

### Connecting serial communications and power (JP2 must be installed to enable console)

An ANSI terminal or a PC running a terminal emulator is required to communicate with your Embedded PC. The serial parameters are **115200 baud**, 8 data, and no parity. If you are running Linux, the minicom program works well, Windows users can run the Hyperterm application.

Connect the DB9 female connector labeled COM1 to a null modem cable. Connect the other end of the null modem cable to a serial port on the host. 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

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 sequence has four distinct stages, TS-BootROM messages, RedBoot ROM monitor messages, Linux Kernel messages and login prompts. Messages, by default, are all displayed on COM1 at 115200 baud.

### Startup sequence

Apply 5VDC power. The board mounted LED will blink, followed immediately by the display of TS-BootROM messages and RedBoot messages

RedBoot, if not interrupted by the user within one second, loads the Linux kernel from flash and boots to the onboard flash chip.

The file system loads, networking, logging, Apache web server, etc. are started.

The login prompt is displayed, type "**root**" to login, with no password. A Bash login prompt will then appear.

### WARNING

Ensure that JP5 is removed when booting from the default flash file-system. Special scripts look for it and run special test programs that will use up system resources.

## Booting

Upon power up, the board executes proprietary Technologic Systems boot-code, then immediately executes RedBoot. RedBoot is a feature rich boot-ROM monitor, that allows manipulation of the on-board flash, JFFS2 images, loading and execution of a kernel or executable from either tftp (trivial ftp), serial console, or from flash, and offers gdb debugging stubs. By default, a pre-existing RedBoot script is executed, if not interrupted by the user within one second a Linux kernel is loaded into memory from flash, booting into the pre-existing JFFS2 file-system. One can view the RedBoot defaults for the board, as well as the default script, by entering at the RedBoot command prompt:

```
fconfig -l
```

The defaults can be changed by simply entering

```
fconfig
```

at the RedBoot prompt and answering the prompts. A final chance to write or discard the changes to the board will be given by RedBoot.

The default script instructs RedBoot to load the Linux kernel from the flash, and instruct the Linux kernel to use the JFFS2 image on the flash chip for its root file-system. The Linux **kernel must be loaded into memory address 0x00218000**. Loading the kernel from flash is done automatically by RedBoot in the default script with the following command:

```
fis load vlinux
```

After loading the kernel, the default script then executes the kernel with the following command:

```
exec -c "console=ttyAM0,115200 root=/dev/mtdblock1"
```

## Loading and executing Kernels from RedBoot

RedBoot can load a kernel or executable in one of three ways. It can load an executable from the serial console, a tftp server, or directly from flash. The default boot-script loads the kernel from flash into memory by executing

```
fis load vmlinux -b 0x00218000
```

To load an image from a TFTP server, use the following command

```
load -b 0x00218000 -h 192.168.0.1 vmlinux
```

the `-h` option specifies the IP address of your TFTP server

To execute a Linux kernel from RedBoot, the following command is used

```
exec -c "console=ttyAM0,115200 root=/dev/mtdblock1"
```

the `-c` option passes the Linux kernel any specified kernel arguments.

## TCP/IP configuration for JFFS2 file system in on-board flash

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

```
DEVICE=eth0           # Name of ethernet interface
IPADDR=192.168.0.50   # Internet Protocol address of this ethernet interface.
NETMASK=255.255.255.0 # Used with NETWORK to determine which IP's are local.
NETWORK=192.168.0.0   # Used with NETMASK to determine which IP's are local.
BROADCAST=192.168.0.255 # Broadcast IP for system wide messages.
BOOTPROTO=static      # Static IP (change 'static' to 'DHCP')
ENABLE="yes"          # Load device on boot.
```

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

```
NETWORKING=yes        # Enable networking on startup
GATEWAY="192.168.0.1" # Gateway for internet access.
GW_DEV=eth0           # Default gateway
HOSTNAME=ts7200       # Host name of this computer.
BOOTPROTO=no
FORWARD_IPV4=no
DEFRAG_IPV4=no
```

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

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

To access the web server, open a web browser and enter "192.168.0.50" as the address.

(The development system on 256 MB Compact Flash is Debian 3, see [www.debian.org](http://www.debian.org) for info.)

## Linux utilities

```
/sbin/ifconfig      Display information about ethernet settings.
/usr/bin/df          Display disk space usage.
/usr/bin/free        Display memory usage.
```

Most of the Linux utilities you would expect are present on the file-system. In total there are over 120 utilities installed on the file system. Many of the basic utilities are implemented using Busybox, check

## Linux notes

After the kernel is done booting the 'init' process reads the file '/etc/inittab'. The inittab file will call /etc/rc.d/rcS.sysinit' as part of the system initialization. The runlevel 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. A halt or reboot command will change the runlevel to 0 or 6 execute the '/etc/rc0.d' scripts or 'etc/rc6.d' scripts respectively.

**NOTE:** Use the "shutdown -h now" command to halt the Linux system when running from Compact Flash to avoid a potentially lengthy file system check on the next boot. The JFFS2 file system is highly tolerant of power cycles while the file system is mounted. The 'shutdown' command is not required when the root file system is JFFS2, but is still recommended.

## **Booting to other media from RedBoot**

### **Compact Flash**

To boot the compact flash card, use the following after loading the kernel into memory:

```
exec -c "console=ttyAM1,115200 root=/dev/hda1"
```

### **NFS root file system**

NFS uses the Ethernet connection to mount the root file system for the SBC as an exported directory on a Linux host PC. This bypasses the need to download any files from the host to the SBC, because any file needed on the SBC is just copied to the exported directory on the host system and is instantly available to the embedded PC.

Mounting the root file system via NFS allows the developer to use the editors, compilers, etc. from the single board computer. After loading the Linux kernel, the following command executes the kernel and passes the appropriate arguments ( substitute the appropriate values for your server):

```
exec -c "console=ttyAM1,115200 ip=dhcp nfsroot=192.168.0.1/path/to/nfsroot"
```

## **Compact Flash Card**

The Development 256 MB Compact Flash card is pre-installed with Debian stable. Along with the basic core utilities, some developer tools have been installed, including a native arm gcc toolchain. Debian utilities, such as apt-get and dpkg, work as expected. For more information regarding Debian, please visit the Debian home-page at [www.debian.org](http://www.debian.org)

## **Cross Toolchains**

For cross platform development, the use of Dan Kegels' crosstools is recommended (<http://kegel.com/crosstool/>). The pre-compiled tar files for Cygwin (Windows) or Linux is available on the developers CD as well as on our web-site ([www.embeddedARM.com](http://www.embeddedARM.com)).

## mounting NFS root file system

For detailed information on how to setup an NFS server, please see <http://nfs.sourceforge.net/nfs-howto/server.html>

To mount the root file system via NFS:

=====

Prepare the exported root directory on the NFS server

=====

1. Download the pre-made Debian tarfile from either our website or from the Arm Linux Developers CD to your Linux host machine. The file is named `tslinux6-8-24-2004.tar.bz2`

2. Untar NFS root package to a directory on the host machine.

```
tar -xvjf tslinux6-8-24-2004.tar.bz2
```

The directory is `'/home/dev/nfsroot'` in this example.

3. Export the directory by adding the following line to the file `'/etc/exports'`

```
/home/dev/nfsroot/ 192.168.0.0/255.255.255.0(rw,no_root_squash,insecure)
```

Note: The IP mask in the above example will only allow NFS connections from computers having an IP starting with 192.168.0. Substitute the appropriate parameters for your local network. See the link mentioned above for more information

4. Restart NFS so the directory is available for export. Typically, this would be:

```
/etc/init.d/nfs-server restart
```

Note: This may be different on your host machine.

5. Modify the `/home/dev/nfsroot/fstab` file on your host system for your local network settings.

```
Add:          "192.168.0.1:/home/dev/nfsroot/    /      nfs      exec,dev,suid  1 1"
Comment out:  "/dev/hda1                          /      ext2     defaults      1 1"
```

6. Modify `/mike/nfs/etc/sysconfig/ifcfg-eth0` to disable DHCP

```
set BOOTPROTO=none
```

7. Load the kernel from Redboot with the following command line options:

```
fis load vmlinux
exec -c "console=ttyAM1,115200 ip=dhcp nfsroot=192.168.0.1:/home/dev/nfsroot/"
```

## Sources of information and document history

TS-Linux for ARM developers manual  
=====

Read the entire manual for detailed information on drivers, file systems, example code, etc.

[http://www.embeddedARM.com/downloads/Manuals/TS\\_ARM\\_L.PDF](http://www.embeddedARM.com/downloads/Manuals/TS_ARM_L.PDF)

### FURTHER SOURCES OF INFORMATION

=====

In the kernel source tree:

`/Documentation/nfsroot.txt`

On the web:

<http://www.linuxdoc.org/HOWTO/NFS-HOWTO/index.html>

<http://www.linuxdoc.org/HOWTO/mini/NFS-Root.html>

<http://www.linuxdoc.org/HOWTO/mini/NFS-Root-Client-mini-HOWTO/index.html>

<http://sources.redhat.com/ecos>

<http://www.kernelnewbies.org>

<http://www.arm.linux.org.uk>

Technologic Systems support pages:

<http://embeddedx86.com/linux/ARM.htm>

Email: [support@embeddedARM.com](mailto:support@embeddedARM.com)

### DOCUMENT HISTORY

=====

08.25.2004    LAY    Created

11.10.2004    MTJ    Added JP2 comment, minor touch up.