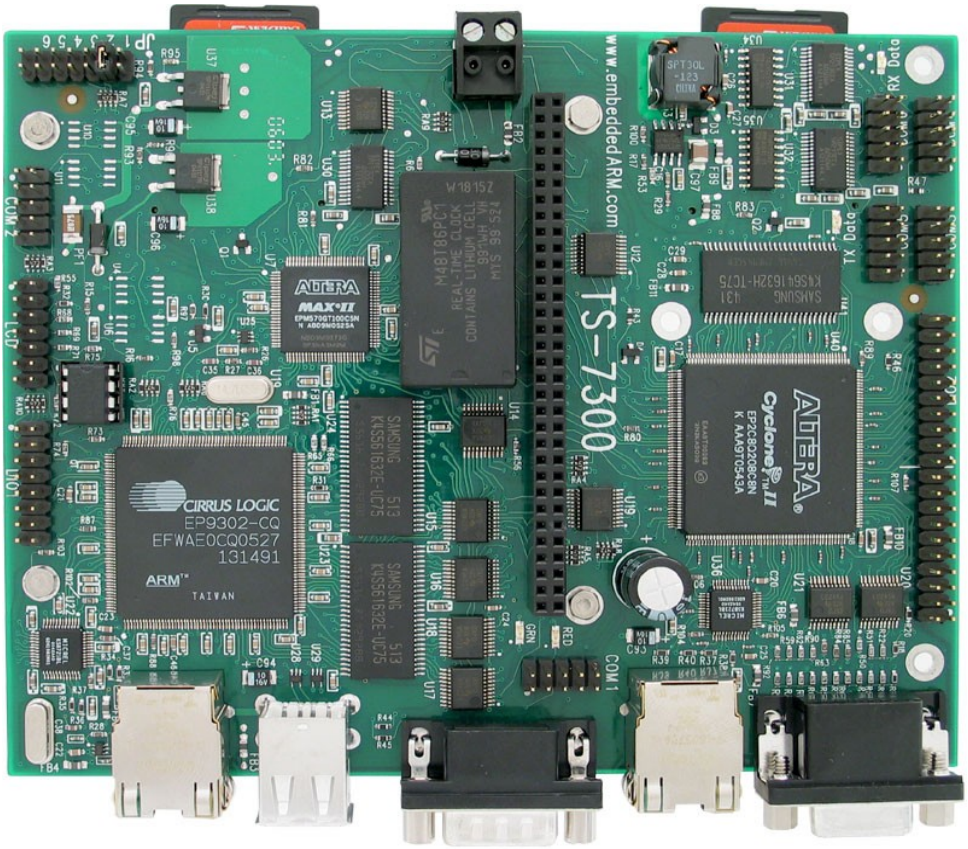
TS-7300 Manual
Hardware & Software

www.embeddedARM.com



Technologic SYSTEMS

Revision 1.7

Feedback and Update to this Manual

To help our customers make the most of our products, we are continually making additional and updated resources available on the **Technologic Systems website** (www.embeddedARM.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 Embedded Computer, please let us know at **support@embeddedARM.com**.

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 process:

A copy of the original invoice must be included when returning the defective unit to Technologic Systems, Inc.

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 repair charge and the cost of return shipping. Please, contact Technologic Systems to arrange for any repair service and to obtain repair charge information.

FCC Advisory Statement

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used properly (that is, in strict accordance with the manufacturer's instructions), may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class A computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the owner will be required to correct the interference at his own expense.

If this equipment does cause interference, which can be determined by turning the unit on and off, the user is encouraged to try the following measures to correct the interference:

- Reorient the receiving antenna.
- Relocate the unit with respect to the receiver.
- Plug the unit into a different outlet so that the unit and receiver are on different branch circuits.
- Ensure that mounting screws and connector attachment screws are tightly secured.
- Ensure that good quality, shielded, and grounded cables are used for all data communications.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The following booklets prepared by the Federal Communications Commission (FCC) may also prove helpful:

- How to Identify and Resolve Radio-TV Interference Problems (Stock No. 004-000-000345-4)
- Interface Handbook (Stock No. 004-000-004505-7)

These booklets may be purchased from the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 About this Manual

This manual is intended to provide the user with an overview of the board and benefits, complete features specifications, and set up procedures. It contains important safety information as well.

1.2 Product Overview

The **TS-7300** Single Board Computer runs on a 200 MHz ARM9 processor with power under 2 Watt. Low board complexity, low component count, and low power/heat makes for an extremely reliable embedded engine. The TS-7300 SBC's are available in thousands of configurations, many of which are Commercial off the Shelf (COTS) and available to ship today.

The EP9302 processor from Cirrus is the highly integrated 200Mhz ARM9 processor that the TS-7300 SBC's are built around and includes an on-chip 10/100 ethernet, USB, serial, and Flash/SDRAM controller. A supplemental PLD provides glue logic, watchdog timer, Compact Flash IDE, and 8 bit PC/104 support. Integer CPU performance is about 20% faster than our 133 Mhz x86 offerings.

Even with the standard power consumption of 2 Watts, the TS-7300 SBC's run without fans or heat sinks in the temperature range of -20° to +70°C. Extended Temperature -40° to +85°C is also standard, but CPU clock must be decreased to about 166MHz for higher temperatures. Digital Signal Processing (DSP) is enabled through a standard 5 channel, 12bit A/D converter (Optional 8 channel, 12 bit A/D converter), 20 DIO lines and 2 standard serial ports.

The 8/16 bit PC/104 interface enables additional functionality through Technologic Systems' broad product line of PC/104 peripheral daughter boards. The TS-7KV adds video, CAN, Com Ports, and A/D conversion. The TS-ETH10 allows the addition of Ethernet ports. The TS-CAN adds CAN connectivity. The TS-Modem boards add both wired and cell phone capabilities.

The **TS-7300** is a compact, full-featured Single Board Computer (SBC) based upon the Cirrus EP9302 ARM9 CPU. The EP9302 features an advanced 200 MHz ARM920T processor design with a memory management unit (MMU) that allows support for high level operating systems such as Linux, Windows CE, and other. As a general-purpose processor, it provides a standard set of peripherals on board and a full set of Technologic Systems featured peripherals via the standard PC/104 Bus.

The **TS-7300** is a multipurpose board designed specifically for customers needing extreme design security, flexibility and reliability in applications such as gaming machines, building security equipment, or critical network infrastructure services such as network gateways or firewalls. Included on the board is a 8256 LUT Cyclone II FPGA that is reprogrammable on-the-fly by the 200Mhz ARM9 CPU running Debian Linux when an additional real-time soft-coprocessor(s), DSP, or specific additional peripheral logic is needed.

1.3 Benefits

Out-of-the-Box Productivity

Technologic Systems Linux products get you to your application quickly. Our Single Board Computers boot directly to Linux as shipped. There is no complicated CMOS setup or configuring of a Linux derivative Operating System to source, define, and load. Technologic Systems has pre-configured each SBC in flash memory.

The **TS-7300**'s user can power up the board and immediately begin application development. Of course, should you wish to configure your own version of Linux or use a different operating system, this is easy too. Technologic Systems provides the solution to fast application development without tedious OS configuration.

Impressive Performance

The ARM920T's 32-bit architecture, with a five-stage pipeline, delivers very impressive performance at very low power. The EP9302 CPU has a 16 KB instruction cache and a 16 KB data cache to provide zero-cycle latency to the current program and data, or they can be locked to guarantee no-latency access to critical sections of instructions and data. For applications with instruction-memory size restrictions, the ARM920T's compressed Thumb instruction set can be used to provide higher code density and lower Flash storage requirements.

As a benchmark, the **TS-7300**'s CPU integer performance, at a supplied 200 MHz, is about twice as fast as the Technologic Systems 133MHz 586-based products.

Linux FPGA

The inclusion of an Altera FPGA and a dedicated high-speed bus between the CPU and FPGA provides unique design possibilities in the hands of creative embedded designers. The FPGA has its own dedicated 8MB SDRAM and customers can load their own bitstream or use the default TS built bitstream that provides the hardware logic for the 8 serial ports, second 10/100 ethernet port, second SD card slot, VGA video and up to 35 DIO lines connected straight to the FPGA. These DIO lines can also be used as a FPGA connected expansion bus instead of the CPU connected regular PC104 expansion bus.

SD Card High-Security and Reliability

Unique security features on the **TS-7300** include the ability for the board hardware to first checksum the boot SD flash card in its entirety before executing a single instruction of code from it. One may also use the security features of a SD card to hardware lock and password protect the boot flash card so that the card is only bootable and readable on a specific **TS-7300** device. Any SD flash card can also be made read-only permanently and irreversibly once your product is finalized for additional design security.

The board also includes many features for high reliability computing and control. SD flash technology is already extremely reliable compared to hard drives, but with 2 hot-swappable flash card sockets running full-blown operating systems such as Linux, you have the option to use RAID on your flash devices to protect against potential individual flash write wearout failures. Also, every component on the board is rated for 70 or 80 degrees Celsius ambient temperatures without heat sinks or fans. Unlike typical "embedded" PC or laptop motherboards which can barely sustain thermal balance at room temperature with large heatsinks, the **TS-7300** operates with relatively huge margins in similar conditions which increases hardware lifetime and reliability.

1.4 Features

The **TS-7300** comes **standard** with these features:

- ✔ 200Mhz ARM9 with MMU and 32 MB SDRAM
- ✔ User-programmable Altera 2C8 Cyclone II **FPGA**
- ✔ 2 **SD Card** flash sockets
- ✔ **TS-VIDCORE** VGA video-out w/8MB dedicated video RAM
- ✔ 2 USB 2.0 Compatible OHCI ports (12 Mbit/s Max)
- ✔ 2 10/100 ethernet ports
- ✔ 10 RS232 serial ports

- ✔ 55 DIO ports (up to 35 **TS-XDIO** capable on FPGA)
- ✔ 1.8 watts power (CPU/SDRAM full speed, ethernets on but unplugged, all serial ports on, and with default FPGA bitstream)
- ✔ Fast boot to Linux prompt-shell in 1.69 seconds, Linux-based bootloader
- ✔ Watchdog timer
- ✔ PC/104 expansion bus
- ✔ SPI bus interface
- ✔ Matrix keypad and text mode LCD support
- ✔ Operating Temperature Range: Fanless from -20° to +70°C
- ✔ Extended Temperature -40° to +85°C standard at lower CPU clock speeds

1.5 Configurability

The **TS-7300** can be configured for your application using the following available on-board options and external accessories:

On-board Options

- ✔ **TS-7300-yyy**: TS-7300 with up to 128 MB of on-board SDRAM. For example, **TS-7300-64** selects model **TS-7300** with 64 MB of SDRAM.
- ✔ **OP-BBRTC**: on-board sealed-battery backed RTC
- ✔ **OP-TMPSENSE**: High-precision temperature sensor
- ✔ **OP-485-FD**: RS-485 full duplex interface on COM2
- ✔ **OP-485-HD**: RS-485 half duplex interface on COM2
- ✔ **OP-ROHS**: RoHS directive compliant built board

Note
The **TS-7300** SBC can be built compliant with the RoHS (Restriction of Hazardous Substances) Directive. Contact Technologic Systems for RoHS support.

External Accessories

- ✔ **SD-512**: 512 MB SD Flash Card with full ARM tool chain installed and Debian
- ✔ **USB Flash Drive**: 256 MB with full ARM tool chain installed and Debian
- ✔ **WIFI-G-USB:** Linux-supported USB 802.11g WiFi transceiver for wireless networking
- ✔ **OP-LCD-LED**: Alphanumeric 2x24 LCD with back light and cable
- ✔ **OP-KPAD:** Matrix keypad with cable
- ✔ **PS-5VDC-REG**: Regulated 5VDC Power Supply, 110 VAC Input
- ✔ **TS-ENC730**: Metal Enclosure with Power Converter and additional features
- ✔ **RC-DB9**: COM2 adapter cable to DB-9

In addition, a complete set of interfacing cables, connectors, and enclosures is available.

Note
Check our website at **www.embeddedARM.com** for an updated list of options and external accessories

1.6 PC/104 Peripherals

Technologic Systems offers many add-on peripherals to complete the requirements of your application. These products directly interface with the **TS-7300** using the PC/104 bus, adding a wide variety of functionalities at very reasonable prices. Some of these functionalities includes:

- **NVRAM** - adds 32K, 128K, 1MB or 2MB bytes of battery-backed SRAM. Battery backed SRAM provides non-volatile memory with very fast write times and unlimited write cycles, unlike Flash memory. This can be very important if the data is constantly being updated several times per minute, since Flash devices can wear-out after a few million write cycles. It also eliminates the latency that Flash memory has during write cycles. This resource is a byte-wide memory device using a lithium battery that will last a minimum of 10 years with or without power applied.
- Analog **VIDEO** interface – When a video monitor is needed.
- **CAN** Bus – Useful for automotive applications.
- **Modems** – Phone Line or GSM Cellular modems.
- Additional **Ethernet** ports.
- Additional **DIO** interface with either 24 or 64 new lines..
- Additional 12 bit **ADC** and **DAC**: useful for industrial automation applications.
- Additional **COM** and **Parallel** ports – Make more communication channels available.
- Power-over-Ethernet
- **Radios** – Long-range wireless radios, Xbee modules

Note
New PC/104 boards are always in development. Contact Technologic Systems or visit the **PC/104 peripherals page** at our website for a complete and updated list of additional functionalities that can be added to the **TS-7300** using the PC/104 bus. You can also contact Technologic Systems about your **custom project design**.

1.7 TS-ARM Development Kit

The TS-ARM Development Kit for the **TS-7300** Single Board Computer includes all equipment necessary to boot into the operating system of choice and start working. The development kit is highly recommended for a quick start on application development.

The TS-ARM Development Kit contains a 256 or 512 MB Flash drive (SD Card) which includes:

- ✔ a self-hosting ARM installation of the **Debian Linux** 3.0 distribution compiled for ARM
- ✔ gcc 2.95.4 and gcc 3.0 compiler with full tool-chain
- ✔ Hardware test routines source code and other example source code
- ✔ Debian package system: apt-get, tasksel, dselect

The development kit additionally includes:

- ✔ SD Card reader
- ✔ 5 VDC regulated power supply (international versions available)
- ✔ NULL modem cable
- ✔ Adapter cable from 10-pin header to DB9
- ✔ Various cables for connection DIO, LCD, Keypad, etc.
- ✔ Development CD with complete TS-Kernel source, manuals, example code, etc.
- ✔ Printed supporting documentation for TS-7300 Hardware, Linux for ARM and Development Kit.

Note
Single Board Computer is not included on the Development Kit (sold separately).

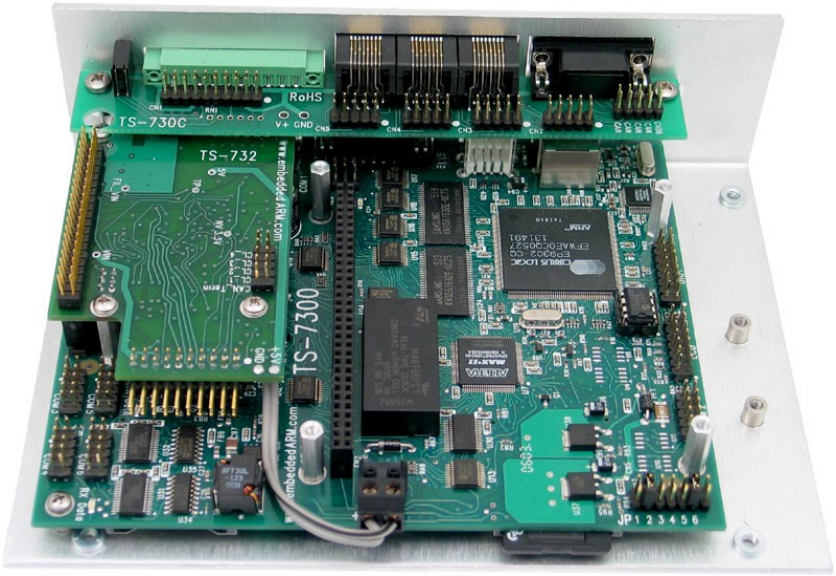
1.8 TS-ENC730 Enclosure

The TS-ENC730 metal enclosure is made to house the TS-7300 Single Board Computer and up to two PC/104 peripheral boards. The internal power regulator efficiently converts unregulated 7-28 VDC input into regulated +5 VDC required by the SBC. Sleep mode allows current drain of 200 microAmps with programmable sleep period.

The TS-ENC730 brings out some features implemented on the TS-7300 FPGA, such as the SJA1000 CAN core, PWM for the DAC lines, 800x600 VGA Video, second ethernet port, additional serial ports, etc.

Furthermore, the TS-ENC730 includes the TS-732 daughter board for the TS-7300 onboard FPGA. The AVR microcontroller on the TS-732 implements ADC/GPIO/LEDs extra functionalities. Also, the physical layer for the CAN bus is provided by the TS-732. Additional features include:

- ✔ 8-28 VDC power input
- ✔ User programmable low power sleep mode (300 uA)
- ✔ Surge suppression on power input
- ✔ Terminal strip with 8 I/O
  - ✔ 4 channel 10 bit ADC 0V - 5V (12-16 bits effective*)
  - ✔ 2 channel 12 bit high power DAC 0V - 10V (sync/source 300mA)
- ✔ 4 high current GPIO pins (sync/source 400 mA)
- ✔ 5-8 COM ports
- ✔ Two software controlled LEDs
- ✔ Status LEDs for both Ethernet ports
- ✔ Optional CAN bus port with Linux driver available
- ✔ Linux Application available for ADC/DAC/GPIO/LEDs control
- ✔ Sturdy metal design reduces noise



1.9 Software and Support

Technologic Systems provides:

- ✔ Free system software and documentation updates available on our web site
- ✔ Free technical support by phone, fax, or email
- ✔ 30-day, money back guarantee on evaluation units
- ✔ One-year, full warranty

Linux OS Support

The ARM processor (the EP9302) comes from Cirrus and the platform is very similar to the Cirrus EDB9302 evaluation board. Cirrus has strongly promoted running Linux on this chip and has done most of the legwork in creating a patch set to the Linux 2.4 kernels, but we have also had to modify the Linux Kernel (TS-Kernel) so it can support NAND nad NOR Flash chips (via mtd drivers), a compact flash IDE driver, A/D converters, SD Card through the TS-SDCORE, video, additional ethernet ports and more. If you want to use Linux and aren't tied to the x86 architecture, the TS-7300 can be very cost-effective.

The TS-7300 SBC's proprietary boot firmware makes it possible for the board to boot directly from the SD Card on the first socket, therefore the TS-7300's are shipped standard with the full-featured Debian Linux installed on the first SD Card. Debian can also be used with an NFS root file system or USB flash drives. The TS-Kernel used is based upon the version 2.4.26, patched and compiled for the Cirrus EP9302 ARM920T processor, and is real-time capable through RTAI.

The root file system used by the Linux OS can be any of the following:

- ✔ EXT2 file system image in the SD card
- ✔ JFFS/YAFFS file system image in the on-board Flash (on-board flash is a non-standard option on the TS-7300, please contact Technologic Systems)
- ✔ NFS root, via Ethernet port (after fast bootup, one need to mount a NFS root and chroot to it)

Note
The TS-Kernel supports the **Real-Time** Application Interface (**RTAI** project), making the embedded operating system capable of handling applications with hard real-time restrictions.

Other OS Support

The **TS-7300** can be loaded with other operating systems such as Windows CE, NetBSD, etc. Technologic Systems will provide support for these, and possibly other operating systems, in the future. Currently, only **Linux** and **NetBSD** are supported on the **TS-7300**.

2 GETTING STARTED & SOFTWARE

2.1 Installation Procedure

Before performing any set up or placement procedures, take the precautions outlined in this section.

Handling the Board Safely

Be sure to take appropriate Electrostatic Discharge (ESD) precautions. Disconnect the power source before moving, cabling, or performing any set up procedures.

> **Warning**
> Inappropriate handling may cause damage to the board.

Setup and Installation Instructions

Follow these guidelines for safety and maximum product performance:

- ✔ Observe local health and safety requirements and guidelines for manual material handling

Setup Tools

Depending on placement and cabling, you may need the following tools:

- ✔ Small flat-blade screwdriver
- ✔ Small Phillips screwdriver

Setup Procedure

After locating, setting up, grounding, and cabling the **TS-7300**:

- ✔ Apply power
- ✔ Monitor the **TS-7300** using a terminal emulator to verify that the board is operating properly

> **Note**
> Your board might include a screw power connector on the power input. Notice this connector is removible. Please pull this connectior off before applying power.

Disconnecting AC Power

- ✔ Unplug from the power source.
- ✔ Disconnect other cables as required.

2.2 Console and Power Up

Although the TS-7300 presents a VGA video output, the COM1 port is typically used as a console port to interface the TS-7300 to a standard terminal emulation program on a Host PC via a serial cable.

An ANSI terminal or a PC running a terminal emulator is required to communicate with your Embedded PC. Simply connect an ANSI terminal (or emulator) to COM1 (DB9 female connector) using a null modem cable (this is included in the TS-ARM Development Kit), using serial parameters of 115,200 baud, 8 data bits, no parity, no flow control, 1 stop bit, and make sure jumper **JP2** is installed. If you are running Linux, the minicom program works well, Windows users can run the Hyperterm application. Technologic Systems offers a null modem cable with both 25 pin and 9 pin connectors at each end as part number CB7-05. Some systems also require the 10-pin header to 9-pin Sub-D adapter

which is P/N: RC-DB9.

The console can be changed to COM2 by installing **JP4** (with JP2 also installed). If your application does not require a console or both COM ports are required, then removing the jumper **JP2** easily disables all console output and redirects it to VGA output.

Connect a regulated 5VDC, (1A minimum) power source using the included 2 screw terminal strip/connector. Please note the polarity printed on the board. The boot messages, by default, are all displayed on COM1 at 115200 baud.

Alternatively, one can also use a USB keyboard attached to the USB host port on the TS-7300 and a VGA monitor attached to the VGA video out on the TS-7300 in order to interface with the SBC. A Technologic Systems logo splash and a shell prompt to the root file system will be shown on the VGA display a few seconds after Ts-7300's power cycle. The console on COM1 will still be active though.

2.3 General Boot Sequence

The TS-7300 differs from the previous TS-72XX boards in the way it implements its bootloader. The TS-7300 does not make use of Redboot or any other traditional bootloader. Instead, the TS-7300 uses a proprietary firmware that enables fast boot to Liniux from the SD Card on the first socket.

Technologic Systems has written a binary Linux driver module and a set of generic, OS-independent read/write routines for accessing the SD flash inside of an ARM object (.o) file. The system boots to an initrd to load the SD driver as a module, since a pre-compiled kernel with the driver built-in could not be provided.

The on-board EEPROM contains code to talk to SD and the board actually boots directly to SD. The hand off from the bootrom to the SD card is very similar to x86 bootup. The first 512 bytes of the SD card (the MBR) is loaded into ram and jumped into. The code inside the MBR calls the BOOTROM routines to read more blocks from the SD card (similar to x86 DOS int 13h) and bootstrap itself.

Currently, our MBR code (Only 446 bytes available after partition table) loads in a kernel and an initrd from dedicated partitions on the SD card and starts Linux. The initrd contains the FPGA bitstream and the Linux driver modules for SD. These are loaded (along with modules for the ethernet and serial ports) and then the initrd does a pivot_root to the SD EXT2 file system and then frees the initrd. There is no on-board flash on the TS-7300, only SD.

The following is a complete description of the **TS-7300** bootup process:

1) Power-on. For approx. 200ms, both LEDs will be illuminated indicating the processor is receiving voltage but is being kept in reset waiting for power to stabilize. On-board MAX2 CPLD arms watchdog for 8 second expiry.

2) CPU begins executing mask-ROM bootup code. This code looks for the magic string "CRUS" in the SPI EEPROM chip. Red LED is now turned off.

3) CPU mask-ROM finds CRUS in SPI EEPROM and loads 2 kilobytes of EEPROM into on-chip SRAM. Processor then jumps into on-chip SRAM which now contains the TS-SDBOOT mini-loader. If the magic string "CRUS" is not found in the EEPROM (due to missing or malfunctioning SPI EEPROM chip) the processor locks up while fast blinking the green LED. 8 seconds later the CPLD watchdog will reset the board and try the whole process again.

4) TS-SDBOOT initializes SDRAM, bus timing, MAC addresses and performs self tests. If a major failure is detected, board disarms watchdog and turns off the green LED, turns on the red LED, and blinks off 3 times every 2 seconds.

5) TS-SDBOOT examines the 36 byte security block at the last 36 bytes of the EEPROM. Based on its contents, the board may:

- ✔ checksum and verify a predetermined number of sectors of SD flash
- ✔ verify a password typed to the console within 8 seconds of bootup
- ✔ verify a specific SD card serial number
- ✔ force booting only to a locked SD card
- ✔ unlock a locked SD card
- ✔ verify the CPU unique ID
- ✔ use the CPU unique ID to decrypt the SD unlock key
- ✔ use the typed password to decrypt the SD unlock key

6) After security processing, TS-SDBOOT loads the first 512 bytes of the SD card to physical SDRAM at address 0x1000. Control is transferred in ARM Thumb mode and two parameters are passed. In register r0 is the address to the TS-SDBOOT routine for reading blocks from the SD card and in register r1 a routine to print messages to the serial port is passed. The C stack pointer still is in on-chip SRAM. The C prototypes of these functions are as follows:

- ✔ void sdread(unsigned int start_sector, char *buffer, int num_sectors)
- ✔ void ser_puts(char *buffer)

If the SD boot sector contains bad code, the SD card is not present, or in the case of another unforseen problem with either the system, SD card, or the boot program that keeps the board from booting Linux, the watchdog will expire and reset the board in approx. 7.5 seconds.

7) The first 446 bytes of sector 0 contain Thumb program code that loads the first partition with id 0xda as a kernel image at RAM location 0x218000 and the next with id 0xda as a Linux initrd at 0x1000000. Linux ATAGs are composed and the kernel image is jumped into.

8) The kernel decompresses itself and initializes. The 8 second watchdog is now disabled. The compressed initrd is found, decompressed, and mounted as the root filesystem.

9) The kernel invokes the program /linuxrc on the initrd, which loads the TS-7300 FPGA and kernel modules for FPGA components. (serial driver, ethernet driver, SD card driver, and video driver) A splash screen is displayed on the VGA out.

10) As a last step, the /linuxrc code mounts /dev/sdcard0/disc0/part3 and pivots to this as the root filesystem and calls /sbin/init on the SD card filesystem. This starts either the Debian or the TS-Linux initialization sequence.

Technologic Systems provides binary dd images of a bootable SD card for download and also provide a utility to install just the MBR code on an arbitrary (pre-partitioned) SD card.

2.4 Fast Bootup Firmware

Through software optimizations, the **TS-7300** ARM SBC is able to bootup to a Linux prompt just 1.69 seconds after power-up. Technologic Systems provides a SD Card image that includes kernel tweaks, initrd tweaks and a few tweaks inside the Debian filesystem itself on partition 3. In addition, fast boot optimizations have also been applied to the TS-SDBOOT bootup firmware that resides in ROM memory.

The new SD image will look at the state of jumper 6. If jumper 6 is on, the full Debian bootup will be bypassed and the system will instead drop straight to a shell prompt. 1.69 seconds after power-on the serial console prompt is active and 2.41 seconds after power-on the video console is displayed. Video takes a bit longer to start up due to the fact that the FPGA must be initialized, a splash screen is displayed, and USB keyboard kernel modules must be loaded. To initiate a full Debian startup, simply type "exit" at either the shell prompt on the serial port or the shell prompt on the VGA monitor (using keyboard).

The time it takes for bootup is also displayed right before the SH prompt is printed. On new Rev C CPLD TS-7300's, the CPLD has a 32-bit counter that starts at 0 at poweron and is used to measure the bootup time extremely accurately. On other boards, the EP9302 983Khz debug4 timer is used since it starts out as 0 also, but is slightly less accurate than the 32-bit 14.7Mhz counter implemented in the new CPLD rev.

If you want something other than a shell prompt running as soon as possible on bootup, it is possible by editing the /linuxrc2 shell script on the initrd. When you do the fast boot, you are actually booting to an initrd with the Debian parts of filesystem mounted read-only. After modifying the /linuxrc2 shell script on the initrd, run the "save" command to save the initrd back to the SD card, otherwise your changes won't "stick".

The TS-7300 bootup firmware was designed to be much faster than a x86 PC/SBC BIOS. So even better bootup times can be achieved with an RTOS, such as eCos, instead of Linux. Of that 1.69s bootup time, about 1.4 represents Linux bootstrap, while about 0.3 of it is the hardware initialization.

Installation

If you board doesn't have the Fast Bootup firmware installed from factory, proceed as follow in order to have your TS-7300 upgraded.

The new SD Card image that enables the fast bootup option is available on our FTP site. Visit our website, TS-7300 page, for further information.

It requires at least a 256MB SD Card. To install it, follow the instructions below:

1. Extract the new SD Card image file, for example:

   $ bunzip2 fastboot-7300-sdcard-7-6-2006.dd.bz2

2. Using an SD Card reader and the "dd" utility, copy the image to the disc device of the 256MB SD Card:

   $ dd if=fastboot-7300-sdcard-7-6-2006.dd of=/dev/sda

   Or, using the second SD Card slot of your TS-7300:

   $ dd if=fastboot-7300-sdcard-7-6-2006.dd of=/dev/sdcard1/disc0/disc

> **!** **Warning**
> This will completely overwrite the contents of your SD Card, so make sure you do the appropriate backups before updating.

3. To use the image on a larger card (512MB, for example), just "dd" to the larger card, use fdisk to increase the size of part3 to the remaining space (taking care to keep the same start sector), and then use the "ext2resize" command on that partition:

   $ fdisk /dev/sdcard1/disc0/disc

   $ ext2resize /dev/sdcard1/disc0/part3

4. To update to the new TS-SDBOOT, download and run (as root) the TS-SDBOOT update utility binary executable on your TS-7300 board. This utility is found at our webiste.

2.5 Loading or Transferring Files

Three methods are available for transferring files between a desktop PC and your **TS-7300**: Ethernet downloads, flash memory devices, and Zmodem downloads. Full descriptions of each are detailed below. Other programs that use serial ports to transfer should work as well.

Transferring Files via the Ethernet Port

The default JFFS Linux root file system includes a small FTP server that can be used for uploading/downloading of files across an Ethernet network. Simply point your preferred FTP client to your **TS-7300** IP address (default is 192.168.0.50). You can login as root or any valid user previously created from the useradd utility. By default, the JFFS image will not accept anonymous FTP.

Transferring Files via Flash Memory Device

The **TS-7300** removable Compact Flash card, an SD card or an USB flash memory card can be used to easily move files from a host system. We suggest using a low-cost SanDisk USB Compact Flash card or SD card interface for your host system. USB memory devices need no extra accessory to connect to the host PC. The flash memory devices can then be hot swapped (inserted or removed without rebooting the host PC).

Zmodem Downloads

Using the Zmodem protocol to send files to and from the **TS-7300** SBC 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-7300** from your host PC, execute lrz at the Linux command line on the TS-7300 (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-7300** to your host PC, execute lsz <FILENAME> at the Linux command line on the **TS-7300** 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 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.

2.6 SD Card High-Security Features

Technologic Systems has developed a Linux application named "sdlock" which can be used to manipulate SD card hardware-enforced password locks and set the card's permanent write-protect feature. Using a password protected SD card is a great way to ensure software security and/or to make sure your **TS-7300** SBC based product cannot be used in an unintended matter once deployed. This solution is only available for the products configured with the TS-SDBOOT firmware. Some of the possibilities include:

- ✔ Password protecting SD Cards
- ✔ Set the SBC to boot only locked SD Cards
- ✔ Set the SD Card readable only on a specific SBC
- ✔ Checksum verification of bootable SD Cards
- ✔ Make an SD Card permanently write-protected

How to use it

Usage and command line help for this command follows:

$ sdlock –help
Usage: sdlock [OPTION] ...
Controls SD card lock and permanent write-protect features.

General options:
-p, --password=PASS Use PASS as password
-c, --clear Remove password lock
-s, --set Set password lock
-u, --unlock Unlock temporarily
-e, --erase Erase entire device (clears password)
-w, --wprot Enable permanent write protect
-h, --help This help

The "sdlock" Linux utility is available for download at our FTP server. Visit the TS-7300 page for the latest download links.

When the **TS-7300** SBC is configured with the TS-SDBOOT bootup firmware, the SD unlock password can be stored in onboard EEPROM for automatic unlocking and booting of password protected SD cards. By default, TS-SDBOOT will still boot unlocked cards, but this behavior can be changed with the "--verifylock" option to the "tsbootrom-update" command described above-- with the "--verifylock" option the TS-7300 SBC will only boot locked SD cards.

TS-SDBOOT firmware contains several features for high security. One feature is the ability to store a checksum of the SD card on the board to verify before bootup. If the checksum fails, the bootup firmware will refuse to boot the inserted SD card. TS-SDBOOT can also verify an arbitrary number of sectors of the SD flash card before allowing bootup. If the stored CRC does not match the actual CRC, the board will refuse to boot and blink the red LED continuously.

Another feature is the ability to boot a password protected SD card. With this, it is possible to make an SD unreadable to any device except the **TS-7300** SBC to which it is assigned. Although not directly a function of TS-SDBOOT, an SD card can also be made permanently write-protected through a software command. The combination of these features allows product designers several options on the security of their software and of their deployed **TS-7300** SBC based devices.

The various SD commands that manipulate the password lock are marked as "optional" in the SD card specification. This means that not all SD card vendors may implement them in their devices. If they are not implemented, you will not be able to set the SD lock with the "sdlock" command.

For further information, contact a Technologic Systems' engineer.

2.7 Debian Linux OS

The **TS-7300** SBC is shipped with full Debian distribution installed on the SD Card and boots Debian Linux by default. The Debian is a full-featured and powerful Linux distribution mostly based upon GNU tools. It includes everything necessary to easily run Linux and develop Linux applications. In addition, various original Linux utilities and installation tools are available to make system utilization and packages updating easy. The Debian Linux is ported to ARM processors and runs with TS-7300 SBCs. With Debian Linux, experienced Linux users have a complete Linux system to take full advantage of their knowledge, and new users have an easy environment to get started with the Linux world.

Technologic Systems makes use of Debian Linux as a development distribution. For example, the 256 MB SD Card card is pre-installed with Debian. Along with the basic core utilities, some developer tools have been installed, including a native arm gcc toolchain, for C/C++ application development. In addition, the Perl interpreter and a wide variety of network services are available, such as FTP, Telnet, SSH and Apache HTTP servers. One can also use the Debian Linux distribution via an NFS root file system or an USB flash memory device.

apt-get

When using the Debian Linux file system, adding new packages and removing undesired ones is done all through Debian's package management. “apt“, “dpkg“, all behave as expected. With Debian, one can easily install and remove software packages. For a quick demonstration of how easy it is to remove and install programs with Debian, try the following commands:

```
apt-get install hexedit
hexedit /etc/passwd
^C (hit CTRL+C to safely exit)
apt-get remove hexedit
```

apt-get install installs a package name, while apt-get remove removes the named package. Visit the Debian home-page for further information, since a full in-depth discussion on Debian is outside the scope of this document.

- http://www.debian.org

Logging In and Basic Commands

After the desired Linux Kernel is loaded and executed through RedBoot, the file system loads and networking, logging, Apache web server, etc. are all started. When the login prompt is displayed, type “**root**” to login, with no password. A Bash login prompt will then appear. At this point, you are ready to enjoy your TS-7300 SBC running Linux. Some very basic commands for one beginner user to start using Linux are:

- pwd: informs the current directory
- ls: lists current directory contents
- cd: changes directory
- man: accesses the system's manual pages of a given command
- cat: displays the entire content of a given file
- vi: Linux most common file editor (reading further documentation is recommended)

The most common file handling commands are “cp”, “mv”, “rm”, “mkdir”. Help information is provided by supplying “--help” to any given command, for example “cp –help”

Shutdown

Use the “shutdown –h now“ command to halt the Linux system when running from Compact Flash, SD or USB memory card to avoid a potentially lengthy file system check on the next boot, since the file system running is EXT2 formatted.

On the other hand, the JFFS/YAFFS file systems are highly tolerant of power cycles while the file systems are mounted. Therefore, the “shutdown” command is not required when the root file system is JFFS/YAFFS, but is still recommended.

Initialization Scripts

The initialization process reads the file "/etc/inittab". The inittab file will call "/etc/rc.d/rcS.sysinit" as part of the system initialization. The run level then defaults to 3, which will run the "/etc/rcS" script and call all the scripts linked in the "/etc/rc3.d/" directory in numerical order. For example, the following are the initialization scripts for run level 3 found at TS-Linux:

```
/etc/rc.d/rc3.d# ls
S10Network  S11portmap  S20inetd  S30telnetd  S40apache
```

Changing the run level or re-invoking the initialization scripts is possible through the "init" command. A "halt" or "reboot" command will change the run level to 0 or 6 and execute the "/etc/rc0.d" scripts or "etc/rc6.d" scripts respectively.

Network Setup

The main utilities for network configuration under Linux are:

- ✔ ifconfig: prints network settings and configures ethernet interfaces
- ✔ ifup: turns given network interface up
- ✔ ifdown: turns given network interface down

Entering "ifconfig" shows the current ethernet settings. These utilities require a network device as parameter. On Linux, the ethernet devices are generally named eth0, eth1, etc. Therefore, the command "ifup eth0" or "ifconfig eth0 up" brings up the on-board ethernet interface on TS-7300 SBCs.

Setting Up the networking with Debian

To configure the network interfaces when booting into Debian Linux, edit the file "/etc/network/interfaces". A typical interfaces file would contain the following:

```
auto lo eth0
# The loopback interface
iface lo inet loopback
# The first network card
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.0.50
netmask 255.255.255.0
gateway 192.168.0.1
```

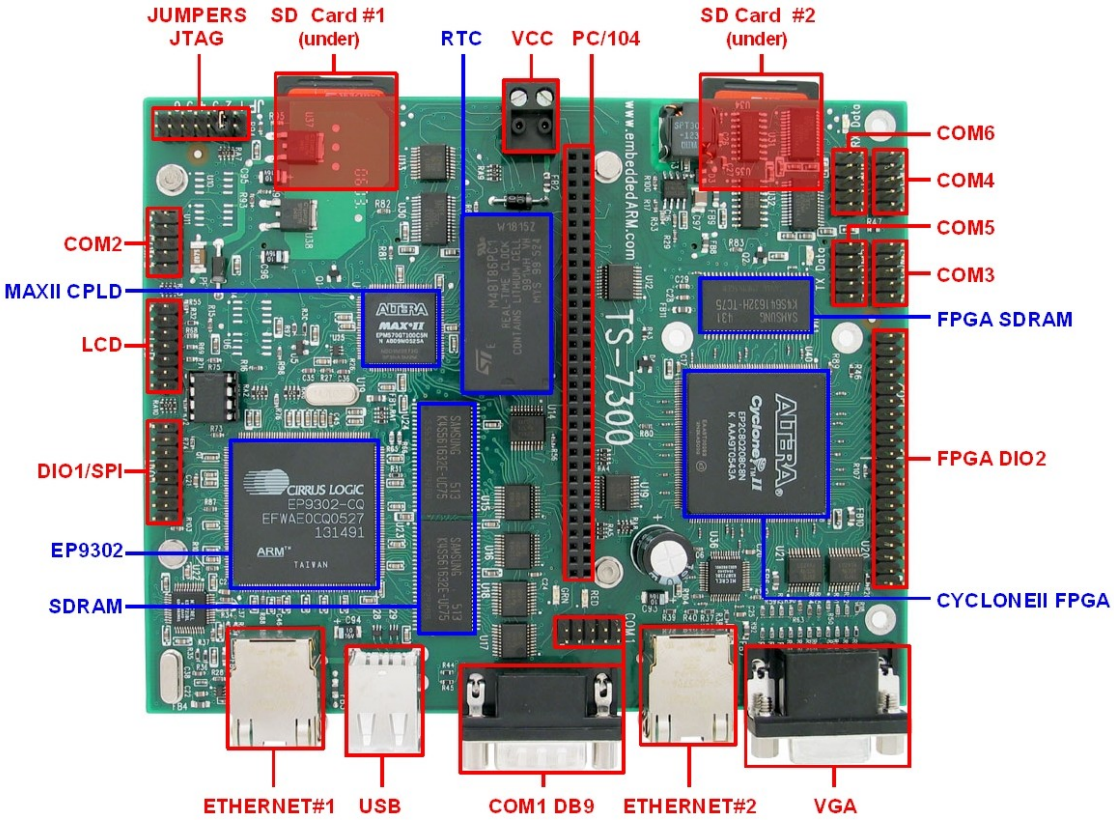
Those lines starting with a # symbol are comments. The line "auto lo eth0" means both the loopback interface and the first ethernet interface will be started automatically by the Debian networking scripts. The above example shows that eth0 would be assigned the static address of 192.168.0.50, using 192.168.0.1 as the default gateway. If one was to comment out those lines, and then uncomment the line iface eth0 inet dhcp, then eth0 would use a dhcp client to obtain it's IP and other relevant network information.

Note
For further information regarding the software solutions available for the **TS-7300** and instructions about Debian Linux, please refer to the **Linux for TS-ARM User's Guide**. This can be found for download at Technologic Systems website.

3 HARDWARE COMPONENTS

The following picture shows where the main headers, connectors and most important hardware components are located on the **TS-7300**. Understanding this picture will help you to follow the header-connector oriented organization of this manual. The blue marked objects on the picture are the on-board chips and components, while the red ones are the various on-board headers and connectors for peripherals.

Picture: TS-7300 Hardware Components



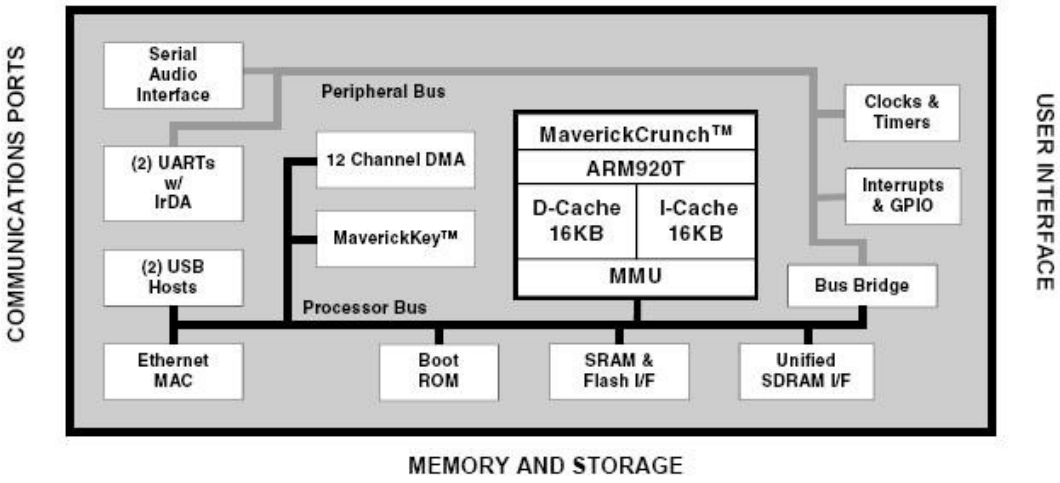
3.1 Processor

Cirrus EP9302

The EP9302 features an advanced 200 MHz ARM920T processor design with a memory management unit (MMU) that allows support for high-level operating systems such as Linux, Windows CE, and other embedded operating systems. The ARM core operates from a 1.8 V supply, while the I/O operates at 3.3 V with power usage between 100 mW and 750 mW (dependent on speed). As a general-purpose processor, it provides a standard set of peripherals on board and a full set of Technologic Systems add-on peripherals via the standard PC/104 Bus.

The ARM920T's 32-bit architecture, with a five-stage pipeline, consisting of fetch, decode, execute, memory, and write stages, delivers very impressive performance at very low power. The EP9302 CPU has a 16 KB instruction cache and a 16 KB data cache to provide zero-cycle latency to the current program and data, or they can be locked to guarantee no-latency access to critical sections of instructions and data. For applications with instruction-memory size restrictions, the ARM920T's compressed Thumb instruction set can be used to provide higher code density and lower Flash storage requirements.

Picture: Cirrus EP9202 Block Diagram



EP9302 key features include:

- ✔ ARM (32-bit) and Thumb (16-bit compressed) instruction sets
- ✔ 32-bit Advanced Micro-Controller Bus Architecture (AMBA)
- ✔ 16 kbyte Instruction Cache with lockdown
- ✔ 16 kbyte Data Cache (programmable write-through or write-back) with lockdown
- ✔ MMU for Linux®, Microsoft® Windows® CE and other operating systems
- ✔ Translation Look Aside Buffers with 64 Data and 64 Instruction Entries
- ✔ Programmable Page Sizes of 1 Mbyte, 64 kbyte, 4 kbyte, and 1 kbyte
- ✔ Independent lockdown of TLB Entries

For further information about the EP9302 features, refer to the **EP9301 User's Guide**.

Note
The EP9302 is identical silicon to the EP9301 except it is rated to run at 200 Mhz, instead of 166 Mhz. The available **EP9301 User's Guide** can still be used as the main reference manual.

MMU

The EP9031 features a Memory Management Unit, enabling high level operating systems such as Embedded Linux and Windows CE to run on the **TS-7300**. In the same way, the Linux TS-Kernel takes advantage of the MMU functionality.

The MMU is controlled by page tables stored in system memory and is responsible for virtual address to physical address translation, memory protection through access permissions and domains, MMU cache and write buffer access. In doing so, software applications can access larger "virtual" memory space than the available physical memory size, allowing multiple programs to run and use the system memory simultaneously.

For further information about the MMU functionalities, refer to the EP9301 User's Guide.

Interrupts

The EP9302 interrupt controller allows up to 54 interrupts to generate an Interrupt Request (IRQ) or Fast Interrupt Request (FIQ) signal to the processor core. Thirty-two hardware priority assignments are provided for assisting IRQ vectoring, and two levels are provided for FIQ vectoring. This allows time critical interrupts to be processed in the shortest time possible.

Internal interrupts may be programmed as active high or active low level sensitive inputs. GPIO pins programmed as interrupts may be programmed as active high level sensitive, active low level sensitive, rising edge triggered, falling edge triggered, or combined rising/falling edge triggered.

The EP9302 interrupt controller also includes the following features:

- ✔ Supports 54 interrupts from a variety of sources (such as UARTs, GPIO and ADC)
- ✔ Routes interrupt sources to either the ARM920T's IRQ or FIQ (Fast IRQ) inputs
- ✔ Three dedicated off-chip interrupt lines operate as active high level sensitive interrupts
- ✔ Any of the 19 GPIO lines maybe configured to generate interrupts
- ✔ Software supported priority mask for all FIQs and IRQs

Note
For peripheral driver development purpose, notice that the external IRQ lines 5,6 and 7, which are ISA/X86 architecture based, are mapped to EP9302 external interrupt lines 22, 33 and 40, respectively. For further information about interrupts, including the EP9302 interrupt controller and map, refer to the **EP9301 User's Guide**, chapter 5.

Warning
Since the TS-7300s FPGA uses IRQ7, any PC/104 daughter board that uses IRQ7 must be open drain in order to allow IRQ sharing, otherwise there will be an IRQ conflict. For instance, the TS-SER4 is not compatible with the TS-7300 when configured to use IRQ7 because the TS-SER4 actively drives the IRQ high/low.

3.2 Memory

TS-7300 uses three type of memory. The SDRAM is the fast access volatile memory used to run applications by the processor and the on-board flash is the non-volatile memory used for storage purpose. Flash memory may also be added using USB memory drivers.

On-Board SDRAM

The **TS-7300** uses 32 MB SDRAM technology to provide 32 or 64 MB of high-speed volatile memory. The memory is soldered directly to the board, making the **TS-7300** more reliable in high-vibration environments.

The **TS-7300's** RAM is not contiguous in the physical memory map of the EP9302. But the MMU is programmed to remap the blocks of RAM to appear as a contiguous block of memory at the very beginning of the virtual memory map. In the case of a 256 Megabit SDRAM chip (32 MB), it is located at 0 through 32 MB in the virtual memory map.

Refer to the MMU section of this manual to understand how the physical memory is mapped and the virtual memory is translated.

Note
It is possible to use larger sizes of the SDRAM chip than the standard 32 MB one. The **TS-7300** is designed to accommodate both 32 MB and 64 MB chips, providing up to 128 MB of RAM memory. Contact Technologic Systems for larger SDRAM sizes.

Battery Backed SRAM

There is a peripheral board available for the **TS-7300** named TS-NVRAM that adds 32K bytes or 128 Kbytes or 512K bytes of battery-backed SRAM. Battery backed SRAM provides non-volatile memory with very fast write times and unlimited write cycles, unlike Flash memory. This can be very important if the data is constantly being updated several times per minute, since Flash devices can wear-out after a few million write cycles. It also eliminates the latency that Flash memory has during write cycles, since Flash technology write cycles are about 10-100 times slower than read cycles.

The TS-NVRAM peripheral board is located at the PC/104 memory space base address of 0x11AA_0000. This resource is a byte-wide memory device using a lithium battery that is guaranteed to last a minimum of 10 years with or without power applied.

On-Board NAND Flash

Warning
The **TS-7300** was designed to accommodate NAND flash chips. However, it is not a standard feature and the NAND chip is not soldered on the board by default. To include NAND support on the **TS-7300**, contact Technologic Systems.

The **TS-7300** uses a NAND Flash chip for its on-board Flash resource. The physical address of the Flash chip is **0x6000_0000**. The first 16KB is reserved for the TS-BOOTROM code. The TS-BOOTROM code initializes various internal configuration registers for proper operation, and initializes and tests the SDRAM. The last 3 MB are reserved for the RedBoot ROM monitor, RedBoot FIS (Flash Image System) and RedBoot FCONFIG (Flash configuration).

The Linux kernel shipped by default is pre-loaded in the FIS and the default boot script and Ethernet MAC address are contained in the FCONFIG. You may also use the RedBoot FIS to store and load images that contain eCos applications or other OS/RTOS boot loaders.

The rest of the on-board flash is used for the Linux YAFFS2 file system. This is a journaling file system that is aware of the wear-out mechanism of the NAND flash and incorporates ECC algorithms at the file system level to maximize Flash lifetime. It is also extremely tolerant of power failures during file write sequences.

The entire Flash chip can be write-protected by removing Jumper 3. When JP3 is not installed, the Flash chip becomes a read-only resource.

Note
It is possible to use larger sizes of the NAND Flash than the standard 32 MB chip. The **TS-7300** is designed to accommodate both 32 MB and 128 MB chips, providing up to 256 MB of on-board flash. Contact Technologic Systems for larger Flash sizes.

Note
The YAFFS1 file system runs on the **TS-7300** boards that feature NAND chips with 512 byte page size, enabling up to 128 MB of on-board flash. The YAFFS2 file system supports the new NAND technology, with 2k page size, hence it will be installed on **TS-7300** boards that are configured with 128-256 MB of flash.

USB Flash Drive or Compact Flash Card

Additional non-volatile storage may be added with a USB flash drive or a Compact Flash card. These devices supply additional non-volatile storage either for data or for a complete operation system distribution, such as Debian. A tar-file of Debian is available on the Technologic Systems website. Alternatively, the developer's kit includes a USB flash thumb-drive or Compact Flash card pre-loaded with Debian.

Flash memory provided by these devices behaves much as a hard drive does with sizes ranging from 32MB to 1GB. These products are inherently more rugged than a hard drive since they are completely solid-state with no moving parts. However, they have the added advantage of being removable media

Use of a Compact Flash card with TS-7300 SBC or higher requires a USB Compact flash adapter, which will also be included in the TS-ARM Development Kit if requested. The USB flash drive has the advantage over a CF card in that the USB drive can be hot swapped.

Note
Drivers are available in the TS-Kernel to support USB flash drives. One can load Debian OS with two scripts provided by the on-board flash TS-Linux file system or available for download at our website. First, invoke /usr/bin/loadUSBModules.sh, then run the script /usr/bin/loadUSB.sh to chroot into the Debian OS.

SD Memory Card

Technologic Systems has a full license for using the additional SD features which are reserved for members of the SD Card Association. This has allowed us to design both the hardware logic core and software specifically tuned to the capabilities of the **TS-7300** CPU using the official SD specification documents. Since both a Linux driver module and an ARM9 object file containing OS-independent access routines are provided to customers purchasing the board hardware, customers do not have to seek SD licensing themselves.

SD Memory Card technology provides large capacity and fast access combined with a compact and slim profile, making it very appealing for a wide range of next generation products and applications. In addition, SD Cards feature content protection, planned capacity growth, high-speed data transfer, and a write protect switch. These devices supply additional non-volatile storage either for data or for a complete operation system distribution, such as Debian, to be used with the **TS-7300** SBC**.**

The Technologic System SD Card core is a very small implementation and can be integrated on the **TS-7300** CPLD and/or the TS-7300 FPGA. Four 8-bit registers are available for the software layer to control the SD Card hardware:

Table: SD Card core registers

I/O Addr	Name	Description
BASE + 0	SDCMD	SD Command register
BASE + 2	SDDAT	SD Data register
BASE + 4	SDSTATE	SD State register
BASE + 6	SDCTRL	SD Control register

FPGA SDRAM

The FPGA has its own dedicated 8MB SDRAM that might be used, for instance, as video memory for the TS-VIDCORE VGA core. The physical address of the FPGA SDRAM chip is **0x7220_0000**.

3.3 Glue Logic CPLD

The TS-7300 ARM SBC's include a MAXII CPLD which is responsible for taking control over the internal components communication through glue logic implementation. For instance, the CPLD is used to control the NAND flash through internal registers configuration.

The CPLD handles control signals on the PC104 bus, has a watchdog timer, enables jumper settings reading, handles the reset button, interfaces to the real-time clock and controls the EEPROM chip select. It also implements peripheral features that, together with EP9302 modules, makes available an advanced set of communication ports, DIO pins, ADC converters, and others.

The inclusion of a CPLD on the SBC allows customized programming for customers with special needs, without having to do a more expensive board redesign. For example, the MAXII CPLD on the TS-7260 can be configured with three different cores:

- ✔ 2TTLCOM option: 2 extra TTL-only serial ports with TX enable signals and that includes a very simple GPIO core (data direction register and data register only).
- ✔ TS-XDIO option: uber-GPIO that can do quadrature, PWM, freq-counter, pulse timing, IRQ and DRQ, etc
- ✔ SDSOCKET option: a special core for a SD interface that requires a special Linux driver module to be of use.

The CPLD can be programmed using the JTAG header and special software/hardware supporting tools. Contact Technologic Systems for support on CPLD programming software and tools.

3.4 Cyclone II FPGA

The inclusion of an Altera FPGA on the **TS-7300** and a dedicated high-speed bus between the CPU and FPGA provides unique design possibilities in the hands of creative embedded designers. The FPGA has its own dedicated 8MB SDRAM and customers can load their own bitstream or use the default TS built bitstream. The FPGA is re-programmable on-the-fly by the 200Mhz ARM9 CPU running Debian Linux when an additional real-time soft-coprocessor(s), DSP, or specific additional peripheral logic is needed. In addition, there are up to 35 DIO lines connected straight to the FPGA. These DIO lines can also be used as a FPGA connected expansion bus instead of the CPU connected regular PC104 expansion bus.

The default full-featured FPGA bitstream from Technologic Systems (TS-BITSTREAM) provides the hardware logic for the TS-SDCORE second SD card core, 8 TS-UART serial ports, second 10/100 ethernet port, TS-VIDCORE VGA video framebuffer and accelerator, and 2 PWM/Timer/Counter TS-XDIO cores for the various GPIO pins.

The FPGA can be programmed through two registers found at base **0x13C0_0000** and **0x13C0_0001**. This hardware functionality is implemented on the CPLD and can be accessed via software. To load the bitstream to the FPGA on the TS-7300, Technologic Systems provides a Linux program "load_ts7300" that takes the ts7300_top.rbf file generated by Altera's Quartus II on the Linux flash filesystem (yaffs, ext2, jffs2, etc) and loads the FPGA. Loading the FPGA takes approx 0.2 seconds this way and can be done (and re-done) at any time during power-up without any special JTAG/ISP cables.

The FPGA is located at the word address space from **0x72A0_0000 to 0x72FF_FFFC**of the EP9302 memory map. The EP9302 **IRQ #40** (ISA IRQ 7) is used by the FPGA to interrupt the processor. In Linux, be sure to use request_irq() with the SA_SHIRQ flag, enabling sharing interrupts among other Linux device drivers.

The TS-BITSTREAM binary comes with board. Contact Technologic Systems for custom FPGA development on the TS-7300 or for non-GPL licensing of this or any of the above TS-cores and OS drivers.

3.5 Real-Time Clock

The **TS-7300** optionally supports a Non-volatile Battery-backed real-time clock (RTC) which is soldered onto the board. This option uses an ST Micro M48T86PC1 module for the real-time clock function. This module contains the lithium battery, 32.768 kHz crystal, and a RTC chip with 114 bytes of battery-backed CMOS RAM. It will maintain clock operation for a minimum of 10 years in the absence of power.

The 114 bytes of non-volatile RAM, physically located in the RTC chip, are available to the user. Contact Technologic Systems for driver support.

The RTC is accessed using two registers. The write-only index register is located at physical address location **0x1080_0000** and the RTC data register is location at physical address location **0x1170_0000**. These are byte-wide registers with the Index Register property of write only. The Data Register has a read/write property. Valid Index Register values are between 0 and 127, decimal. The first 14 index locations are used for accessing the RTC Time and Date registers. The next 114 locations are non-volatile RAM locations.

This option is NOT compatible with the TS-5620, a peripheral board that also uses an ST Micro RTC module for real-time clock functionality. While the two options are mutually exclusive, it is possible to use the TS-5620 peripheral board on a **TS-7300** that does not have the on-board RTC option installed. Any source code that utilizes the RTC is compatible with both optional installations. The TS-Kernel shipped with the boards includes support for the TS-5620 peripheral board.

3.6 Watchdog Timer

The **TS-7300** incorporates a Watchdog Timer (WDT) unit. The WDT can be used to prevent a system "hanging" due to a software failure. The WDT causes 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 specific value to a specific memory location.

Table: Watchdog Control Registers

Register	Address	Access
WDT Control register	0x2380_0000	Read/Write
WDT Feed register	0x23C0_0000	Write Only

The WDT Control register must be initialized with the timeout period desired. This may be as short as 250 mS or may be as long as 8 seconds. After the WDT has been enabled, the WDT counter begins. The application software can reset this counter at any time by "feeding" the WDT. If the WDT counter reaches the timeout period, then a full system reset occurs.

Table: Watchdog Timeout Register

Value	MSB	MID	LSB	Timeout Period
0x00	0	0	0	Watchdog Disabled
0x01	0	0	1	250 mS
0x02	0	1	0	500 mS
0x03	0	1	1	1 second
0x04	1	0	0	-- Reserved
0x05	1	0	1	2 seconds
0x06	1	1	0	4 seconds
0x07	1	1	1	8 seconds

In order to load the WDT Control register, the WDT must first be “fed”, and then within 30uS, the WDT control register must be written. Writes to this register without first doing a “WDT feed”, have no affect. In order to clear the WDT counter (feeding the watchdog), a value of Hex 05 must be written to the WDT Feed register.

By default, a user process does not have the physical address space (access) of the watchdog registers mapped. When using the Linux OS, the watchdog can be reached from user C code by using the mmap() system call on the /dev/mem special file to map the areas of physical address space into process user address space.

Warning
Use only the Watchdog Timer implemented by Technologic Systems in the CPLD. The Watchdog Timer included in the EP9302 has serious problems.

4 COMMON INTERFACES GENERAL INFORMATION

The purpose of this section is to provide general information about the common interfaces, such as Serial Ports and Digital Input/Output, which appear in more than one header or connector of the **TS-7300**. For further information on these features, refer to the Connectors and Headers section of this manual.

4.1 Serial Ports

The **TS-7300** have two asynchronous serial ports (COM1 and COM2) which provide a means to communicate with external serial devices. Each is independently configured as a 16C550- type COM port that is functionally similar to a standard PC COM port. These ports have 16-byte FIFOs in both the receive and the transmit UART channels. Both COM ports can support all standard baud rates up through 230.4Kbaud. Both COM ports may be configured to use a DMA channel (useful when very high baud rates are being used).

COM1 and COM2 UARTs can generate:

- ✔ Four individually maskable interrupts from the receive, transmit, and modem status logic blocks
- ✔ A single, combined interrupt that is asserted if any of the individual interrupts are asserted and unmasked

The COM1 port can also support the HDLC protocol. Refer to the Cirrus EP9301 User's Guide for more details. The COM2 port can optionally support RS-485 half or full duplex levels.

The **TS-7300** has either six or ten total asynchronous serial ports and the capability to add more on-board via the PC/104 Expansion Bus.

Two serial ports are standard on-board and are available through the COM1 (DB9 and 10-pin header) and COM2 (10-pin header) interfaces. They use RS232 signal level and 16 bytes FIFO. The minimum and maximum baud rate are 110bps and 230.4Kbps, respectively. In addition, COM2 is RS-485 capable.

Four RS-232 COM ports are integrated on the FPGA and are available through the COM3, COM4, COM5 and COM6 10-pin headers by default. Extra four COM ports from the FPGA may be provided using the same headers. Each is independently configured and is functionally similar to a PC COM port. ALL FPGA COM ports can operate in 8-bit mode or 9-bit mode. The minimum and maximum baud rate are 2400bps and 115.2Kbps, respectively.

4.2 Digital I/O

There are 55 total Digital Input/Output (DIO) lines available on the **TS-7300** . These are available on three headers labeled “DIO”, “LCD” and “DIO2”. The header labeled LCD can be used as 11 DIO lines or as an alphanumeric LCD interface. The header labeled DIO has 8 DIO pins available. In addition to the DIO signals, each header also has a power pin and Ground available. The LCD header has 5V power available while the DIO header has 3.3V power.

The DIO2 header (FPGA DIO) is controlled by the on-board FPGA. It is a 40-pin header divided in two sub-headers of 20 pins each. On the first header, by default, there are 17 video signals that can be changed into digital I/O, while the second, labeled DIO2, has 18 digital I/O lines which implement two XDIO ports. Thus, there are 35 total DIO lines connected directly to the on-board FPGA on the **TS-7300** available through the 40-pin header. The first header has 5V power available and the second (DIO2) has 3.3V power.

Three pins on the DIO header are used to bring out the EP9302 SPI bus. By using some of the DIO pins as peripheral Chip Select signals, a complete interface is available for SPI peripherals. It is also possible to bring out a fourth SPI bus function [SPI_Frame] by adding a 10 ohm resistor in the position labeled R1 on **TS-7300** boards. This signal is not required for many SPI peripherals but it may prove useful in some applications.

All of the DIO lines are programmable as either inputs or outputs and the direction of each I/O pin can be individually programmed. All DIO control registers are 8-bits wide and aligned on word (32-bit) boundaries. For all registers, the upper 24 bits are not modified when written and are always read back as zeros. Every DIO pin has two registers used to access it, an 8-bit data register and an 8-bit data direction register (DDR). The DDR controls whether each DIO pin is an input or an output (”1” = output). Writing to the data register only affects pins that are configured as outputs. Reading the data register always returns the state of the DIO pin.

Many of the DIO pins (DIO_0 thru DIO_8 and LCD_0 thru LCD_7) can also be programmed to cause interrupts. The interrupt properties of these pins are individually configurable. Each interrupt can be either high or low level sensitive or either rising or falling edge triggered. It is also possible to enable debouncing on any of these interrupts. Debouncing is implemented using a 2-bit shift register clocked by a 128 Hz clock. Refer to the **EP9301 User's Guide** and see Chapter 21 for details

The 5V power output pin on the LCD Header is current limited to 1000mA to prevent excessive current. The 3.3V power on the DIO1 Header is current limited to approximately 1 Amp by a 3.3V regulator.

4.3 A/D Converters

The EP9302 A/D converter is standard on all **TS-7300** boards**.** The Cirrus EP9302 features a 5 channel, 12-bit Analog to Digital Converter with an analog multiplexer, with an input range of 0 to 3.3 V.

The **TS-7300** uses 4 of the 5 EP9302 A/D channels for on-board measurements, therefore only the ADC4 channel is available for the user on the DIO1 header.

5 CONNECTORS AND HEADERS

5.1 10/100 Base-T Ethernet Connector

The EP9302 Ethernet LAN controller incorporates all the logic needed to interface directly to any MII compatible Ethernet PHY chip. A low-power Micrel KS8721 chip is used to implement the Ethernet PHY function and an integrated RJ-45 connector with built-in 10/100 transformer and LED indicators completes the Ethernet sub-system.

The **TS-7300** 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-7300** is powered and properly connected to a 10/100BaseT Ethernet network. The LINK/ACTIVITY LED will blink to indicate network activity for either inbound or outbound data. The SPEED LED (right side of connector, amber) will be on when a 100Mb network is detected and off for a 10Mb network. Both of these LEDs are controlled by the KS8721 and do not require any overhead by the processor.

The Ethernet PHY chip can be powered down, under software control, to save approximately 90 mA of current consumption. This is controlled by the EP9302 Digital output on Port H, bit 2. A logic zero will power down the KS8721 PHY interface.

Note
TS-Kernel provides all the software support to use the EP9302 10/100 Ethernet core. For more details, find the TCP/IP configuration instructions on the Linux documentation.

5.2 Second 10/100 Base-T Ethernet Connector

The default TS-BITSTREAM core for the **TS-7300** on-board FPGA implements an adaptation of the OpenMAC core from OpenCores.org. This makes available a second 10/100 base-t ethernet port on the **TS-7300**. As the default EP9302 ethernet LAN controller, a low-power Micrel KS8721 chip is used to implement the Ethernet PHY function and an integrated RJ-45 connector with built-in 10/100 transformer and LED indicators completes the Ethernet sub-system.

The second ethernet is about 15%-20% slower than the default high-speed MAC core inside the EP9302 system-on-chip. The physical memory location of the OpenMAC core is **0x7210_0000**. The following is a basic memory map for the OpenMAC, which uses 32-bit registers:

Table: OpenMAC Memory Map

I/O Address	Length	Description
0x7210_0000	0x2000	RAM for Ethernet Packets
0x7210_2000	0x400	OpenMAC core 32-bit registers
0x7210_2400	0x400	Ethernet Buffer Descriptors memory

Note
The TS-Kernel provides all the software support to use the OpenMAC 10/100 Ethernet core. The driver is named open_eth. For further information on the OpenMAC core such as the registers description, find the specific documentation at **www.opencores.org** website.

5.3 Video Out Connector

A standard DB15 VGA connector is available on the **TS-7300** for video interface. The video hardware core, named TS-VIDCORE, stored on the FPGA is a proprietary Technologic Systems implementation totally optimized for high-performance along with the **TS-7300** hardware. TS-VIDCORE features include:

- ✔ 16-bit color, 640X480 resolution, 800x600 available
- ✔ 8MB dedicated video memory running @ 95Mhz
- ✔ Simple and fast video accelerator

The TS-VIDCORE included on the **TS-7300** FPGA is a unique video core specifically designed in-tandem with the Linux framebuffer driver. Hardware accelerations were implemented and designed to mesh well with the capabilities of the Linux 2.4 frame buffer device driver API. The FPGA has dedicated video memory which allows CPU throughput and real-time response to remain fast and predictable, a clear advantage over devices which share system memory.

The physical address of the framebuffer SDRAM memory is **0x7220_0000**. The five specific 16-bit registers for the TS-VIDCORE framebuffer are located at physical address **0x7200_0030**.

Table: 16-bit register map for TS-VIDCORE control

I/O Addr	Description	Data Bits and such
BASE + 0	BLTCTRL: Bit blit control register - R/W	**Bits 15-13**: upper 3 bits of box pixel width **Bit 12:** bit blit source mode (0 – rectangle, 1 – linear) **Bits 11-6:** upper 6 bits of destination address of bit blit operation **Bits 5-0:** upper 6 bits of start address of bit blit operation
BASE + 2	BLTSZ: Bit blit width height register – R/W	**Bits 15-9:** box pixel width (lower 7 bits) **Bits 8-0:** box pixel height (0-512)
BASE + 4	SRCBLT: Bit blit source - R/W	**Bits 16-0:** lower 16 bits of source address or pixel fill color
BASE + 6	DSTBLT: Bit blit destination - R/W	**Bits 16-0:** lower 16 bits of destination address
BASE + 8	VIDCTRL: Video control register - R/W	**Bit 11:** raster page committed **Bit 10:** bit blit operation in progress **Bit 9:** horizontal sync enabled **Bit 8:** vertical sync enabled **Bit 7:** bit blit direction **0** – top to bottom: SRCBLT and DSTBLT are topleft corner addr **1** – bottom to top: SRCBLT and DSTBLT are bottomleft corner addr **Bit 6:** pixel fill enable (SRCBLT is pixel color instead of addr) **Bits 5-3:** raster page select (0-7) **Bits 2-0:** bus page select (0-7)

Note
Technologic Systems provides Linux drivers, pre-compiled graphical libraries and demo applications for the TS-VIDCORE video interface with framebuffer. Also, an USB keyboard and an USB mouse can be used with the video interface to come up a Linux console/terminal on the **TS-7300** by default. Refer to the Linux for TS-ARM User's Guide or contact us for further information.

5.4 USB Connector

The USB Connector on the **TS-7300** provide two USB interfaces for the user. These are directly connected to the EP9302 processor, which integrates an USB dual-port Open Host Controller Interface (Open HCI), providing full-speed serial communications ports at a baud rate of 12 Mbits/sec. Up to 127 USB devices (printer, mouse, camera, keyboard, etc.) and USB hubs can be connected to the USB host in the USB "tiered-star" topology. This includes the following features:

- ✔ USB 2.0 compatible
- ✔ Open HCI Rev 1.0 compliant
- ✔ USB device connections support at both low-speed (1.5 Mbps) and full-speed (12 Mbps)
- ✔ Root HUB integrated with 2 downstream USB ports
- ✔ Transceiver buffers integrated, over-current protection on ports
- ✔ Supports power management
- ✔ Operates as a master on the bus

Note
TS-Kernel implements all the necessary driver support to enable the USB OHCI. Also, a wide variety of USB drivers for devices such as mouse, keyboard and flash memory are available. Refer to the Linux for TS-ARM User's Guide or contact us for further information on how to integrate an USB device and an USB Linux driver with your **TS-7300**.

5.5 SD Card Connector – CPLD

The SD Card socket (ALPS connector) at the back side of the **TS-7300** enables SD Cards to be plugged to the SBC. The hardware core implemented by Technologic Systems is integrated inside the on-board CPLD. Technologic Systems has written a binary Linux driver module and a set of generic, OS-independent read/write routines for accessing the SD flash inside of an ARM object (.o) file. The format of the SD card must be in EXT2 format for proper operation with Linux as a root file system.

5.6 SD Card Connector – FPGA

There is an second SD Card socket (ALPS connector) on the **TS-7300**. This is not bootable and can be used as additional storage memory or for backup applications such as RAID functions.

The hardware core that controls the operation of the second SD card is integrated on the FPGA, therefore it will be only available if the appropriate FPGA core bitstream is loaded. The default TS-BITSTREAM for TS-7300 includes support for the second SD Card. In addition, Technologic Systems provides Linux drivers which enable reading and writing operations.

The SD Card core on the FPGA is located at physical address is **0x7200_0020**.

5.7 COM1 Connector

The COM1 RS-232 port uses a DB-9 male connector on the **TS-7300**.

Table: COM1 DB9 Pin-Outs

DB9 Pin	Signal	Description
1	DCD	Data Carrier Detect
2	RXD	Receive Data
3	TXD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9	--	--

The base address of COM1 appears in the physical address space at **0x808C_0000**. Technologic Systems provides software drivers to access this port. This full complement of RS-232 signals (all except Ring Detect) allows COM1 to interface to almost any serial RS-232 device.

5.8 COM2 Header

The base address of COM2 appears in the physical address space at **0x808D_0000**. Technologic Systems provides software drivers to access this port. COM2 has RS-232 support only for the TXD and RXD signals. This is sufficient to interface with serial devices that only require transmit and receive data lines. COM2 is accessible on a 10-pin header labeled COM2.

Table: TS-7300 COM2 Serial Port Header Pin Out

TX-	NC	NC	RX-	NC
6	**7**	**8**	**9**	**10**
1	**2**	**3**	**4**	**5**
TX+	RXD	TXD	RX+	GND

Note
The COM2 serial port header uses 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 3 (TXD) on the header connects to pin 3 on the DB-9.

Table: COM2 Serial Port Pin Configuration of Connectors

COM2 Header	DB9 Connector	RS-232	Half Duplex RS-485	Full Duplex RS-485
1	**1**		TX+ / RX+	TX+
2	**2**	RXD		
3	**3**	TXD		
4	**7**			RX+
5	**5**	GND	GND	GND
6	**6**		TX- / RX-	TX-
7	**7**			
8	**8**			
9	**9**			RX-
10	**-**	-	-	-

Adapter cables are available to convert COM2 to a standard 9-pin SubD male connector. In addition, COM2 has optional support for full or half-duplex RS-485

Note
The COM2 serial port 10-pin header has both RS-232 and RS-485 signals. Only one set may be used for a given application – it is not possible to use both at the same time.

Optional RS-485 Support

The COM2 Port supports RS-485 half duplex or full duplex operation. 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 ioctl Linux system call is used to initialize the automatic RS-485 control logic. For full-duplex operation, two twisted pairs are used and the transmit driver is typically enabled all the time. 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. In a typical RS-485 installation, the RS-485 drivers are frequently “tri-stated.” If fail-safe bias resistors are not present, the RS-485 bus may “float” and small amounts of noise can cause spurious characters at the receivers. 4.7K Ohm 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 in many installations are not required when cable lengths are less than 50 meters and the baud rate is 9600 or less. Termination resistors may be required for reliable operation when running longer distances or at higher baud rates. Termination resistors (typically 100-150 ohms) 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 **0x2240_0000** bit 1 will return a "1" when the RS-485 option is installed.

Note
A special ioctl command has been added to the TS-Kernel's serial code to turn off and on the automatic RS-485 feature. Refer to the Linux for TS-ARM User's Guide for additional software information on RS-485 support.

RS-485 Quick-Start Procedure

Perform the following steps:

5. Check that the RS-485 option is installed in the **TS-7300** (required).
6. Attach the RS-485 cable to the 10-pin header. (Adapter cables to SubDB9 connector are available from Technologic Systems).
7. Set the COM2 UART serial parameters (baud rate, data, parity, and stop bits, interrupts, etc) in your source code.
8. Use the TIOC_SBC's485 ioctl() to enable the automatic RS-485 feature.
9. Simply read() and write() data to the COM2 UART – the automatic RS-485 logic controls TX enable.

Automatic RS-485 TX Enable

The **TS-7300** supports fully automatic TX enable control under hardware control. This simplifies the design of half-duplex systems since turning off the transmitter and enabling the receiver in software is typically difficult to implement. If this is implemented in software, then the COM2 UART transmit holding register and the transmit shift register both must be polled until empty before deasserting the TX enable signal to the RS-485 driver. The design gets more complicated when using the TX FIFO or when using a multi-tasking OS such as Linux.

The implementation on the **TS-7300** uses hardware to automatically control when the RS-485 drivers and receivers are selected. The automatic RS-485 logic eliminates the need for any software polling and supervision. This circuit automatically turns on/off the RS-485 transceiver and receiver at the correct times. This only requires the COM2 Mode registers to be initialized once based on baud rate and data format.

Table: COM2 Mode Register

Value	MSB	MID	LSB	Mode
0x00	0	0	0	RS-232
0x01	0	0	1	Full-Duplex RS-485
0x02	0	1	0	-- Reserved
0x03	0	1	1	-- Reserved
0x04	1	0	0	Half-Duplex 9600 baud
0x05	1	0	1	Half-Duplex 19.2Kbaud
0x06	1	1	0	Half-Duplex 57.6Kbaud
0x07	1	1	1	Half-Duplex 115.2Kbaud

COM2 will default to RS-232 mode at reset (COM2 mode register = Hex 00). In order to switch COM2 into Full duplex RS-485 mode, the COM2 Mode register at **0x22C0_0000** must be set to Hex 01. In this mode, the TX drivers are always asserted.

In order to switch COM2 into half-duplex RS-485, then the COM2 mode register at 0x22C0_0000 must be set to Hex 04, 05, 06 or 07 depending upon the baud rate required. A second COM2 Mode register at **0x2300_0000** must be set to Hex 01 if the COM2 is in half-duplex mode and the data format is using 8 data bits with parity or 8 data bits with 2 stop bits.

Note
COM2 in half-duplex RS-485 supports baud rates of 9600, 19.2K, 57.6K and 115.2K. If you need to operate at any other baud rate, then please contact Technologic Systems

5.9 COM3-COM4-COM5-COM6 Headers

The **TS-7300** has four 10-pin headers named COM3, COM4, COM5 and COM6. These enable the four COM ports which are integrated on the FPGA. The FPGA must be loaded with the TS-BITSTREAM provided by default. These 4 FPGA COM ports support TXD, RXD, GND and the handshake lines RTS and CTS.

Table: TS-7300 COM3-4-5-6 Header Pin Out

NC	**RTS**	**CTS**	NC	NC
6	**7**	**8**	**9**	**10**
1	**2**	**3**	**4**	**5**
NC	**RXD**	**TXD**	NC	**GND**

By changing a specific register, the RTS signal will change into TXD2 and the CTS signal will change to RXD2. Therefore, each COM port in the FPGA can become 2 ports, totalizing 8 COM interfaces with none RTS and CTS handshake lines. Each COM port can be switched independently for maximum flexibility. This allows 6, 7, 8, 9, or 10 COM ports total on the **TS-7300** , all controlled via software.

The physical base address for the COM ports on the FPGA is **0x7200_0000**. Each COM port is controlled by two 16-bit registers, therefore 32 bytes of memory is used for the eight COM ports.

Table: FPGA COM ports addresses on TS-7300

COM port	I/O Addr	COM port	I/O Addr
COM3	BASE + 00	**COM7**	BASE + 10
COM4	BASE + 04	**COM8**	BASE + 14
COM5	BASE + 08	**COM9**	BASE + 18
COM6	BASE + 0C	**COM10**	BASE + 1C

All the FPGA COM ports can operate in both 8-bit mode or 9-bit mode. Technologic Systems provides 8-bit mode and 9-bit mode RS-232 driver, named “tsuart73”, for all the FPGA COM ports. Refer to the Linux for TS-ARM User's Guide for more information on Linux driver support.

5.10 DIO1 Header

The DIO1 Header provides +3.3V, GND, and 9 digital I/O lines that may be used to interface the **TS-7300** with a wide range of external devices. DIO lines DIO_0 thru DIO_7 are a byte-wide port using Port B on the EP9302 and are accessed via the data register at address location **0x8084_0004**. The DDR for this port is at address location **0x8084_0014**.

Table: DIO1 Header Pin Configuration

DIO1 Pin	Default Signal	TS-7300
1	DIO_0	
2	GND	
3	DIO_1	
4	Port_C0	EXT_RESET
5	DIO_2	DIO_8
6	SPI_Frame	
7	DIO_3	
8	DIO_8	ADC4
9	DIO_4	
10	SPI_MISO	
11	DIO_5	
12	SPI_MOSI	
13	DIO_6	
14	SPI_CLK	
15	DIO_7	
16	+3.3 V	

DIO_8 is accessed via bit 1 of Port F in the EP9302. The Port F data register is at address location **0x8084_0030**. The DDR address for this port is location **0x8084_0034**.

The Pin 4 of the DIO1 Header, in the default configuration, is accessed via bit 0 of Port C in the EP9302. The address location **0x8084_0008** is Port C Data Register and **0x8084_0018** is Port C Directon Register.

When accessing these registers, it is important not to change the other bit positions in these Port F registers. Other DIO1 Port functionality, used for dedicated **TS-7300** functions, utilize these same control registers. All accesses to these registers should use read-modify-write cycles.

Warning
All pins on the DIO header use 0-3.3V logic levels. Do not drive these lines to 5V.

When the DIO pins are configured as outputs, they can “source” 4 mA or “sink” 8 mA and have logic swings between GND and 3.3V. When configured as inputs, they have standard TTL level thresholds and must not be driven below 0 Volts or above 3.3 Volts. DIO lines DIO_0 thru DIO_3 have 4.7K Ohm “pull-up” resistors to 3.3V biasing these signals to a logic”1”. The other DIO pins have 100K Ohm bias resistors biasing these inputs to a logic “1”.

Note
On the TS-7300 some of the pins on the DIO1 header (Pin 1/DIO_0, Pin 3/DIO_1, Pin 7/DIO_3, Pin 9/DIO_4) are used for DMA operations during bootup by the SD card driver and therefore they will be driven as outputs. In order to use these pins as GPIO pins DMA will need to be disabled programming the following registers:

```
*(sysconbase + 0xC0) = 0xAA; //SFTWR_LCK register
*(sysconbase + 0x80)  = 0x08140d00; //DEVICE_CFG register
```

Refer to the **EP9301 User's Guide** for further information on SYSCON registers.

SPI Interface

The EP9302 Synchronous Serial Port is available on the DIO1 header. This port can implement either a master or slave interface to peripheral devices that have either Motorola SPI, or National Semiconductor Microwire serial interfaces.

The transmit and receive data paths are buffered with internal FIFO memories allowing up to eight 16-bit values to be stored for both transmit and receive modes. The clock rate is programmable up to 3.7 MHz and has programmable phase and polarity. The data frame size is programmable from 4 to 16 bits.

By using some of the DIO1 Header pins as peripheral Chip Select signals, a complete interface is available for addressing up to 9 SPI peripherals. The SPI bus pins are defined in the table below:

Table: DIO1 Header Pin-Outs for SPI

DIO1 Pin	Name	Function
10	SPI_MISO	Master In/Slave Out
12	SPI_MOSI	Master Out/Slave In
14	SPI_CLK	Clock
6	SPI_Frame	SPI Frame pin

A fourth SPI bus function [SPI_Frame] is available by adding a 10 ohm resistor in the position labeled R1 on **TS-7300**. This signal is not required for many SPI peripherals but it may prove useful in some applications.

> **Warning**
> The SPI bus pins use 0-3.3V logic levels. Do not drive these lines to 5V.

Refer to the **EP9301 User's Guide** and see Chapter 19 for more details on using SPI bus.

Matrix Keypad

DIO signals DIO_0 thru DIO_7 are physically arranged to allow a 16-pin (2x8) ribbon cable to directly connect a **4x4 matrix keypad.** Sample code is available for the Matrix Keypad. Contact Technologic Systems for further information.

A/D Converter - Cirrus EP9302

The Cirrus EP9302 features a 5 channel, 12-bit Analog to Digital Converter with an analog multiplexer, having an input range of 0 to 3.3 V. The Cirrus A/D converter can do a maximum of 925 samples per second, and requires a settling time of 2 milliseconds between channel switches.

On the **TS-7300**, the DIO Port pin 8 is connected to one of the EP9302 12-bit A/D converter inputs (ADC4). The A/D lines on the DIO1 header can be used to measure analog signals in the range of 0 to 3.3V.

Table: ADC Switch Values (EP9302)

Input to Measure	*ADC Switch Value*
ADC4	0x0000_0610

To maintain 12-bit accuracy, the analog signal being measured must have a low source impedance (less than 10 ohms). Otherwise, an operational amplifier may need to be added to buffer the A/D input. For detailed information, please see the Cirrus **EP9301 User's Guide**, page 518.

The following steps outline the software execution to use the Cirrus A/D converter:

1. Unlock the software lock before setting the TSEN bit in the ADCClk register by writing 0xAA to the ADCSWLock register (0x8090_00C0). “OR” in the TSEN bit (bit 31) to the ADCClkDiv register (0x8093_0090)
2. Unlock the software lock (again) before OR'ing in the ADCEN (ADC clock enable, bit 31) to 0x8093_0080
3. Clear bit 2, the ADCPD (ADC Power Down) bit, at 0x8093_0080. This bit MUST be set to 0 (see page 91 of the EP9301 User's Guide)
4. After unlocking the software lock, write the channel's magic value (see Cirrus EP9301 User's Guide, table 20-2) to the ADCSwitch register (0x8090_0018) to select that channel for the next data acquisition
5. Poll the ADCResult register (0x8090_0008) until bit 31 is not set
6. Using a 32 bit read operation, read the result from 0x8090_0008, masking off the upper 16 bits

Interpreting Cirrus A/D Converter

The Cirrus on-chip A/D converter is a successive approximation A/D converter. Each A/D channel is calibrated on the **TS-7300** and these 16-bit values are stored in non-volatile EEPROM. These calibration values minimize the offset errors and gain errors in the EP9302 A/D. It is important for the user program to use these values as per our sample code, which can be found either on our website or in the CD included in the Developer's Kit. Two reference points, 0 and 2.5 Volts, with the corresponding reference values stored in EEPROM. Bytes 0x07EB through 0x07FE of the EEPROM hold a two dimensional array:

[channel number][0V ref. point, 2.5V ref. Point]

The reference points are stored as a 16 bit value, and should be used to correlate the values returned by the Cirrus A/D converter to voltage.

5.11 DIO2 Header

The **TS-7300** provides up to 35 DIO lines connected straight to the FPGA through the FPGA DIO header. The FPGA DIO is a 40-pin header divided in 2 sub-headers of 20 pins each. The first header is unamed on the board, while the second is called DIO2. The 40 total FPGA DIO lines can also be used as a FPGA connected expansion bus instead of the CPU connected regular PC104 expansion bus.

The default TS-BITSTREAM provided with **TS-7300** enables hardware functionality for VGA video. This default bitstream maps the VGA video signals to the first 20-pin FPGA DIO header (unamed one). The 17 total video signals can be changed to XDIO lines if the video is disabled on the FPGA bitstream. The following is the numbering scheme for this header:

Table: FPGA DIO Pin-out (first header)

GND	RED0	RED1	RED2	RED3	RED4	HSYN	VSYN	OEV	+5V
2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
BLU0	BLU1	BLU2	BLU3	BLU4	GRN0	GRN1	GRN2	GRN3	GRN4

Pin #20 of this header is fused +5V (polyfuse), and pin #2 is ground. Pin #18 can be externally driven high to disable DB15 VGA connector DACs.

In the same way, the default TS-BITSTREAM maps the 18 total XDIO signals to the second 20-pin FPGA DIO header, named DIO2 on the board:

Table: FPGA DIO Pin-out (second header - DIO2)

GND	XD2_2 DIO10	XD2_3 DIO11	XD2_4 DIO12	XD2_5 DIO13	XD2_6 DIO14	XD2_7 DIO15	RX_LD DIO16	TX_LD DIO17	+3.3V
2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
DIO0 XD1_0	DIO1 XD1_1	DIO2 XD1_2	DIO3 XD1_3	DIO4 XD1_4	DIO5 XD1_5	DIO6 XD1_6	DIO7 XD1_7	DIO8 XD2_0	DIO9 XD2_1

Pin #20 of the DIO2 header is regulated 3.3V, and pin #2 is ground. Pin #16 and #18 also go to the red and green LEDs (active low). Pin #19 is a dedicated clock input and cannot be programmed for output

The on-board FPGA can be loaded with any user bitstream. This enables the 40-pin header to be used for any specific function. To check how the 35 total DIO pins are connected from the header to the FPGA, refer to the TS-7300 schematic provided by Technologic Systems.

Warning
DO NOT DRIVE THE FPGA DIO SIGNALS OVER 3.3V.
The DIO lines of the 40-pin header are connected straight into the FPGA pads unbuffered. Make sure to use signal conditioning or any other protection technique required by your FPGA design.

TS-XDIO

The following is a description of the TS-XDIO port on the **TS-7300**. This core is designed to implement common controller tasks that are difficult, costly (CPU intensive), or impossible to accomplish in software with regular DIO / GPIO hardware and the facilities of the 200Mhz ARM processor.

The default TS-BITSTREAM for the **TS-7300** FPGA implements two XDIO ports which appear at physical locations **0x72000040- 0x72000043** and **0x72000044-0x72000047**. The first XDIO core uses signals DIO0 to DIO7 on the FPGA DIO2 header, while the second XDIO core uses signals DIO8 to DIO15. DIO16 and DIO17 signals are used for Rx and Tx LEDs. The 2 available clock sources are the 75Mhz-13ns fast clock and the 37.5Khz-26667ns slow clock (as opposed to 14.7456Mhz and 32.768Khz on the TS-7260) The IRQ connected is ARM IRQ 40 (ISA IRQ 7). TS-XDIO port features includes:

- ✔ 8-pins, individually programmable as schmitt-trigger inputs, outputs, or special functions (3.3V levels)
- ✔ glitch-detector (all pins) to catch edges/pulses as short as one fast clock period without continuous software polling
- ✔ edge-counter/frequency meter (pin 7) max incoming clock period of one-half the core clock frequency
- ✔ quadrature counter (pins 6,7) with max increment / decrement rate of one-half the maximum edge-counter frequency
- ✔ Pin 5 can optionally be used as an index pulse for reseting the edge counter or quadrature counter to 0
- ✔ PWM (pulse width modulation) output on pin 4 with period between two fast clock periods and 8190 slow clock periods, with 0.02% duty cycle precision
- ✔ One-shot programmable high or low output pulse (pin 5) of duration between one fast clock period and 4095 slow clock periods

- ✔ pulse / period timer (any pin) can measure high/low pulse time or full period lengths at fast clock period resolution

The TS-XDIO functionality can be controlled through 4 control/status registers that appear at physical memory. The register 0 is used to set the XDIO mode and general configuration:

Table: TS-XDIO Register #0

Bits	Function	Binary Values and Description
Bits 7-6	MODE (selects what registers 1-3 are)	**00** - pin direction, data register, IRQ/DRQ control **01** - edge / quadrature counter and glitch monitor **10** - input pulse timer and pin select **11** - PWM / pulse high / low-time set
Bits 5-4	Pulse timer mode and polarity	**00** - negedge to posedge (low pulse time) **01** - posedge to negedge (high pulse time) **10** - negedge to negedge (period time) **11** - posedge to posedge (period time)
Bit 3		pulse timer reset (W) or waiting for trailing edge (R)
Bit 2	Quadrature/Edge	quadrature enable / edge counter disable
Bit 1	PWM	PWM enable / pulse start
Bit 0	Clock select	clock select (0 – fast clock, 1 – slow clock)

Registers 1-3 are dependent upon the mode set on Bits 7-6 of Register 0. The table below describes the remaining XDIO registers:

Table: TS-XDIO Registers 1-3 with MODE 0

Register	Bits	Function and Description
Reg #1		DIO direction register (**1** output, **0** input)
Reg #2		DIO data register
Reg #3	**bit 0**	irq/drq on quadrature change direction
	bit 1	irq/drq on quadrature / edge overflow / underflow
	bit 2	irq/drq on pin glitch (pins 0-3)
	bit 3	irq/drq on pin glitch (pins 4-7)
	bit 4	irq/drq on pulse timer trailing edge
	bit 5	irq/drq on quadrature / edge counter index pulse (if enabled)
	bit 6	reserved (may be used in customer specific TS-XDIO cores)
	bit 7	drq select (**1** drq, **0** irq)

Table: TS-XDIO Registers 1-3 with MODE 1

Register	Bits	Function and Description
Reg #1		Glitch monitor, read or write resets all to 0
	bit 7	enable index pulse on pin 5
	bits 6-0	1 - glitch detected on pin 0 - inactive since last read
Reg #2	**bits 7-0**	bits 7-0 of edge / quadrature counter
Reg #3	**bits 7-0**	bits 15-8 of edge / quadrature counter

Table: TS-XDIO Registers 1-3 with MODE 2

Register	Bits	Function and Description
Reg #1		Pin select / timer
	bit 7	pulse time accumulate - keeps on running adding pulse times
	bits 6-4	input pin number (0-7)
	bits 3-0	bits 19-16 of pulse timer
Reg #2	**bits 7-0**	bits 7-0 of pulse timer
Reg #3	**bits 7-0**	bits 15-8 of pulse timer

Table: TS-XDIO Registers 1-3 with MODE 3

Register	Bits	Function and Description
Reg #1	**bits 7-4**	bits 11-8 of high time
	bits 3-0	bits 11-8 of low time
Reg #2	**bits 7-0**	bits 7-0 of high time
Reg #3	**bits 7-0**	bits 7-0 of low time, one-shot mode enabled if all zeros

TS-XDIO Programming Considerations

- ✔ If quadrature index pulse is enabled, counter is reset on positive edge of index pulse if DIO Data Register, bit 5 is set to 0, negative edge if set to 1.
- ✔ Pulse width modulation (PWM) time is 2 + high / low time clock periods of 14.7456 MHz or 32 KHz.
- ✔ If using Pulse Width Modulation (PWM) in “One shot” mode (single pulse), Type is high pulse (low-high-low) if DIO Data Register, bit 4 is set to 0, low pulse (high-low-high) if set to 1.
- ✔ One could use the 20-bit Pulse Timer as a free-running timer by setting the Accumulate bit and setting pin select to PWM output (pin 5). 14.7456 MHz counter can be enabled by “watching periods,” slower free-running counters can be had by clever use of high-time and low-time PWM and accumulating pulse times. (e.g. high-time 1, low-time 224 creates a 65.536Khz free-running timer that overflows once every 16 seconds).
- ✔ Edge counter counts both edges of a waveform; divide by 2 for number of periods / frequency. To count X edges and then interrupt, pre-load counter value with 2^16 - X and enable IRQ on overflow.
- ✔ Max/min quadrature can be counted by enabling IRQ / DRQ on quadrature change of direction. Max / Min is value +/- 1 as long as IRQ / DRQ latency is less than 1 quadrature cycle time.
- ✔ The XDIO Core does not reset -- registers and state remain if board undergoes a reset.
- ✔ Upon power-up, DIO Data Direction Register is all 0's (inputs), but other register contents are undefined

Note
Technologic Systems provides C API, source code and examples for TS-XDIO programming. Visit the **TS-XDIO webpage** for more details.

5.12 LCD Header

The LCD Port can be used to interface to a standard alphanumeric LCD display or as 11 additional digital I/O lines. The header has been arranged to allow a 14-pin ribbon cable to directly connect to industry standard LCD displays.

Table: LCD Header Pin Configuration

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

Technologic Systems has a 2x24 LCD display available with software drivers for rapid development.

DIO lines LCD_0 thru LCD_7 are a byte-wide port using Port A on the EP9302 and are accessed via the data register at physical address location **0x8084_0000**. The DDR for this port is at **0x8084_0010**. Because this port is interfacing to a 5V LCD, 1.0K Ohm resistors have been added in series between the EP9302 and the LCD_0 thru LCD_7 pins. This is required since the LCD data bus could be driving these lines above 3.3V. The series resistors prevent the LCD from overdriving the EP9302 Port A pins.

Note
On the **TS-7300 only**, the LCD_7 (eight data bit) is accessed on **bit 0** of Port C (address location 0x8084_0008 is Port C Data Register and 0x8084_0018 is Port C Directon Register).

If using these pins for general purpose DIO, the current sourcing and sinking capability of these DIO pins is limited by the 1.0K Ohm resistors.

LCD_EN, LCD_RS, and LCD_WR are DIO pins using EP9302 Port H bits 3 through 5 respectively and are accessed via the data register at physical address location **0x8084_0040**. The DDR for this port is at **0x8084_0044**. When these DIO pins are configured as outputs, they can source 4 mA or sink 8 mA and have logic swings between 3.3V and ground. When configured as inputs, they have standard TTL level thresholds and must not be driven below 0 Volts or above 3.3 Volts. These DIO pins have 100K Ohm bias resistors biasing these inputs to a logic “1”. It is important not to change the other bit positions in these Port H registers since the other DIO pins are being used on the **TS-7300**. All accesses to these registers should use read-modify-write cycles.

Pin 4 on this header (labeled Bias) is a 620 Ohm resistor to ground for LCD contrast biasing. The 5V power on the LCD header has a 750 mA Poly-Fuse to limit the current.

Warning
LCD pins 3, 5, and 6 use 0-3.3V logic levels. Do not drive these lines to 5V.

LCD Interface

A 14-pin LCD connector is provided for interfacing with standard alphanumeric LCD displays. These displays use a common controller, the Hitachi HD44780 or equivalent. While software written for the HD44780 works 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.

Table: LCD Header Signals

PIN	Function	Comments
1	LCD 5V	LCD Power
2	LCD_GND	
3	LCD_RS	Register select
4	Bias	620 Ohm to GND
5	LCD_EN	Active high enable
6	LCD_WR#	Active low write
7	LCD_D1	D0 - D7: Buffered bi-directional data bus
8	LCD_D0	
9	LCD_D3	
10	LCD_D2	
11	LCD_D5	
12	LCD_D4	
13	LCD_D7	
14	LCD_D6	

Note that the pin-outs in the above table are not the standard given for LCD displays. However, these pin-outs allow a standard ribbon cable to be used when the ribbon cable is attached to the backside of the LCD.

> **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.

Sample code is available for the LCD. Contact Technologic Systems for more information.

5.13 JTAG Header

The JTAG header can be used to program the on-board CPLD using special software and hardware support tools. However, it is not available for application debug purposes since it has no connection to the EP9302 JTAG interface. The following table shows which pins of the JTAG header are used for the JTAG interface signals:

Table: JTAG signals at JTAG Header

PIN	Signal	Description
9	TCK	Test Clock
11	TDI	Test Data In
13	TMS	Test Mode Select
14	TDO	Test Data Out
1, 3, 5	3.3 VCC	Power Supply
10, 12	GND	Ground

The JTAG header is also utilized for jumper configuration. See the Jumpers section of this manual for more details.

Technologic Systems has made the design choice to save on board real-estate and not bring out the JTAG header. If you need access to the JTAG pins, a skilled technician can solder wires to the pins.

5.14 Power Supply Header

The **TS-7300** requires regulated 5VDC at 450 mA @ 200 MHz(maximum). It is possible to lower this power significantly by lowering the CPU clock rate or by powering-down the Ethernet PHY chip. For example, by shutting down the Ethernet PHY chip and scaling down the CPU clock rate to 20 MHz, one can obtain power consumptions less than 150 mA. If you really need a low-power board, consider the **TS-7260**, which is optimized for low power applications.

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.

> **Warning**
> Supply voltages over 6 VDC may damage the **TS-7300**.

Be sure to use a regulated 5 VDC power supply, preferably with current limiting to 1 to 3 Amps. A current limited supply is very forgiving of common errors during development. A PC power supply that may be capable of supplying 20 Amps or more is not recommended. It is possible to do irreversible damage to the **TS-7300** if the polarity on the power leads is reversed.

6 PC/104 BUS EXPANSION

The PC/104 is a compact implementation of the PC/AT ISA bus ideal for embedded applications. Designers benefit from an established industry standard bus that already has well-developed documentation and many compatible peripherals available in the market place. 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.

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, No. IEEE P996.1.

This bus allows multiple peripheral boards 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, Digital I/O, Ethernet ports, and LCD drivers may be easily added.

Note
The **TS-7300** implements a sub-set of the full PC/104 bus. This allows the support of many common I/O peripheral boards. Some of the PC/104 signals are not supported; for example, the DMA signals. These pins are used on the **TS-7300** to support data lines D8- D15. This means that a full 16-bit data bus implementation of the PC/104 bus is possible by only using the 64-pin connector. The supplemental 40-pin connector is not an available option for the **TS-7300** – any new design can implement a full 16-bit data bus with only the 64-pin connector.

PC/104 peripherals will appear in the **TS-7300's** physical address space in one of four address regions depending upon whether it is emulating an x86 Memory cycle or I/O cycle and whether it needs to be a 8-bit cycle or a 16-bit cycle. Each region is a full 1MB in size, even though the I/O region will typically only use a 1 Kbyte region for legacy products.

Table: PC/104 Memory Map

Physical Address Region	Emulates x86 cycle
11E0_0000 thru 11E0_03FF	8-bit I/O cycles
21E0_0000 thru 21E0_03FE	16-bit I/O cycles
11A0_0000 thru 11AF_FFFF	8-bit Memory cycles
21A0_0000 thru 21AF_FFFE	16-bit Memory cycles

I/O cycles on the PC/104 expansion bus strobe either IOR# or IOW#, while Memory cycles strobe the MEMR# or MEMW# signals. For example, a TS-SER1 peripheral board can be jumper-selected as COM3, which would correspond to a PC I/O base address of 0x3E8. Since this is an 8-bit peripheral, this COM port must be accessed at the physical base address of 0x11E0_03E8.

Table: Signals on the 8-Bit Bus

Pin	Signal	Pin	Signal
A1	BHE#*	B1	GND
A2	Data 7	B2	RESET
A3	Data 6	B3	+5 V
A4	Data 5	B4	Data 8*
A5	Data 4	B5	Reserved
A6	Data 3	B6	RTC_CS*
A7	Data 2	B7	Reserved
A8	Data 1	B8	RTC_ALE*
A9	Data 0	B9	Reserved
A10	IOCHRDY	B10	GND
A11	Address 20*	B11	MEMW#
A12	Address 19	B12	MEMR#
A13	Address 18	B13	IOW#
A14	Address 17	B14	IOR#
A15	Address 16	B15	Reserved
A16	Address 15	B16	Reserved
A17	Address 14	B17	Data 9*
A18	Address 13	B18	Data 10*
A19	Address 12	B19	Address 21*
A20	Address 11	B20	Data 12*
A21	Address 10	B21	IRQ7
A22	Address 09	B22	IRQ6
A23	Address 08	B23	IRQ5
A24	Address 07	B24	GND*
A25	Address 06	B25	Data 11*
A26	Address 05	B26	Data 13*
A27	Address 04	B27	Data 14*
A28	Address 03	B28	Data 15*
A29	Address 02	B29	+ 5 V
A30	Address 01	B30	Osc (14.3 MHz))
A31	Address 00	B31	GND
A32	GND	B32	GND

* These signals have a non-standard usage.

6.1 Adding Serial Ports

If your project requires more than the included **TS-7300's** serial ports, additional ports may be added via the PC/104 expansion bus. Technologic Systems offers three different **peripheral boards** (TS-SER1, TS-SER2, and TS-SER4) that add 1, 2, or 4 extra COM ports respectively. These can be configured using PC designations COM1, COM2, or higher.

These ports will appear in the PC/104 8-bit IO address space (**0x11E0_0000** base address). For example, a TS-SER1 peripheral board can be jumper selected as COM4, which would correspond to a PC I/O base address of 0x2E8. On the **TS-7300**, this COM port will appear at the physical base address of 0x11E0_02E8. The PC/104 Expansion Bus has IRQ5, IRQ6, and IRQ7 available for additional serial port interrupts.

Typically each serial port has a dedicated interrupt, but the TS-SER4 peripheral board 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.

Drivers are available in the TS-Kernel to support these extra COM ports.

6.2 Adding Ethernet Ports

Additional ethernet ports can be added to the **TS-7300** using appropriate PC/104 hardware provided by Technologic Systems, allowing the implementation of hardware and software applications such as network firewalls, gateways or routers.

The **TS-ETH10** is a 10Mb ethernet PC/104 card based on the CS8900 ethernet chip that is perfectly suited for a WAN uplink. Up to 4 TS-ETH10 cards can be stacked on the PC/104 connector since the add-on board has 4 separate address decodes. TS-ETH0 driver is available for TS-Kernel. Documentation and support is provided by us as well.

6.3 Adding CAN Bus

Controller Area Network (CAN) is a serial network primarily used in embedded systems that was originally designed for the automotive industry, but has also become a popular bus in industrial automation and other applications. The addition of a CAN bus to the **TS-7300** is easily accomplished by the optional TS-CAN1 board through the PC/104 bus.

The **TS-CAN1** provides one single channel CAN interface and enables a PC/104 embedded systems to be connected to CAN1.1 and CAN2.0a networks. It uses a Philips SJA1000 CAN controller which is opto-isolated from the physical CAN network, increasing the system security. Up to four boards can be installed in any single system. The Philips SJA1000 CAN controller can also be found at Technologic Systems **TS-7KV** multi-function video board.

Technologic Systems provides complete SJA1000 Linux driver support for TS-CAN1 and TS-7KV, including CANOpen high level protocol. Documentation and support for CAN application development are also provided.

6.4 Adding Video Support

The **TS-7KV** multi-function PC/104 peripheral board can be added to the **TS-7300** if your application needs a video solution, . The TS-7KV implements the TS-VIDCORE using an FPGA solution to deliver 16-bit color, 640X480 resolution video, 8MB dedicated video memory running at 95Mhz, and a simple fast video accelerator. The TS-7KV includes a standard DB15 VGA connector and a 10 pin header. Regarding software support, an accelerated Linux framebuffer driver is available for TS-Kernel. In addition, the QT/Embedded graphical library can be used with the video solution and compiled packages can be found at Technologic Systems website.

The TS-VIDCORE included on the FPGA is a unique video core specifically designed in tandem with the Linux framebuffer driver. Hardware accelerations were implemented and designed to mesh well with the capabilities of the Linux 2.4 frame buffer device driver API. Technologic Systems engineers carefully chose which functionality was implemented in hardware, which functionality was left out for software and which functionality was hard-coded in the bitstream. The result is a higher speed design that is technically elegant and simple in good embedded systems fashion. The TS-7KV has dedicated video memory which allows CPU throughput and real-time response to remain fast and predictable, a clear advantage over devices which share system memory.

The **TS-7300** SBC has on-board VGA video functionally integrated on the FPGA, using the same TS-VIDCORE used on TS-7KV.

Note
When running Linux and using a TS-7KV video board, a USB mouse and a USB keyboard, it is possible to use the **TS-7300** embedded system as a desktop-like system, independent of the host machine. TS-Kernel provides the complete solution for automatic configuration during boot time and console output at the plugged monitor. Contact Technologic Systems for further information.

7 LEDS, JUMPERS AND BUTTONS

7.1 Status LEDs

The **TS-7300** has two LEDs (one Red and one Green) available for user software. These LEDs may be used for diagnostics, status messages, and simple output. When power is first supplied to the **TS-7300**, both LEDs are immediately turned on under hardware control. Once the processor begins execution, the LEDs are turned off, and then flashed on and off again briefly. After booting is complete, these LEDs can be used for user applications.

The RED and Green LEDs can be controlled at physical address location **0x8084_0020**. Bit 1 is the RED LED and bit 0 is the Green LED. A Logic "1" turns the LED on.

7.2 Jumpers

The **TS-7300** contains six default jumpers for specific board configuration:

Table: TS-7300 Jumpers Description

Jumper	Description
JP1	Boot to Serial Port COM1. This is a recovery mechanism that allows the **TS-7300** to boot using COM1 instead of the on-board Flash chip.
JP2	Enable Serial Console. COM1 is used as the Console. (If JP4 is not installed). When removed VGA output is the default console.
JP3	Write Enable Flash. Remove this jumper to write protect the Flash drive.
JP4	Console swapped to COM2 (Requires JP2 installed to enable console).
JP5	User Jumper.
JP6	Fastboot Jumper. If JP6 is present then the TS-7300 will fast boot to a shell prompt in the initial ramdisk (busybox filesystem) with the Debian Linux filesystem (partition 3) mounted read only on /mnt/root/.

Jumpers 2, 3, 4, and 5 can be read at physical address location 0x1080_0000. The status of all jumpers is available to software through the physical memory as describes the table below:

Table: Memory Map of Jumpers

Address	JP and Bit	On/Off
0x1080_0000	JP2 is bit 0	Logic "1" = Jumper ON
0x1080_0000	JP3 is bit 1	Logic "1" = Jumper ON
0x1080_0000	JP4 is bit 3	Logic "0" = Jumper ON
0x1080_0000	JP5 is bit 4	Logic "0" = Jumper ON
0x2280_0000	JP6 is bit 0	Logic "1" = Jumper ON

7.3 Buttons

On the **TS-7300** model, the reset push-button is replaced by pins 4 and 2 of the DIO1 Header. Refer to the DIO1 Header section to further information about the reset and shutdown process.

8 SPECIFICATIONS

To ensure optimum product operation you must maintain the operational environmental specifications listed in the table below.

Table: Environmental Specification for TS-7300

Environmental Specification	Standard Temp	Extended Temp
Ambient Temperature	-20° to +70° C The internal temperature must not exceed +70° C.	-40° to +85° C Extended temperature range is also standard in our TS-7300 product **Note:** Extended Temp requires lower CPU speed (<=166Mhz) at higher temperatures **Note:** Refer to your product manual, or contact Technologic Systems if the environmental temperature of the location is in doubt.
Relative Humidity	0 to 90% relative humidity. Not to exceed 90% non-condensing.	Not to exceed 90% non-condensing.

9 FURTHER REFERENCES

- **Getting Started with TS-Linux**
  (http://www.embeddedarm.com/documentation/software/arm-tslinux-ts72xx.pdf)
- **Linux for TS-ARM User's Guide**
  (http://www.embeddedarm.com/documentation/software/arm-linux-ts72xx.pdf)
- **TS-7300 Data Sheet** (http://www.embeddedarm.com/documentation/ts-7300-datasheet.pdf)
- **EP9301 User's Guide**
  (http://www.embeddedarm.com/documentation/third-party/ts-7000_ep9301-ug.pdf)
- **EP9301 Data Sheet**
  (http://www.embeddedarm.com/documentation/third-party/ts-7000_ep9302-ds.pdf)
- **TS-7000 Yahoo Users' Group** (http://groups.yahoo.com/group/TS-7000/)

APPENDIX A: DOCUMENT HISTORY

Date of Issue/Revision	Revision Number	Comments
March 03, 2006	Preliminary Draft	Preliminary release for first customer ship – not complete
May 18, 2006	1.1	Empty sections completed
Jul 20, 2006	1.2	COM ports pin-out fixed + further info about FPGA provided + minor corrections and improvements
Jun 26, 2007	1.3	Minor corrections and improvements, include of software content on the getting started section, series renamed to TS-72XX
Nov 21, 2007	1.4	Minor corrections
July 2, 2008	1.5	Fixed broken web links
June 1, 2009	1.6	Updated mailing address
January 27, 2010	1.7	Changed JP5 description, minor spelling corrections
April 1, 2010	1.8	Corrected section 7.3, reset pins were incorrect

APPENDIX B: MEMORY AND REGISTER MAP

Address Region	Function
0xF000_0000 - 0xFFFF_FFFF	nCS0 (not used)
0xE000_0000 - 0xEFFF_FFFF	SDRAM (TS-7250)
0xD000_0000 - 0xDFFF_FFFF	SDRAM (not used)
0xC000_0000 - 0xCFFF_FFFF	SDRAM (not used)
0x8084_0000 - 0x8084_00C8	GPIO control registers
0x8000_0000 - 0x800F_FFFF	AHB mapped registers
0x71C0_0000 - 0x71FF_FFFF	CS7 PC/104 8/16 bit I/O (user selectable timing)
0x7180_0000 - 0x71BF_FFFF	CS7 PC/104 8/16 bit Memory (selectable timing)
0x7000_0000 - 0x70FF_FFFF	TS-9420 Flash
0x7000_0000 - 0x7FFF_FFFF	CS7 (bit bus cycles)
0x6000_0000 - 0x60FF_FFFF	on-board Flash (TS-7200)
0x6000_0000 - 0x6FFF_FFFF	CS6 (Flash)
0x3000_0000 - 0x3FFF_FFFF	CS3 (not used)
0x21C0_0000 - 0x21FF_FFFE	PC/104 16-bit I/O
0x2180_0000 - 0x21BF_FFFF	PC/104 16-bit Memory
0x2000_0000 - 0x2FFF_FFFF	CS2 (16-bit bus cycles)
0x11C0_0000 - 0x11FF_FFFF	PC/104 8-bit I/O
0x1180_0000 - 0x11BF_FFFF	PC/104 8-bit Memory
0x1000_0000 - 0x1FFF_FFFF	CS1 (8-bit bus cycles)
0x0001_0000 - 0x0000_FFFF	SDRAM region

Register Address	Function
0x8090_0020	Cirrus A/D lock register
0x8090_0018	Cirrus A/D channel select register
0x8090_0008	Cirrus A/D result register (RO)
0x808D_0000 - 0x808D_FFFF	UART2 control registers
0x808C_0000 - 0x808C_FFFF	UART1 control registers
0x808A_0000 - 0x808A_FFFF	SPI control registers
0x8084_0044	LCD_EN, LCD_RS, LCD_WR direction reg.(bits3-5)
0x8084_0040	LCD_EN, LCD_RS, LCD_WR data reg.(bits3-5)
0x8084_0034	DIO_8 direction register (bit 1)
0x8084_0030	DIO_8 data register (bit 1)
0x8084_0020	On-board LEDs register (bits 0, 1)
0x8084_0018	Port C direction register (TS-7300) LCD_7 direction register (bit 7)
0x8084_0014	DIO_0 thru DIO_7 direction register (R/W)
0x8084_0010	LCD_0 thru LCD_7 direction register (R/W)
0x8084_0008	Port C data register (TS-7250) 1000 mA driver output on DIO (bit 0) (TS-7300) LCD_7 data register (bit 7)
0x8084_0004	DIO_0 thru DIO_7 data register (R/W)
0x8084_0000	LCD_0 thru LCD_7 data register (R/W)
0x8081_0000 - 0x8081_FFFF	Timer Control registers
0x8080_0000 - 0x8FFF_FFFF	APB mapped registers
0x800B_0000 - 0x800B_FFFF	VIC 0 registers
0x8006_0000 - 0x8006_FFFF	SDRAM control registers
0x8002_0000 - 0x8002_FFFF	USB registers
0x8001_0000 - 0x8001_FFFF	Ethernet MAC registers
0x7220_0000 - 0x729F_FFFF	(TS-7300) FPGA SDRAM region
0x7210_0000 - 0x7200_FFFF	(TS-7300) FPGA MAC core 32-bit registers

Register Address	Function
0x7200_0044 - 0x7200_0047	(TS-7300) FPGA DIO2 TS-XDIO #2 registers
0x7200_0040 - 0x7200_0043	(TS-7300) FPGA DIO2 TS-XDIO #1 registers
0x7200_0030 - 0x7200_003A	(TS-7300) FPGA Video core 16-bit registers
0x7200_0020 - 0x7200_0027	(TS-7300) FPGA SD Card core 16-bit registers
0x7200_0000 - 0x7200_001F	(TS-7300) FPGA COM3-4-5-6-7-8 16-bit registers
0x6000_0000	(TS-7250) NAND Flash data register
0x6040_0000	(TS-7250) NAND Flash control register (bits 0-2)
0x6080_0000	(TS-7250) NAND Flash Busy status (bit 5)
0x23C0_0000	WDT Feed register (bits 0-2)
0x2380_0000	WDT Control register (bits 0-2)
0x2340_0000	PLD version (bits 0-2)
0x2300_0000	COM2 RS-485 control register
0x22C0_0000	COM2 RS-485 control register (bits 0-2)
0x2280_0000	JP6 (bit 0)
0x2280_0000	Booting from TS-9420 (bit 1)
0x2280_0000	TS-9420 present (bit 2)
0x2240_0000	MAX197 A/D option present (bit 0)
0x2240_0000	COM2 RS-485 option present (bit 1)
0x2200_0000	Model Number (bits 0-2)
0x21E0_0000 - 0x21E0_03FE	PC/104 16-bit I/O (legacy support)
0x2100_0000	CF IDE 16-bit register
0x13C0_0000 - 0x13C0_0001	(TS-7300) FPGA loader registers
0x1340_0000 - 0x1340_0002	(TS-7260) Additional COM5 registers
0x1300_0000 - 0x1300_0002	(TS-7260) Additional COM4 registers
0x12C0_0000 - 0x12C0_0003	(TS-7260) DIO2 Header TS-XDIO registers
0x12C0_0000 - 0x12C0_0001	(TS-7260) DIO2 Header Basic DIO registers
0x1240_0000 - 0x1240_0002	(TS-7260) COM3 serial port registers
0x1200_0000	(TS-7260) Power Management register
0x11E0_0000 - 0x11E0_03FF	PC/104 8-bit I/O (legacy support)
0x11A0_0000 - 0x11AF_FFFF	PC/104 8-bit Memory (legacy support)
0x1170_0000	RTC R/W data register
0x1100_0001 - 0x1100_0007	CF IDE 8-bit registers
0x10F0_0000 - 0x10F0_0001	MAX197 A/D registers
0x1080_0000	MAX197 A/D busy bit (bit 7) (RO)
0x1080_0000	JP2-JP5 (bits 0,1,3,4) (RO)
0x1080_0000	COM1 DCD (bit 6) (RO)
0x1080_0000	Write Only index register
0x1040_0006 - 0x1040_0007	CF AUX IDE 8-bit registers

APPENDIX C: DOWNLOADS - SCHEMATICS AND MECHANICAL DRAWING

- **TS-7300 schematic**
  (http://www.embeddedarm.com/documentation/ts-7300-schematic.pdf)
- **TS-7300 mechanical drawing**
  (http://www.embeddedarm.com/documentation/ts-7300-mechanical.pdf)
- **TS-7300's download section**
  (http://www.embeddedarm.com/epc/ts7300-spec-d.htm)

APPENDIX D: TS-ARM SBC FEATURE MATRIX

Product	TS-7200	TS-7250	TS-7260	TS-7300	TS-7400
CPU	200 Mhz AMR920T	200 Mhz AMR920T	200 Mhz AMR920T	200 Mhz AMR920T	200 Mhz AMR920T
PC/104 connector	Yes	Yes	Yes	Yes	No
On-board FPGA	No	No	No	Yes	No
RAM	32 MB	32 MB	32 MB	32 MB	32 MB
Optional RAM	64 MB	64 MB 128 MB	64 MB 128 MB	64 MB 128 MB	64 MB 128 MB
On-board Flash	8 MB	32 MB	32 MB	No	32 MB
Optional on-board Flash	16 MB	64 MB 128 MB 256 MB	64 MB 128 MB 256 MB	Possible Contact us	64 MB 128 MB 256 MB
Standard A/D	Yes - 2 ch	Yes	Yes - 2 ch	Yes - 1 ch	Yes - 4 ch
Optional A/D	8 ch	8 ch	No	No	No
Ethernet	1x 10/100	1x 10/100	1x 10/100	2x 10/100	1x 10/100
USB1.1 ports	Yes - 2x	Yes - 2x	Yes - 2x	Yes - 2x	Yes - 2x
VGA Video Out	No	No	No	Yes	Yes
IDE Compact Flash	Yes	No	No	No	No
SD Card Interface	No	No	Yes - 1x on RevB	Yes - 2x	Yes - 1x
Digital I/O	20	20	30	55	20
TS-XDIO	No	No	Yes - 1	Yes - 2	No
RS-485	Opt. full/half	Opt. full/half	Opt. full/half	Opt. full/half	Opt. full/half
Standard COM Ports	2	2	3	10	3 TTL
Optional COM Ports	No	No	2	Yes	No
RS-232 Console	Yes	Yes	Yes	Yes	No
RTC	Opt.	Opt.	Opt	Opt.	Opt.
LCD Interface	Yes	Yes	Yes	Yes	No
Keypad Interface	Yes	Yes	Yes	Yes	No
Linux2.4	Yes	Yes	Yes	Yes	Yes
Bootloader	Redboot	Redboot	Redboot	Linux	Linux
Real-Time RTAI	Yes	Yes	Yes	Yes	Yes
Java	Yes	Yes	Yes	Yes	Yes
USB WiFi Support	Opt.	Opt.	Opt.	Opt.	Opt.
USB Flash Support	Opt.	Opt.	Opt.	Opt.	Opt.
Extended Temperature	Yes	Yes	Yes	Yes	Yes
Switching-Mode Power Supply	No	No	Yes	No	Opt.
Quantity 1 Pricing	$149.00	$149.00	$179.00	$219.00	$129.00
Quantity 100 Pricing	$119.00	$119.00	$149.00	$189.00	$99.00

APPENDIX E: CONTACT TECHNOLOGIC SYSTEMS

16525 East Laser Drive
Fountain Hills, AZ 85268
TEL 1.480.837.5200
FAX 1.480.837.5300

www.embeddedARM.com
support@embeddedARM.com

Call us Monday-Friday, **from 9 am to 5 pm**, Arizona-USA time;
or email us at any time.

Our engineers answer tech support calls and are more than happy to talk to you about your needs and help you find the best solution for your project.