OVERVIEW

The TS-7800 is a RoHS compliant Single Board Computer (SBC) based on a Marvell 500MHz ARM9 CPU with internal PCI bus and that provides a standard set of on-board peripherals such as Gigabit Ethernet, dual SATA and dual High-Speed host/slave USB 2.0. The TS-7800 also features a 12,000 LUT on-board FPGA which is user-programmable via Linux software and provides extra peripherals such as 110 GPIO lines, additional serial ports and SD card sockets.

On the software side, the TS-7800 uses an in-house improved Linux 2.6 Kernel that allows 2 seconds bootup and provides driver support for all on-board hardware. In addition, the 512 MB on-board Flash enables a full Debian distribution to be installed with a complete embedded development environment. The Eclipse IDE is also available on the TS-7800 development kit, including configured crosstool chain and build/debug environment integrated with the Debian Linux running on the TS-7800.

> **!**
>
> **Warning**
>
> The TS-7800 is a new product and is still in development. Further documentation and downloads will be made available through our website as development advances. Please make sure to visit Technologic Sysmtes periodically at **http://www.embeddedarm.com.**

STARTUP

Console and Power Up

The TS-7800 SBCs have no video controller or keyboard interface. This was done to keep the board size small and the cost low. COM1 is typically used as a console port to interface the TS-7800 to a standard terminal emulation program on a Host PC.

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 (8N1), and make sure jumper **JP2** is installed. If you are running Linux, the minicom program works well, Windows users can run the Hyperterm application. Also, the 2GB development SD Card contains a version of the **Eclipse IDE** with a serial Terminal plugin already configured to work with the TS-7800. 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.

If **JP1** is installed, than the board will attempt boot (load kernel and initial ramdisk) from the SD Card, totally independent from the on-board flash. If **JP1** is not installed, the board boots from the on-board flash. Both boot sequences are identical. Total booting from SD Card (JP1 installed) allows on-board flash recovery.

JP3 installed allows the CPU to run at 333MHz instead of 500MHz, saving power and increasing reliability in extreme thermal conditions.

Connect a regulated 5VDC, (1A minimum) power source using the included 2 screw terminal strip/connector. In case the TS-7800 is equipped with the OP-SWITCHREG option, connect a 8-30VDC power supply at the connector available through the small board (switching power regulator) soldered onto the TS-7800. Please note the polarity printed on the board. The boot messages, by default, are all displayed on COM1 at 115200 baud, N81.

Boot Sequence

The TS-7800 board has Linux installed by default on on-board flash. Upon bootup, The board will boot within 2 seconds to a Linux prompt on UART #0 (/dev/ttyS0). The serial port will be set to 115200 baud, 8N1, with no flow control by default. The board will also answer telnet connections to IP address 192.168.0.50.

The default fastboot shell has available several standard Linux commands accommodated by the "busybox" program. Technologic Systems has made several modifications to the busybox source code to keep bootup as fast and simple as possible. The modified source code is available to Technologic Systems customers at **ftp://ftp.embeddedARM.com**.

Upon bootup, you should see out of your serial port:

```
>> TS-BOOTROM - built Sep 27 2007
>> Copyright (c) 2006, Technologic Systems
.
.
.
.
Finished booting in 1.88 seconds
Type 'tshelp' for help
$
```

At this point, if you type 'exit' from the serial shell, the TS-7800 will then attempt a full Debian Linux bootup from the SD card on partition #4 (/dev/tssdcardb4). If the SD card is not present or the EXT2 filesystem does not pass a basic sanity test or the special file "/notrootfs" exists, a complete Debian Linux distribution contained on the onboard flash itself is instead booted (partition #3 /dev/mtdblock3).

Should you wish to automatically bypass the fastboot and proceed directly into starting the SD card version of Linux, you can do so with the following command issued to the fastboot shell:

```
ln -sf /linuxrc-sdroot /linuxrc; save
```

For the onboard flash the command is:

```
ln -sf /linuxrc-mtdroot /linuxrc; save
```

To get back to the fastboot shell, you can do so by placing the file "/fastboot" in the root directory of the filesystem.

The '/linuxrc' file is a shell script that is the very first thing run by the kernel on startup. Several sample startup scripts are included and can either be used directly ("ln -sf /linuxrc-XXX /linuxrc" command) or modified to include custom bootup logic. These shell scripts were designed to be as fast and simple as possible (approximately 70 lines of code) for easy customer modifications. It is anticipated that this shell script be modified from the default to implement things in the customer's product such as backup configurations, software field updates, conditional booting/verification of SD cards, etc. Technologic Systems professional services is available should you need help in implementing a specific feature.

Although it is easy to get your board into an unbootable state during development if you botch a modification, it is equally easy to use an bootable SD Card to recover the default startup and reprogram the on-board flash. To do so, place the **JP1** jumper on the TS-7800 and reset the board. The board will now completely boot from the SD Card, if the SD Card has the right partitions programmed. See the partition map below in order to create valid on-board flash and SD card devices.

On-board NAND Flash partitions programmed from the factory:

/dev/mtdblock0: the whole device
/dev/mtdblock1: 4MB for the uncompressed Linux Kernel Image
/dev/mtdblock2: 4MB for the uncompressed initial ramdisk with busybox filesystem
/dev/mtdblock3: complete Debian Linux filesystem

SD Card partitions programmed from the factory:

/dev/tssdcardb: the whole device
/dev/tssdcardb1: vfat partition with Eclipse IDE on 2GB or bigger cards or 4MB empty
/dev/tssdcardb2: 4MB for the uncompressed Linux Kernel Image
/dev/tssdcardb3: 4MB for the uncompressed initial ramdisk with busybox filesystem
/dev/tssdcardb4: complete Debian Linux filesystems

DEBIAN LINUX

Debian is installed by default on the 512MB on-board flash and also on the SD Card provided by Technologic Systems. 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-7800 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. 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.

When using Debian, 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. "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

DEVELOPMENT WITH ECLIPSE: 2GB DEVELOPMENT SD CARD

A Eclipse IDE configured for embedded development with TS-7800 is provided along with the 2GB SD Card. Technologic Systems makes use of the Eclipse Europa release and the DSDP+CDT plugins in order to provide an advanced IDE. An ARM crosstool chain and a version of Java are installed on the card, therefore Eclipse IDE will run out-of-the-box. The Eclipse environment is already configured to allow a quick startup with TS-7800 embedded development. The 'helloworld' example shows the user how to use the Eclipse functionalities.

Follow the next steps to get Eclipse running:

1) In order to run the Eclipse Development environment for the TS-7800, double click 'runEclipse-Win.bat' file at the development SD Card root directory.

2) Inside Eclipse, a terminal is configured to

3) Connect the DB9 and Ethernet cables between your TS-7800 and Host PC.   Apply power to the TS-7800. See the TS-78000 booting messages using the Eclipse IDE

(last 'Terminal' tab on the Eclipse interface).

The TS-7800 will boot by default to a Linux initial ramdisk, which enables fast bootup time and provides the user with earlier control over the Linux system. Full Debian Linux can then be started from any source, including SD Card, onboard flash and NFS root. Type 'tshelp' at the shell prompt to find more information. In order to initiate full Debian bootup, type 'exit'.

helloworld.c file

The following is the source code for the "helloworld.c" program that is configured to be compiled and debugged with the Eclipse IDE for TS-7800. This helloworld program will be available to be compiled/debugged within Eclipse IDE after it is loaded. Follow the steps described in the source code in order to use helloworld, Eclipse and the TS-7800.

```
/*
 * TS-7800 Eclipse-based Development Environment - HelloWorld Example
 *
 * Getting Started: Connect the DB9 and Ethernet cables between your TS-7800
 * and Host PC. Apply power to the TS-7800. See the TS-78000 console messages
 * at the last 'Terminal' tab below.
 *
 * The 'Remote Systems' tab below allows you to connect to the TS-7800 target
 * either via SSH or FTP. The first 'Terminal' tab performs telnet to the target.
 * The second 'Terminal' tab is the serial console configured to COM1/115200/N81.
 * The following is the default TS-7800 Linux configuration:
 * Default TS-7800 target IP: 192.168.0.50
 * Default TS-7800 eclipse user/password: eclipse/eclipse
 * User 'root' has blank password, allowing login via telnet or console
 *
 * This Eclipse environment is already configured to use a correct ARM crosstool.
 * In order to build this example, right-click 'helloworld' at Project Explorer ->
 * Build Configurations -> Build All. Binaries will appear under 'Binaries'.
 *
 * The Eclipse interface requires Debian Linux to be running on the TS-7800.
 * If you are booting to the fastboot Linux filesystem, type the 'exit' command to
 * initiate complete Debian Linux boot.
 *
 * Follow the steps below to compile and run helloworld:
 * 1) run Debian Linux on the TS-7800 (type 'exit' on the console 'Terminal')
 * 1) play with the 'Remote Systems' tool in order to enabling FTP and SSH
 *    connection to the TS-7800 and to browse the root filesystem.
 * 2) build the 'helloworld' project and copy the binary to the eclipse home
 *    (drag the file to the desired folder) at the TS-7800 target via FTP or SSH
 *    using the Eclipse 'Remote Systems' tool.
 * 3) open a telnet session using the 'Terminal' tool, login as eclipse
 * 4) set helloworld as executable: 'chmod 755 /home/eclipse/helloworld'
 * 5) execute helloworld: ./helloworld
 *
 * Follow the steps below to debug helloworld using gdb-server:
 * 1) compile and copy the debug helloworld binary to the target
```

```
* 2) using a telnet session, run gdb-server:
*    gdbserver localhost:10000 /home/eclipse/helloworld
* 3) run the 'helloworld Debug' option by clicking the bug icon on the top bar
* 4) play with the Debug perspective in order to debug helloworld
*
* Refer to the Eclipse/DSDP documentation for further information:
* http://www.eclipse.org/
*
*/

#include <stdio.h>
#include <stdlib.h>

int main ()
{
  char *s;
  int i;

  s = "TS-";
  i = 7800;
  printf ("Hello World %s%d!\n", s, i);

  return EXIT_SUCCESS;
}
```

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

DOCUMENT HISTORY

10.09.2007 – CREATED
07.09.2009 – Updated Mailing address