8 thru DIO1_11 and DIO_13 are accessed in the lower 6 bits of I/O location Hex 7F. I/O location Hex 7D is a control port for the DIO. The direction of DIO lines DIO1_0 thru DIO1_3 is controlled by bit 0 of I/O location Hex 7D, and the direction of DIO1_4 thru DIO1_7 is controlled by bit 1 of I/O location Hex 7D. The direction of DIO1_8 thru DIO1_11 is controlled by bit 5 of I/O location Hex 7D, while DIO1_13 is always an input. 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".

5 V	16	15	DIO1_7
DIO1_13	14	13	DIO1_6
Reset#	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

When bit 7 of I/O location Hex 7D is a "1", DIO1_13 is connected to IRQ6 allowing this port to trigger an interrupt.

Pin 12 of the DIO port is an active low reset. If this pin is pulled to ground for more than 1 mS, it will cause a hard reset to the entire board. It has a 4.7K resistor pull-up to 5V, so a simple switch to ground can be used.

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_5 have 4.7KΩ pull-up resistors to 5V biasing these signals to a logic"1".

6.2 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 IRQ1 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".

7 LCD Interface

A 14-pin LCD connector is provided on the TS-5400 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 bi-directional data bus
11	LCD_D5	
12	LCD_D4	
13	LCD_D7	
14	LCD_D6	

Table 3 - LCD Header Signals

The TS-5400 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)

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

Figure 6 - LCD Header Pinout

The LCD can be written to as the standard I/O device. This means that software can be 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.embeddedarm.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 accessed as a hardware device. This is useful if one wishes to use console redirection or to use a PC

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.embeddedarm.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 The 10/100 Base-T Ethernet Port

The TS-5400 has two full-function IEEE 802.3 (10/100 Mbit/sec) Ethernet ports provided by Davicom DM9102A Ethernet controllers. The DM9102A is a single-chip, 32-bit PCI Ethernet controller that includes such features as full-duplex operation, power saving shutdown modes, LED indicators for link status/activity and auto-sense network speed detection. The physical interface is 10/100Base-T (RJ45 connector).

The TS-5400 has both a LINK/ACTIVITY LED and a 10/100 speed LED built into each RJ-45 connector that indicates the current Ethernet status. The LINK LED (left side of connector, green) is active when a valid Ethernet link is detected. This LED should be ON whenever the TS-5400 is powered on and properly connected to a 10/100BaseT Ethernet network. The LINK/ACTIVITY LED will blink to indicate network activity either inbound or outbound. The SPEED LED (right side of connector, amber) will be on when a 100 Mbit network is detected and off for a 10 Mbit network. Both of these LEDs are controlled by the DM9102A and do not require initialization by the processor.

DOS TCP/IP Configuration – Packet Driver and WATTCP

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.embeddedarm.com/old/downloads/wat2001t.zip

In addition, we have written a simple DOS HTTP web server using WATTCP that is included on the TS-5400 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 (DM9PCIPD.COM) is loaded by `AUTOEXEC.BAT` once DOS starts. Hardware settings are read from the EEPROM chips and used by the packet driver to initialize each DM9102A. Notice in the autoexec.bat there two instances of the packet driver loaded, one for each of the Ethernet ports. The second Ethernet port marked Eth_2 on the board is referred to as 'slot=2' in the autoexec.bat. Below is an example of the DOS command line to load a packet driver for each Ethernet interface:

```
DM9PCIPD 0X60
DM9PCIPD –SLOT=2 0X61
```

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-5400 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 10/100 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.

DOS TCP/IP configuration – MS Client for DOS

The Davicom Ethernet adapter has the MS Client for DOS driver available. This has been tested and found to work on Windows workgroups as well as Samba workgroups. Check the Technologic Systems web page for driver downloads (www.embeddedx86.com). Email support@embeddedx86.com with questions.

LINUX TCP/IP configuration

When using the TS-5400 with TS-Linux, the DM9102A driver can be included in the kernel or loaded as a kernel module. If the Technologic Systems kernel is used, the DM9102A driver is built into the kernel. The settings stored in EEPROM on the TS-5400 are used to configure the DM9102A.

The TCP/IP settings for the TS-Linux are configured in the file '/etc/sysconfig/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.
   ENABLE=yes              # Initialize on startup
```

The second Ethernet port on the TS-5400 is configured by the file: /etc/sysconfig/ifcfg-eth1
By default the eth1 config file has ENABLE=no, which means eth1 will not be configured on boot up. Change this to yes if you want eth1 configured by the startup scripts.

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

```
### Technologic Systems
### General Network Configuration File
###
     NETWORKING=yes
     GATEWAY=192.168.0.1                  # Gateway for internet access
     GW_DEV=eth0                          # Gateway device to use
     Hostname="miniepc.embeddedx86.com"   # Host name for this computer
     BOOTPROTO=no
     DEFRAG_IPV4=no
     FORWARD_IPV4=no
```

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 Ethernet ports.

10 Real Time Clock

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 48 bytes (index 50h 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.embeddedarm.com/old/downloads/KEYPAD.ZIP

11 Watchdog Timer

Bits 0-7	Timeout Value
01h	0.5 milliseconds
02h	0.5 seconds
04h	1 second
08h	2 seconds
10h	4 seconds
20h	8 seconds
40h	16 seconds
80h	32 seconds

Table 4 Watchdog Timer Timeout Values

Bit Position	Function
Bit 15	1 = Enable WDT
Bit 14	1 = Generate Reset at Timeout 0 = Generate Interrupt at Timeout
Bit 13	Must always be 0
Bit 12	1 = WDT Interrupt has occurred (Interrupt routine must clear this bit)
Bits 8-11	Must always be 0
Bits 0-7	Timeout Value (See Table 4)

Table 5 – WDTMRCTL Register

Register	Memory Address
WatchDog Control (WDTMRCTL)	DFCB0h
WDT Counter Bits 0-15 (Read only)	DFCB2h
WDT Counter Bits 16-31 (Read only)	DFCB4h
WDT Interrupt Mapping	DFD42h

Table 6 - WatchDog Timer Memory Map

The AMD Elan520 contains a 32-bit watchdog timer (WDT) unit that can be used to prevent a system "hanging" due to a software failure. The WDT can be programmed to cause an interrupt or a full system reset when the WDT times out allowing a guaranteed recovery time from a software error. To prevent a WDT timeout, the application must periodically "feed" the WDT by writing a "clear-count key sequence" to the Watchdog Timer Control (WDTMRCTL) register.

The WDT must be initialized with the timeout period desired. This may be as short as 0.5 mS or may be as high as 32 seconds. After the WDT has been enabled, the 32-bit watchdog counter starts counting from zero. The application software can reset this counter at any time by writing a "clear-count key sequence" to the WDTMRCTL. If this counter reaches the timeout period, then an interrupt is generated, or a system reset occurs.

If programmed to cause an interrupt, the WDT must be fed within the next timeout period or else a system reset occurs as an additional failsafe feature.

In order to load the WDTMRCTL register, a specific sequence of three word writes is required. A 3333h, followed by CCCCh, followed by the value to be loaded into the WDTMRCTL register must be written to the WDTMRCTL register.

In order to clear the WDT counter (feeding the watchdog), a "clear-count key sequence" must be written to the WDTMRCTL register. This is a specific two word write sequence with a write of AAAAh, followed by a write of 5555h. (Writing this "clear-count key sequence" has no effect on the contents of the WDTMRCTL register)

Typically a system reset at timeout is the preferred method for using the WDT, if you wish to have an interrupt occur before the system reset occurs (bit 14 = 0 in the WDTMRCTL Register), additional programming is required to "attach" a specific IRQ to the WDT. An interrupt service routine must also be written to handle this IRQ. Please see the AMD Elan520 User's Manual or contact Technologic Systems.

When using the Linux OS, we provide a driver to initialize and "feed" the WDT. We also provide a Linux sample application program with full source.

If using DOS, the following sample of Turbo C code (real-mode DOS) will enable the WDT with a 4 second timeout period :

```
int main() {
  unsigned int far *wdtctrl ;
   wdtctrl = (unsigned int far *) MK_FP (0XDF00, 0X0CB0);
   *wdtctrl = 0x3333;
   *wdtctrl = 0xCCCC;
   *wdtctrl = 0xD010 ;          /* 0xD0 = Enable WDT to Reset Mode, 0x10 = 4 second timeout */
   printf ("System will reset in 4 seconds");
   return 0;
 }
```

12 LED, Jumpers and Reset Button

Jumper	Function
JP1	**Automatic CMOS** - CMOS memory is reset every time the system boots, IDE0 and IDE2 Compact Flash are automatically detected and configured in the BIOS Configuration menu.
JP2	**Enable Serial Console -** Default to COM2. (see JP6)
JP3	**Write Enable Drive A –** Remove this jumper to write protect flashdrive A:
JP4	**Fast Console** - Console is set to 115K baud. (JP4 not installed = 9600 baud)
JP5	**User Jumper** - Read using BIOS call or directly in I/O space. (See appendix D & E)
JP6	**Console switched to COM1** (requires JP2 installed to enable console)

Table 7 - Jumper Listing

The TS-5400 has an LED available for user software. The user LED may 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-5400, 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 can be used for user applications. The state of the LED is complemented by any activity to the Compact Flash card. (accesses to the CF causes the LED to flicker)

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 can be also accessed at I/O location 77h bit 0. A Logic "1" is LED on condition.

The CF activity signal is exclusive ORed with the control bit at I/O location 77h. This causes the LED to always flicker when there is access to the CF card.

The reset button is connected in parallel with the external reset pin on the DIO connector. Pressing this button with cause a hardware reset. The system will perform a 'cold' boot.

13 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	IRQ1 †
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 †
B17	DACK1#
B18	DRQ1
B19	User Chip Select# †
B20	SYSCLK (8.25 MHz)
B21	IRQ7
B22	IRQ6
B23	IRQ5
B24	
B25	
B27	TC
B28	BALE
B29	+5V
B30	OSC (14.3 MHz)
B31	GND
B32	GND

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

† these signals have a non-standard usage.

Pin #	Signal Name
B5	-5V
B6	DRQ2
B7	-12V
B8	ENDXFR#
B9	+12V
B26	DACK2#

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

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-5400 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.

IRQ3 and IRQ4 are typically used by COM2 and COM1 and are not available on the PC/104 bus. If a daughter board must use either IRQ3 or IRQ4, it is possible to reconfigure the Elan520 registers and reassign IRQ lines. Contact us for details.

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#
D5	IRQ12
D6	IRQ15
	IRQ14
D7	(shared with CF)
D8	DACK0#
D9	DRQ0
D16	+5V
D18, D19	GND

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

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.

All Technologic Systems 3000 and 5000 series products will support the User Chip Select feature.

Memory range C8000-CFFFF is available for custom daughter boards to map into the memory space on the PC/104 bus. See section 17.4 on configuring the Custom CMOS settings.

14 Loading or Transferring Files

Two methods are available for transferring files between a desktop PC and your TS-5400: Compact Flash and Zmodem downloads. Full descriptions of each are detailed below. Other programs that use serial ports to transfer should work as well (for example, FastLynx).

14.1 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-5400 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-5400 target.

14.2 Transferring files with Compact Flash

The TS-5400 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-5400 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 you host computer. The IDE adapter will allow you can boot to CF, run fdisk, format, etc.

14.3 Zmodem Downloads

Using the Zmodem protocol to send files to and from the TS-5400 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-5400 from your host PC, execute **`DL.BAT`** at the DOS command line on the TS-5400 (while using console-redirection 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-5400 to your host PC, execute **`UL.BAT <FILENAME>`** at the DOS command line on the TS-5400 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-5400 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-5400 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 Debugging

There are two main methods for debugging on the TS-5400: using the integrated BIOS debugger (INT3) and/or using your development tool's debugger (this usually requires the TS-9500 video and keyboard).

15.1 Integrated BIOS Debugger

To provide simple, direct access to the TS-5400 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 **ENTER SYSTEM BIOS DEBUGGER** 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-5400 Utility Disk and also available from the Technologic Systems web site. Entering '**?**' will list all available commands, and a '**g**' (go) will return execution to the point where the debugger was called.

15.2 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.

16 Video, Keyboard, and Console Redirection

The TS-5400 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-5400 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 **JP2** is installed. (JP4 will change the console to 115.2K Baud) 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-5400 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-5400. 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 **JP2** 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 **JP2.** If a video board is present, all console input is disabled and the TS-5400 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.

17 System BIOS Setup Screens

17.1 Main CMOS Configuration Screen

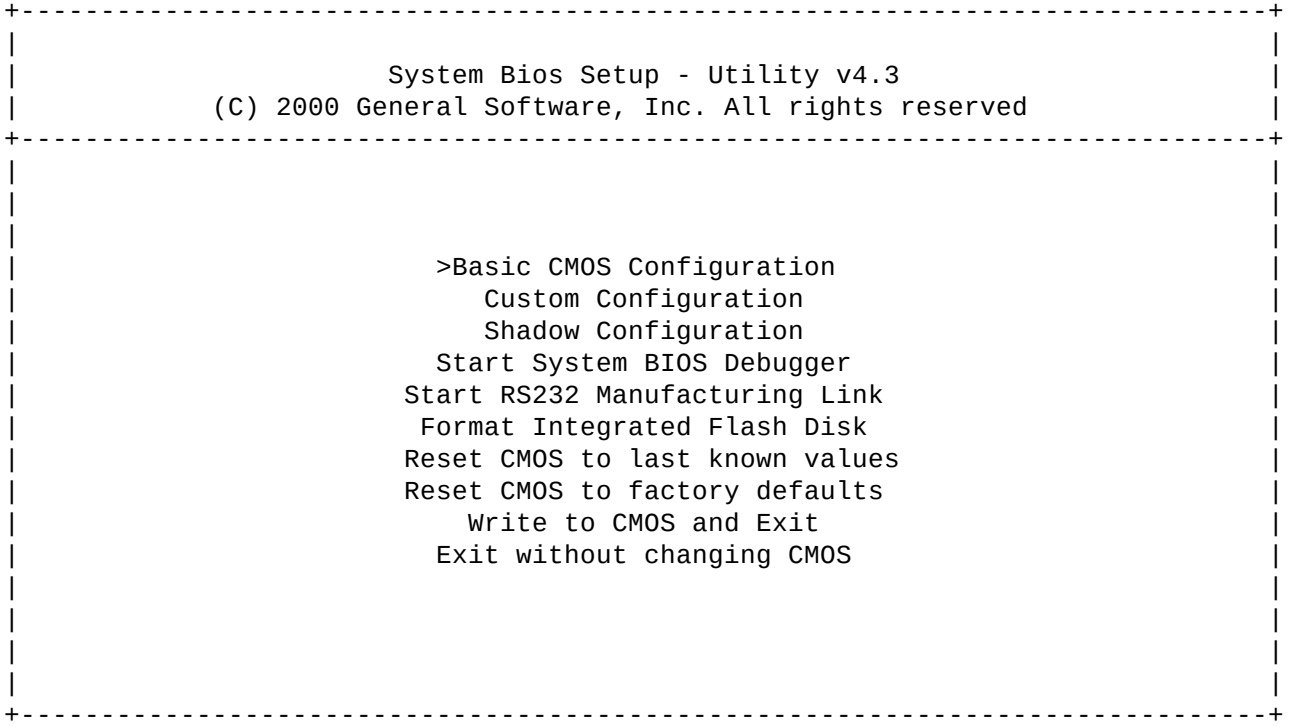
The BIOS setup screen allows the user to configure the BIOS settings, which are then stored in battery-backed "CMOS" memory in the Real Time Clock. The BIOS setup screen also allows entry into the system debugger, manufacturing mode and formatting of the integrated flash disks.

To enter the BIOS Setup screen, hit Ctrl-C during the memory test. If you have a TS-9500 installed with video/keyboard enabled hit the Delete key during the memory test.

When navigating the BIOS setup screen with the console on COM2, the arrow keys do not work. Use the Ctrl-X, Ctrl-E and Tab keys to navigate, use the Space bar, '+' and '-' keys to change fields. Use ESC to back out of a menu.

NOTE: Jumper JP1 will force a reset of the CMOS memory on every boot, any changes will be lost.

"Basic CMOS Configuration"	Setup disk drives, drive mapping, boot order, misc.
"Custom Configuration"	Setup custom features for Technologic Systems' boards.
"Shadow Configuration:	Setup ROM BIOS shadowing in RAM.
"Start System BIOS Debugger:	Enter debugger, **see section 15.**
"Start RS232 Manufacturing Link"	Enter manufacturing mode to link with host PC.
"Format Integrated Flash Disk"	Low-level format of flash drives A: and B:
"Reset CMOS to last known values"	Resets changes since last boot.
"Reset CMOS to factory defaults"	Reset to settings as shipped from Technologic Systems.
"Write to CMOS and Exit"	Save all changes made, and reboot.



```
|                          <Esc> to continue (no save)                          |
```

www.gensw.com

17.2 Basic CMOS Configuration

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-5400 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    |
+--------------------------+ Ide 0: 2 = AUTOCONFIG, PHYSICAL      |  Base:    |
| FLOPPY DRIVE TYPES:      | Ide 1: Not installed                 |   640KB   |
| Floppy 0: Not installed  | Ide 2: Not installed                 |  Ext:     |
| Floppy 1: Not installed  | 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 1st" 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".

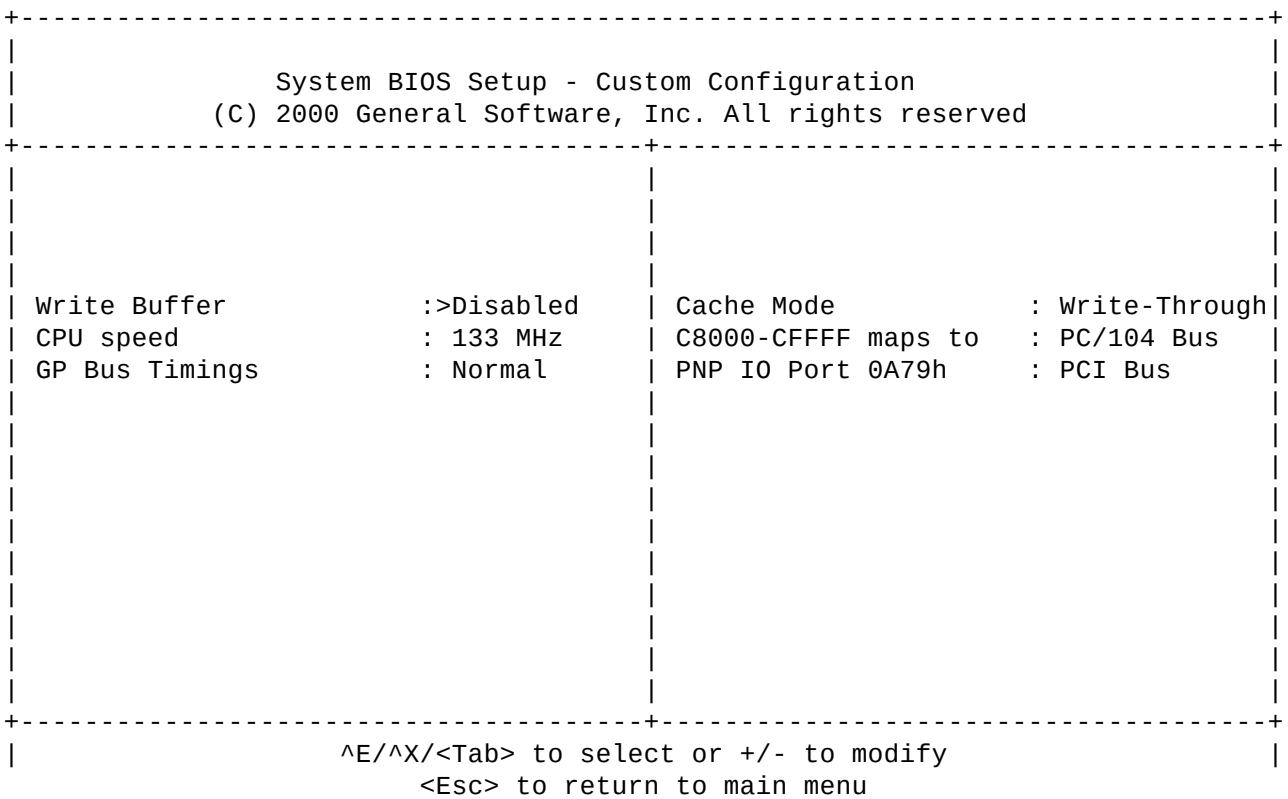
In this example the Compact Flash socket on the TS-5400 (Ide 0) is set as "AUTOCONFIG, PHYSICAL" and mapped so that drive C: is assigned to "Ide 0/Pri Master". This drive is also set as the 1st boot device.

17.3 Adding a TS-9400 or TS-9500 with Compact Flash

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 card may be installed in the TS-5400 CF socket and the CMOS settings should be exactly as shown above. The TS-9500 CF card will be mapped as Drive C: and will be the boot drive. If the TS-9500 CF card is set to Secondary Master (Ide 2), then the TS-5400 CF 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".

17.4 Custom CMOS Configuration



Write Buffer When write buffering is 'Enabled' the internal 16K byte L1 cache is used to allow zero wait state writes to SDRAM . This setting defaults to Disabled and may be 'Enabled' by the user to optimize performance.

Refer to the AMD Elan SC520 users manual section 11 for more information on write buffering.

Cache Mode When set to Write-Through all writes to SDRAM will be updated in the internal cache as well as the SDRAM. When set to Write-Back only the internal cache is updated, the cache controller will determine when to flush cache to SDRAM. This setting defaults to 'Write-Through' and may be set to 'Write-Back' by the user to optimize performance.

Refer to the AMD Elan SC520 users manual section 8.4.2.2 for more information on cache mode.

CPU Speed This will set the clock speed for the CPU core. This can be set to '100 MHz' or '133 Mhz'. The default speed for commercial temperature products is 133 MHz, the default for industrial temperature products is 100 MHz. Changing this setting will not affect industrial temperature products because they are specified by AMD with a maximum clock speed of 100 MHz.

C8000-CFFFF maps to This setting allows a block of memory space to be mapped to either the PC/104 bus or the onboard SDRAM. This setting defaults to PC/104 bus.

GP Bus Timings This setting allows the PC/104 bus timing to be stretched to allow slower devices to operate. If Plug-N-Play boards are installed, or in the case of older PC/104 daughter cards setting this entry to 'Slow' may be desired. This setting defaults to 'Normal' which has been tested and found to work with the vast majority of boards.

PNP IO Port 0xA79 This setting allows the Plug-N-Play configuration address to be mapped to the PC/104 bus. The I/O space above 0x400 is mapped to the PCI bus by default. If a Plug-N-Play device is connected to the PC/104 bus change this setting to 'GP Bus'.

AMD Elan SC520 Users Manual may be found at:
ftp://ftp.embeddedarm.com/old/downloads/Components/520_user.pdf

17.5 Shadow Configuration

```
+-------------------------------------------------------------------------------+
|                                                                               |
|                 System BIOS Setup - Shadow/Cache Configuration                |
|             (C) 2000 General Software, Inc. All rights reserved               |
+--------------------------------------+----------------------------------------+
| Shadowing                :>Chipset   | Shadow 16KB ROM at C000 : Enabled      |
| Shadow 16KB ROM at C400 : Enabled    | Shadow 16KB ROM at C800 : Disabled     |
| Shadow 16KB ROM at CC00 : Disabled   | Shadow 16KB ROM at D000 : Disabled     |
| Shadow 16KB ROM at D400 : Disabled   | Shadow 16KB ROM at D800 : Disabled     |
| Shadow 16KB ROM at DC00 : Disabled   | Shadow 16KB ROM at E000 : Enabled      |
| Shadow 16KB ROM at E400 : Enabled    | Shadow 16KB ROM at E800 : Enabled      |
| Shadow 16KB ROM at EC00 : Enabled    | Shadow 64KB ROM at F000 : Enabled      |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
|                                      |                                        |
+--------------------------------------+----------------------------------------+
|                    ^E/^X/<Tab> to select or +/- to modify                     |
                         <Esc> to return to main menu
```

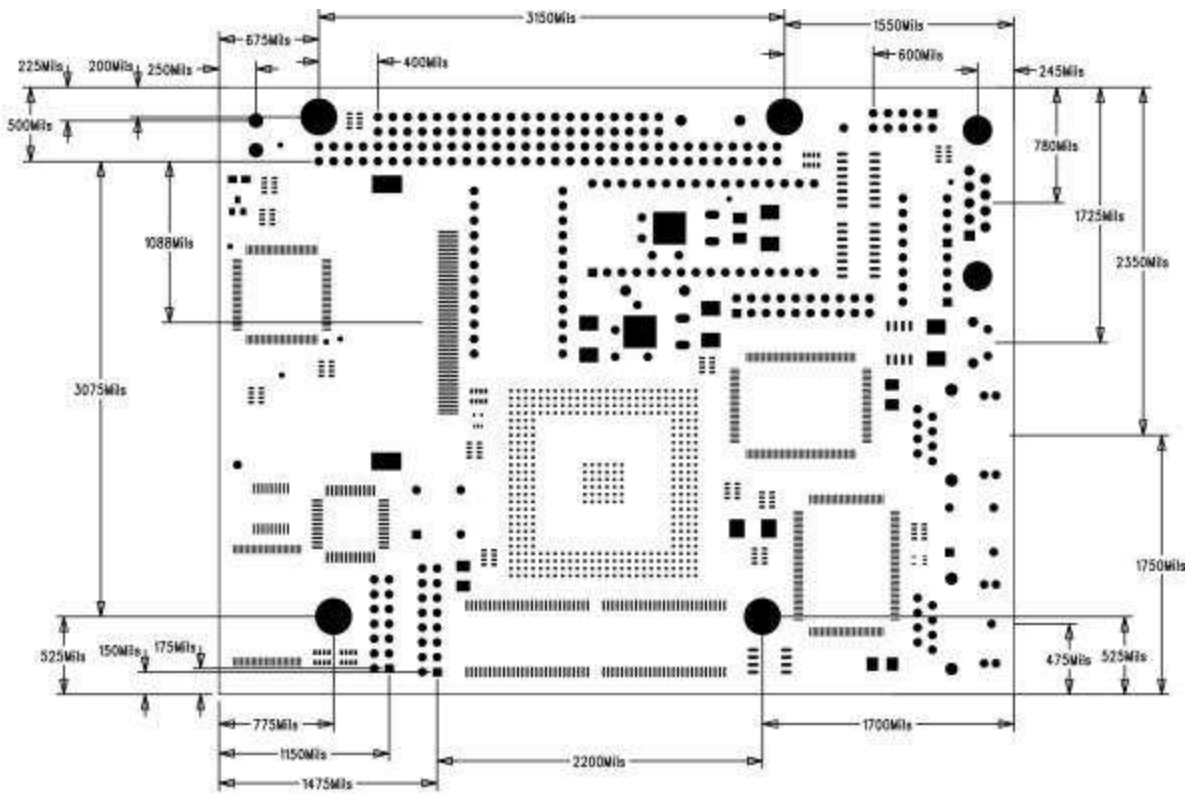
The Shadow Configuration menu allows BIOS extension ROMs to be copied to SDRAM in upper memory regions to reduce access times. Execution directly from ROM is significantly slower than executing from SDRAM. The region from C000-C7FF is usually reserved for the video BIOS extension in a PC compatible system, shadowing is enabled to optimize the video BIOS functions. The region from E000-FFFF contains the embedded BIOS code in the TS-5400. This region is shadowed to optimize BIOS code execution and to allow reprogramming of the BIOS while the BIOS is executing out of SDRAM. Other regions can be shadowed as required by specific applications.

18 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). 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-5400 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-5400A.pcb - Tue Jan 07 14:11:44 2003

Appendix B - Operating Conditions

Operating Temperature: 0 to 70° C
Extended temperature range is optional

Operating Humidity: 0 to 90% relative humidity (non-condensing)

Appendix C - System Memory Map

Resource	Starting Address hex	Starting Address decimal	Size
Linear Flash Chip	9400000h	148M	2M
Unused			
Extended Memory (RAM)	100000h	1M	15M or 31M or 63M
BIOS (Shadow RAM)	E0000h	896k	128k
Elan520 Configuration Registers	DF000h	892k	4k
PC/104 Bus	D2000h	840k	52k
DiskOnChip or SRAM	D0000h	832k	8k
PC/104 Bus or SDRAM (user configurable in CMOS setup, see sec. 17.4)	C8000h	800k	32k
PC/104 (typically video BIOS)	C0000h	768k	32k
PC/104 (typically video memory)	A0000h	640k	128k
Lower Memory (RAM)	00000h	00000	640k

Figure 3 – TS-5400 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
400h - FFFFh	PC/104 Bus (Not recommended for use)
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)
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	Reserved for A/D Converter
170h – 177h	Secondary IDE (TS-9500)
140h – 15Fh	User Chip Select (PC/104 Bus)
080h – 0FFh	Internal Elan520 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 Elan520 peripherals

Table 11 – TS-5400 I/O Map

I/O Address	R / W	Resource
74h	Read	Product Code
		40h = Product Code for TS-5400
75h	Read	Bit 0 = SRAM Option Present = "1"
	Read	Bit 1 = RS-485 Option Present = "1"
	R/W	Bit 6 = Enable RS-485 RTS mode
	R/W	Bit 7 = Enable RS-485 Automatic mode
76h		Reserved
77h	R/W	Bit 0 = LED (1 = ON)
	Read	Bit 1 = JP1
	Read	Bit 2 = JP2
	Read	Bit 3 = JP3
	Read	Bit 4 = JP4
	Read	Bit 5 = JP5
	Read	Bit 6 = JP6
78h		Reserved
7Dh	R/W	**Control Register for DIO and LCD**
		Bit 0 = Direction of DIO_0 – DIO_3
		Bit 1 = Direction of DIO_4 – DIO_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 DIO_8 – DIO_11
		Bit 6 = Enable LCD_RS to drive IRQ11
		Bit 7 = Enable DIO_13 to drive IRQ5
7Eh	R/W	DIO_0 thru DIO_7 (Bits 0-7)
7Fh	R/W	DIO_8 thru DIO_13 (Bits 0-5)

Table 12 – TS-5400 DIO and Control Registers

The I/O ranges of 100h – 13Fh and 200h – 26Fh 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.

Appendix E - BIOS Interrupt Functions

We have extended the standard BIOS interrupts with several functions that simplify interfacing with the TS-5400 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-5400.

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-5400 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 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
0	JP1	Auto CMOS configuration
1	JP2	Console Redirected to COM port (JP6 determines COM port)
2	JP3	Write Enable flash A:
3	JP4	Console set to 115K baud (JP4 off = 9600 baud)
4	JP5	User definable
5	JP6	Console switched to COM1 (JP2 must be installed)

Table 13 - Int15h/Function B020h register BX return values

Appendix F - Using a Higher Rate (10x) Baud Clock

The COM1 and COM2 serial port baud rate generator clock can be independently switched between either the standard 1.8432 MHz clock or a ten times rate of 18.432 MHz clock.

Changing to the 10x rate allows baud rates higher than 115.2 kbaud (such as 230.4 kbaud or 1152 kbaud), as well as, non-standard lower baud rates (such as 24 kbaud).

The baud rate clock for each COM port is controlled by bit 2 in the UART Control Register (UART1CTL and UART2CTL):

UART1CTL @ DFCC0h
UART2CTL @ DFCC4h

Clearing bit 2 to a "0" will change the clock to the 10X rate.

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 code written in Turbo C provides an example for changing COM1 to run at a 10X Baud rate:

```
int main () {
      unsigned char far *comreg;
      unsigned char comvalue;
      comreg=(unsigned char far *) MK_FP (0xDF00, 0xCC0);

      comvalue = *comreg;         /* Load Register */
      comvalue = comvalue & 0xFB       /*Modify it*/
      *comreg = comvalue;         /*Write it out*/

      return 0;
}
```

Appendix G - 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 H - Manual Revisions

12-14-01	Initial release
12-31-02	Updates for Rev C hardware, added mechanical dwg., added custom CMOS setup section
10-23-03	Minor editing - corrected errors
07-24-08	Fixed broken web links
05-21-09	Updated mailing address