

VIPER / VIPER-Lite

AEL Embedded Linux
Development Kit

Quickstart Manual



Disclaimer

The information in this manual has been carefully checked and is believed to be accurate. Eurotech Ltd assumes no responsibility for any infringements of patents or other rights of third parties, which may result from its use.

Eurotech Ltd assumes no responsibility for any inaccuracies that may be contained in this document. Eurotech Ltd makes no commitment to update or keep current the information contained in this manual.

Eurotech Ltd reserves the right to make improvements to this document and/or product at any time and without notice.

Warranty

This product is supplied with a 3 year limited warranty. The product warranty covers failure of any Eurotech Ltd manufactured product caused by manufacturing defects. The warranty on all third party manufactured products utilised by Eurotech Ltd is limited to 1 year. Eurotech Ltd will make all reasonable effort to repair the product or replace it with an identical variant. Eurotech Ltd reserves the right to replace the returned product with an alternative variant or an equivalent fit, form and functional product. Delivery charges will apply to all returned products. Please check www.eurotech-ltd.co.uk for information about Product Return Forms.

Trademarks

Linux is a registered trademark of Linus Torvalds.

RedBoot, Fedora and Red Hat are registered trademarks of Red Hat Inc. This product contains a copy of the installation media for the Fedora Core Linux distribution. This media is not a product of Red Hat, Inc. or the Fedora project and is not endorsed by Red Hat, Inc. or the Fedora project. It is a product of Eurotech Ltd and we have no relationship with Red Hat, Inc. or the Fedora project. The media is identical in every respect to the standard Fedora Core install media.

ARM and StrongARM are registered trademarks of ARM Ltd.

Intel and XScale are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

CompactFlash is the registered trademark of SanDisk Corp.

All other trademarks and copyrights referred to are the property of their respective owners.

This product includes software developed by the University of California, Berkeley and its contributors.

All other trademarks recognised.

Revision History

<i>Manual</i>	<i>PCB</i>	<i>Date</i>	<i>Comments</i>
Issue A		18th Aug 2006	First release of manual.
Issue B		25th Sep 2006	Update for Development Kit's Ethernet cable contents and corresponding changes.
Issue C		16th October 2007	Minor updates, Eurotech Ltd rebranding.

© 2007 Eurotech Ltd.

For contact details, see page [31](#).



ISO 9001
FM12961

Contents

Introduction	4
Handling your board safely	4
About this manual	5
Related documents	5
Conventions	6
Getting started	7
What's in the kit?	7
What else do I need?	8
What peripherals are supported?	9
Unpacking and connecting up	10
Power supply	13
RedBoot	14
What is RedBoot?	14
AEL Embedded Linux	16
What's in AEL Embedded Linux?	16
Booting the system	16
Logging in	17
Setting up networking	18
The file system	19
Linux kernel	20
Hardware configuration	21
Software configuration	28
Examples for VIPER / VIPER-Lite	29
Target support software	30
Appendix A – Contacting Eurotech Ltd	31
Index	32

Introduction

The VIPER / VIPER-Lite is an ultra low power PC/104 form-factor single board computer based on the Intel 400MHz PXA255 (VIPER) or 200MHz PXA255 (VIPER-Lite) XScale processor. The PXA255 is an implementation of the Intel XScale micro-architecture combined with a comprehensive set of integrated peripherals including:

- Flat panel graphics controller.
- Interrupt controller.
- Real time clock (RTC).
- Multiple serial ports.

The VIPER / VIPER-Lite board offers many features that make it ideal for power-sensitive embedded communications and multimedia applications.

The AEL Embedded Linux Development Kit provides a ready-to-run platform for Linux application development and deployment. AEL Embedded Linux is based on the *standard* v2.6 kernel and has been configured to load and run from embedded Flash memory.

Handling your board safely

Anti-static handling

The SBC and other circuit boards contain CMOS devices. These could be damaged in the event of static electricity being discharged through them. Please observe anti-static precautions at all times when handling circuit boards. This includes storing boards in appropriate anti-static packaging and wearing a wrist strap when handling them.

Packaging

Should a board need to be returned to Eurotech Ltd, please ensure that it is adequately packed, preferably in the original packing material.

About this manual

This manual gets you started using the VIPER / VIPER-Lite with RedBoot and AEL Embedded Linux. It lists the items you should find in your kit, and tells you how to unpack, connect and configure the board.

AEL Embedded Linux for VIPER / VIPER-Lite is based on the *standard* v2.6 kernel and includes the essential features for embedded communication and display solutions. Eurotech Ltd includes dedicated drivers for the on-board hardware and uses the embedded Flash memory to store applications.

Related documents

Manuals that you may need are provided in the \manuals section on the CD-ROM that accompanies your Development Kit. These include the VIPER / VIPER-Lite Technical Manual, the AEL Embedded Linux Technical Manual and the HOWTO documents. Most of the HOWTO documents relate to common Linux tasks. Others have been written specifically for features on the VIPER / VIPER-Lite.

Detailed information about AEL Embedded Linux is provided in the AEL Embedded Linux Technical Manual. The AEL Embedded Linux Technical manual includes information about the following:





- Configuration files and boot scripts.
- Making an application run automatically at boot.
- AEL Embedded Linux contents and configuration.
- Installation.
- Example programs and utilities.
- Secure Shell (SSH).
- Compiling a kernel.

Please note that the AEL Embedded Linux Technical Manual is a generic manual for AEL Embedded Linux supplied on Eurotech Ltd boards. As such, the AEL Embedded Linux manual uses conventions and examples that may or may not be specific to your board. It is advised that the AEL Embedded Linux manual be used in conjunction with the instructions and information supplied in the Quickstart and Technical manuals for your board.

Conventions

Symbols

The following symbols are used in this guide:

Symbol	Explanation
	Note - information that requires your attention.
	Tip - a handy hint that may provide a useful alternative or save time.
	Caution - proceeding with a course of action may damage your equipment or result in loss of data.
	Indicates that a feature is <i>not</i> available on the standard VIPER-Lite configuration. Eurotech Ltd can provide custom configurations (subject to a minimum order quantity) for the VIPER-Lite populated with this feature. Please contact our Sales team (see Appendix A – Contacting Eurotech Ltd , page 31) to discuss your requirements.

Typographical conventions

This manual contains examples of commands that you can enter. These are shown as follows:

\$ make install DESTDIR=/tmp/target-install

The initial symbol (\$ in this case) indicates the prompt that the command is for, and should not be typed.

The prompts used are explained in the following table:

Prompt	Explanation
\$	Linux (bash shell) as a regular user.
#	Linux (bash shell) as root.
RedBoot>	RedBoot command line.

Different fonts are used throughout the manual to identify different types of information, as follows:

Font	Explanation
<i>Italics</i>	Parts of a command that should be substituted with appropriate values.
Bold	Information that you enter yourself.
Screen text	Information that is displayed on screen.

Getting started

What's in the kit?

The VIPER / VIPER-Lite Development Kit includes:

- VIPER-M64-F32, VIPERL-M64-F32 or VIPERL-M64-F16.
- Optional GPIO based VIPER-I/O board (technical manual included on CD)*.
- Standard serial port breakout cable (up to five ports).
- RS-232 serial cable (2m).
- RS-232 null modem adapter.
- Flat panel display cable.
- VIPER FPIF-CRT (breakout assembly with standard CRT socket).
- Ethernet breakout assembly with standard RJ45 socket.
- Ethernet crossover cable.
- GPIO cable with un-terminated ends.
- Reset switch.
- +5V @ 4A max (100-240 VAC input) power supply.
- PC/104 mounting pillars (two sets).
- Development Kit support and installation CD.
- Fedora Core 5 installation DVD¹.
- AEL Embedded Linux Technical Manual (on CD).
- ⓧ • USB breakout cable.
- ⓧ • Audio breakout cable.

* Development Kit option

¹ fedora.redhat.com/

What else do I need?

An x86 Linux system compliant with version 1.3 of the Linux Standard Base² is required for the AEL Embedded Linux host environment. Suitable systems include PCs with Debian GNU/Linux, RedHat Linux, Fedora Core or SUSE Linux. The Fedora Core installation DVD is supplied for this purpose.

In addition, the following information and equipment may be useful:

- Networking details for the VIPER. If your network does not have a DHCP server, ask your network or system administrator for the following:
 - IP address and subnet mask (netmask).
 - Default gateway IP address (if required).
 - Name server (DNS) IP address (optional).
- Networking details for the host machine.
- Standard Ethernet cable to connect VIPER / VIPER-Lite to a switch or hub. Alternatively, for standalone systems, the board can be connected directly to your host machine using the crossover cable provided in the Development Kit.

For more information on setting up a network connection, see the section [Setting up networking](#), page [18](#).

² www.linuxbase.org

What peripherals are supported?

The following peripherals/features are supported in this release:

- Five UARTs, COM1 through COM5, including the processor's internal FFUART, BTUART and STUART. The VIPER-Lite is only fitted with three COM ports.
- On-board Flash array.
- CompactFlash+.
- Ethernet interface.
- A variety of flat panels running with screen resolutions of 320 x 240 or 640 x 480.
- Power management.
- GPIO.
- Real time clock.
- SRAM.
- USB client.
- ⓧ VL • Audio codec.
- ⓧ VL • PC/104 bus. Drivers are available for a range of Eurotech Ltd and third party PC/104 boards.
- ⓧ VL • USB host controller.
- ⓧ VL • RS-422 and RS-485 serial port support.

Unpacking and connecting up

This section guides you, step-by-step, through the process of connecting up the VIPER / VIPER-Lite for the first time and powering up the system. The VIPER / VIPER-Lite is delivered ready to run. There are no link settings or configuration switches that need to be adjusted for the board to operate.

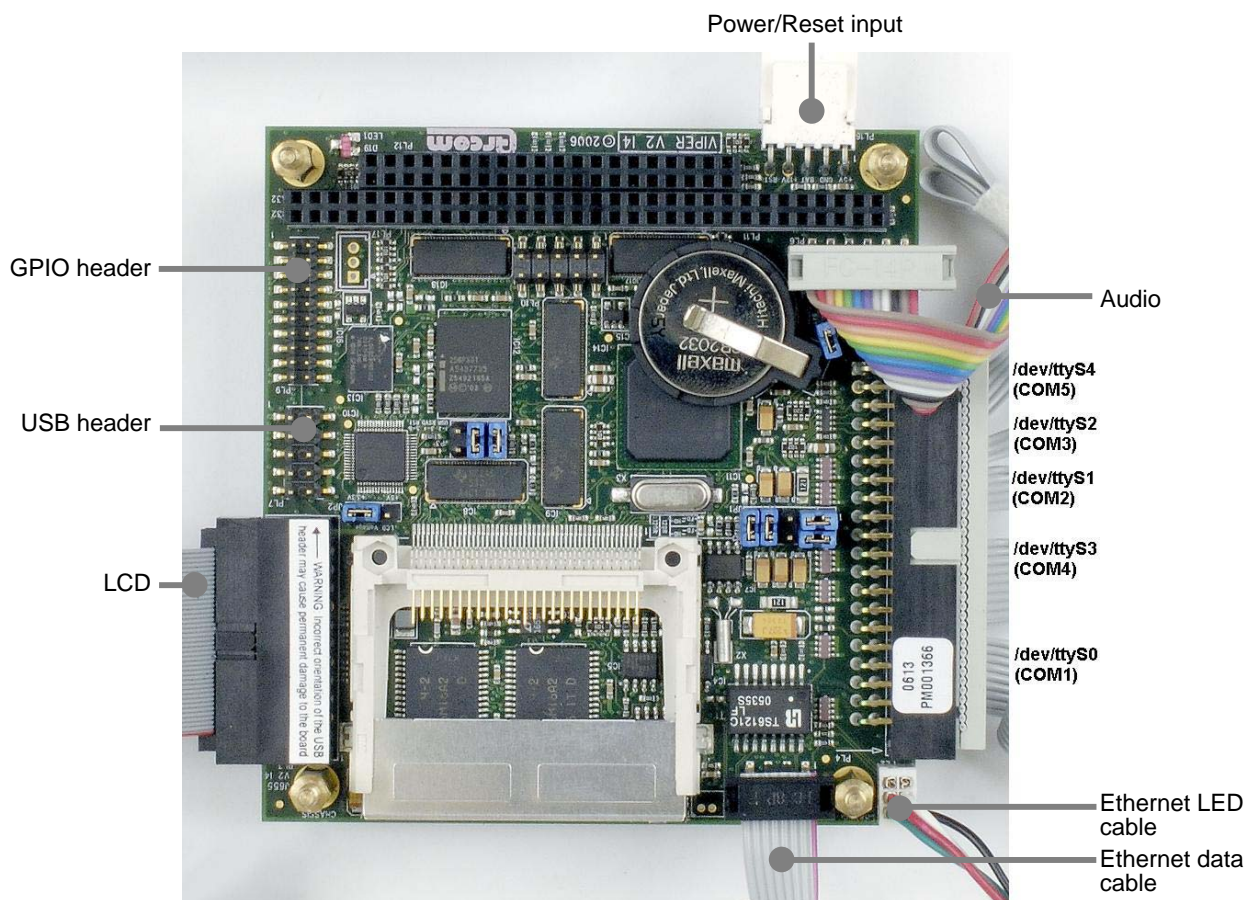


A number of the components are static-sensitive, so standard anti-static precautions must be taken during use. For more information, see [Anti-static handling](#), page 4.

If you follow the procedure described below, the board powers up automatically and loads the RedBoot boot loader from the on-board Flash array. At this point you can interrupt the boot process, if you wish, and access the Redboot command prompt. If you do not interrupt the normal boot sequence, after a 1 second pause, the system continues to load and boot AEL Embedded Linux.

To power up the VIPER / VIPER-Lite and boot AEL Embedded Linux, follow these steps:

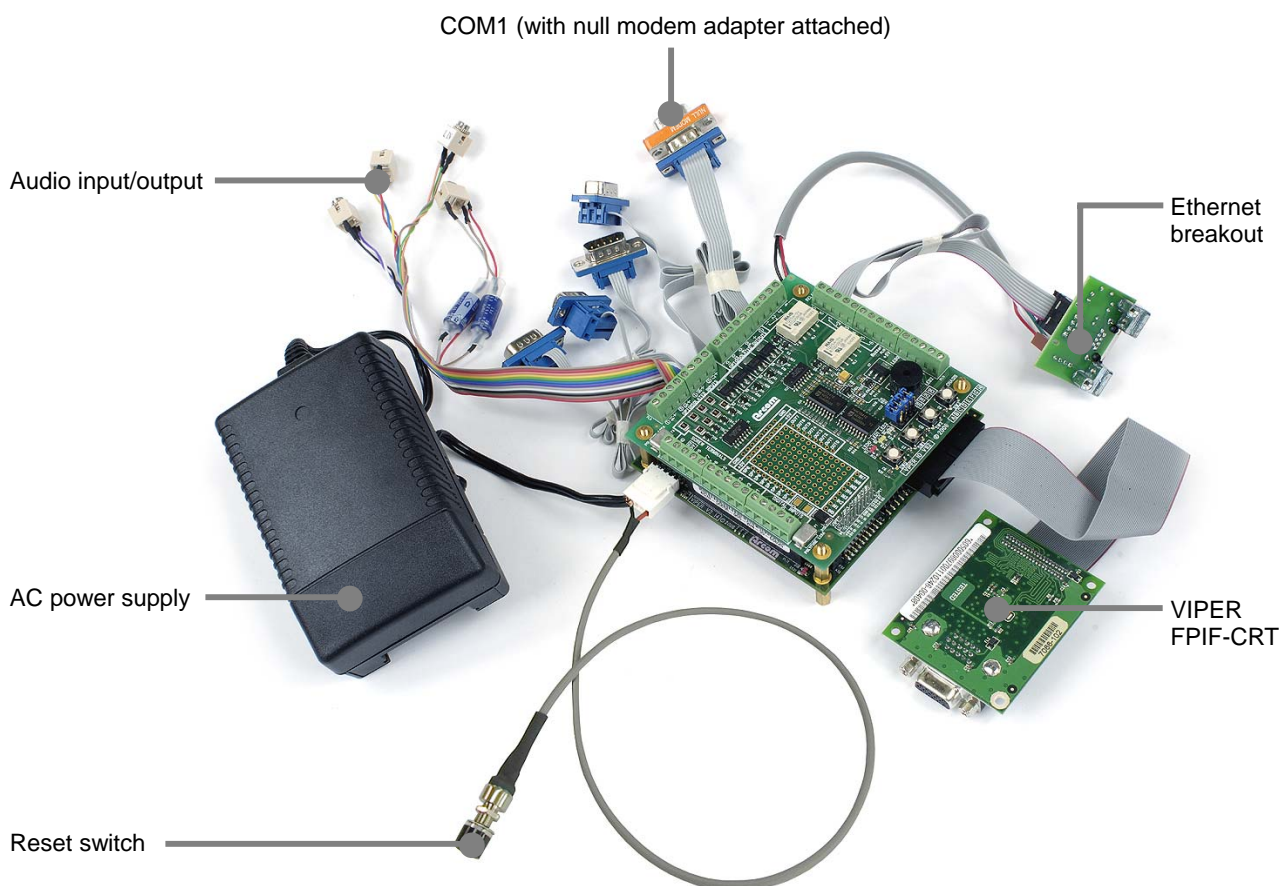
- 1 Remove the VIPER / VIPER-Lite board and all other cables from its packaging. If it shows any signs of damage during transit, please contact Eurotech Ltd before proceeding.





For all connections, the red wire on the cable is pin 1. Pin 1 on each connector is marked on the board.

- 2 The VIPER I/O board can be attached above the main VIPER / VIPER-Lite board once all connections to the main board are complete.



- 3 Fit the VIPER / VIPER-Lite power supply cable to the plug on the board.
- 4 Connect a serial terminal emulator³ to COM1 using the supplied null modem adaptor and cable.
- 5 Configure the serial terminal as follows:
 - 115200 baud.
 - No parity.
 - 8 data bits.
 - No handshaking.
- 6 Switch on the VIPER / VIPER-Lite board once the terminal emulator is set up on the host PC.

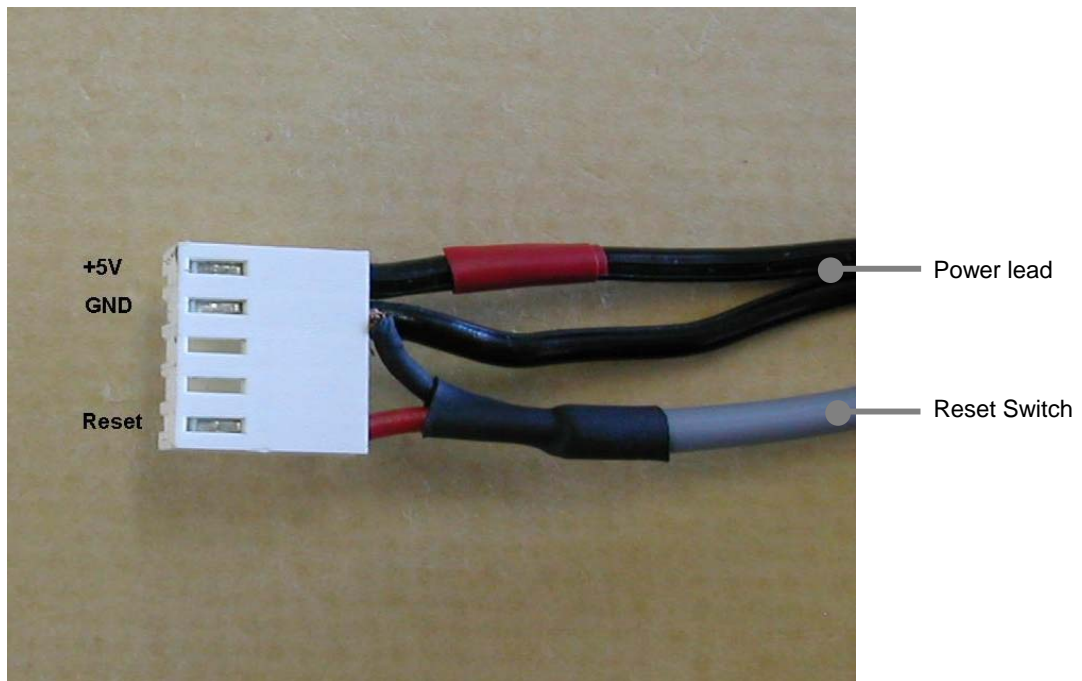
The Linux operating system is loaded from the on-board Flash memory, and the board displays a Linux prompt. For a description of AEL Embedded Linux see page 16 of the Technical Manual, which is in the /manuals/ folder on the Development Kit CD.

³ Such as 'minicom' running on a Linux PC or 'Hyper Terminal' on a Windows PC.

Power supply

The AC Power Supply Unit (PSU) supplied in the kit is rated at 4A @ +5V DC. This is designed to support the requirements of the VIPER / VIPER-Lite and VIPER-I/O. If additional devices are powered from this supply, ensure that the overall current rating is not exceeded.

The following picture shows the PSU connector:



The power connector is polarised to prevent incorrect attachment to the VIPER / VIPER-Lite board.

RedBoot

What is RedBoot?

RedBoot is a complete bootstrap environment for embedded systems. Based on the eCos Hardware Abstraction Layer, RedBoot inherits the eCos qualities of reliability, compactness, configurability, and portability.

RedBoot enables download and execution of embedded applications via serial or Ethernet, including embedded Linux and eCos applications. It can be used for both product development (debug support) and in deployed products in the field (Flash update and network booting).

Ethernet download and debug support is included. This means RedBoot can retrieve its IP parameters via BOOTP or DHCP, and program images can be downloaded using TFTP or HTTP. Images can also be downloaded over serial (using X- or Y-modem) or loaded from a JFFS2 file system on the on-board Flash array.

An interactive command-line interface is accessible via serial or Ethernet and allows management of Flash images, image download, RedBoot configuration, etc. For unattended or automated startup, boot scripts can be stored in Flash memory. This allows, for example, loading of images from Flash or a TFTP server.



You cannot use the command-line interface via Ethernet if you have an automatic boot script enabled to launch an application or another operating system. RedBoot does not stop to listen for TCP/IP activity.

The AEL Embedded Linux Technical Manual contains details of many of the RedBoot commands that are useful when using RedBoot as a boot loader for an operating system (such as Linux) or to launch an application. Full RedBoot documentation is available in the eCos Reference Manual, which is in the `/manuals/` folder on the Development Kit CD.

Default boot script

The default boot script on the VIPER / VIPER-Lite is configured to load a Linux kernel from the JFFS2 file system and execute it.

The default boot script for VIPER-Lite is as follows:

```
clock -l 27 -m 2 -n 10
mount -t jffs2 -f filesystem
load -r -b ${FREEMEMLO} ${kernel}
exec -c ${cmdline}
```

The default boot script for VIPER is as follows:

```
clock -l 27 -m 4 -n 10
mount -t jffs2 -f filesystem
load -r -b ${FREEMEMLO} ${kernel}
exec -c ${cmdline}
```

The only difference is that the default boot script initialises the clock to 400MHz for VIPER and to 200MHz for VIPER-Lite. Refer to following section for more details on '**clock**' command.

`%{FREEMEMLO}` is a predefined alias for the lowest address of available memory.

`%{kernel}` is an alias for the full path name of the Linux kernel to boot. For example:

```
Redboot> alias kernel "-f /boot/bzImage"
```

`%{cmdline}` is an alias for the kernel command line to use. For example:

```
Redboot> alias cmdline "\root=/dev/mtdblock2 rootfstype=jffs2 ro console=ttyS0,115200\""
```

Refer to the AEL Embedded Linux Technical Manual for further details.

CPU core clock frequency and the clock command

Redboot (V3I7) does not change the clock rate - it remains at the default 100MHz setting. However, the **clock** command can be used to set the clock to any of the frequencies described in the section [CPU clock frequency scaling](#) on page 22. This takes parameters l, m and n correspondingly for the L, M and N settings, which control the clock speed as described.

While specifying on Redboot command-line, the value of n should be 10 times the desired value of N, i.e. to set N to 1.5 use the option **-n 15**. For example:

- To set the clock frequency to 300MHz power saving mode:

```
RedBoot> clock -l 27 -m 1 -n 30
```

- To set the clock frequency to 200MHz performance mode:

```
RedBoot> clock -l 27 -m 2 -n 10
```



Attempting to run a VIPER-Lite board above 200MHz or a VIPER board above 400MHz may damage your board permanently.

AEL Embedded Linux

What's in AEL Embedded Linux?

AEL Embedded Linux is a space-optimised Linux distribution based on Linux kernel version 2.6. AEL Embedded Linux includes many of the standard file, shell and text utilities. As well as the basic Linux system, the default installation includes:

- OpenSSH Secure Shell.
- Minimal X Window System.
- Web server.

For a complete list of software and an explanation of how to add and remove packages please consult the AEL Embedded Linux Technical Manual.

Booting the system

Once the system has been connected and powered up (as described in [Unpacking and connecting up](#), page [10](#)) the board automatically boots to a Linux prompt, as shown below.

The example below is typical for a VIPER-M64-F32 board:

```
Uncompressing Linux..... done, booting the kernel.
...
CPU: XScale-PXA255 [69052d06] revision 6 (ARMv5TE)
...
Machine: Arcom VIPER SBC
Memory policy: ECC disabled, Data cache writeback
Memory clock: 99.53MHz (*27)
Run Mode clock: 398.13MHz (*4)
Turbo Mode clock: 398.13MHz (*1.0, active)
```



For VIPER-Lite, the clock shows 199.07MHz instead of 398.13MHz (as shown above).

```
...
Kernel command line: root=/dev/mtdblock2 rootfstype=jffs2 ro
console=ttyS0,115200
...
3 RedBoot partitions found on MTD device flash
Creating 3 MTD partitions on "flash":
0x00000000-0x0001f000 : "FIS directory"
0x0001f000-0x00020000 : "RedBoot config"
0x00020000-0x02000000 : "filesystem"
```



This partition table is for a VIPER-M64-F32 board (with 32MB of Flash memory). The layout is different for boards with 16MB Flash memory.


```
...
INIT: version 2.85 booting
...
Configuring network interfaces...
```



If you do not have a DHCP server and you have not configured a static IP address the board pauses at this point while it searches for a DHCP server. See [Setting up networking](#), page 18.

```
... done.
...
INIT: Entering runlevel: 3
Starting system log daemon: syslogd.
Starting kernel log daemon: klogd.
Starting NFS common utilities: statd.
Loading /etc/console/keymap.gz
Starting internet superserver: inetd.
Starting http server: thttpd.
Starting OpenBSD Secure Shell Server: sshd.
Starting periodic command scheduler: cron.
Arcom Embedded Linux (ttyS0)
vipер login:
```

Logging in

You can login to the VIPER / VIPER-Lite as root (the superuser or administrator) using the following details:

Login name: **root**

Default password: **arcom**

Alternatively you can log into a user account called arcom using the following details:

Login name: **arcom**

Default password: **arcom**

Setting up networking

As supplied, the board uses the DHCP protocol to configure the network. This requires a DHCP server to be available on your network. For more information, please consult your network administrator.

To configure the VIPER / VIPER-Lite to use a static IP address, follow these steps:

- 1 Log in to the VIPER / VIPER-Lite by entering the following log in details:

Login name: **root**

Password: **arcom**

- 2 Edit the file `/etc/network/interfaces` using the nano editor.

nano /etc/network/interfaces



The vi editor is also installed, in case you prefer to use this.

- 3 Modify the **iface eth0 inet dhcp** line to look like the following (substituting your own IP address and netmask):

```
iface eth0 inet static
address 10.7.28.2
netmask 255.255.0.0
gateway 10.7.1.1
```

- 4 Save the file by holding down **Ctrl** and pressing **O**.
- 5 Exit by holding down **Ctrl** and pressing **X**.
- 6 Optional. Set up a Domain Name System (DNS) server if required. You need a DNS server if you want to use domain names (such as www.eurotech-ltd.co.uk) when accessing the Internet.

To set up a DNS server, edit the file `/etc/resolv.conf` and add your DNS server address as follows:

```
nameserver 10.7.1.9
```

- 7 Reinitialise the networking by typing:

```
# /etc/init.d/networking restart
```

The following is displayed:

```
Reconfiguring network interfaces...done
```

For standalone systems where the VIPER / VIPER-Lite is connected directly to the host system through an Ethernet crossover cable, you can modify the settings of the VIPER / VIPER-Lite and the host system, similar to the example shown above, to set up a network between them.

Chapters 25 and 26 of the RUTE Manual provide a useful introduction to TCP/IP networking. This manual is on the Development Kit CD, in the `/manuals/` folder.

The file system

The file system used on the VIPER / VIPER-Lite Flash memory is a compressed Journaling Flash File System (JFFS2). JFFS2 has the following advantages:

- Robustness.
- Power fail safe.
- No integrity scan (fsck) is required at boot time.
- Flash wear levelling.
- Transparent compression, with a typical compression ratio of 2:1.

The on-board Flash is partitioned using RedBoot FIS, as shown in the following table:

Partition #	Block device	Size (KB)	Offset (KB)	Description
0	/dev/mtdblock0	124	0	Partition table
1	/dev/mtdblock1	4	124	RedBoot configuration
2	/dev/mtdblock2	remainder	128	Root file system



Previous versions of AEL Embedded Linux on the VIPER / VIPER-Lite made use of a separate kernel partition. From version 4 issue 1 onwards, the kernel is stored as a regular file in the /boot/ folder on the root file system.

In addition, a 4MB RAM-based file system is mounted on /var/tmp for temporary files. These are lost on reboot.

The basic AEL Embedded Linux install uses about 13.9MB of physical Flash. On a board with 32MB of Flash memory fitted, this leaves about 18.1MB free. This is about 36MB of free space, assuming a typical 2:1 compression ratio. For a board fitted with 16MB of Flash memory, there is approximately 2MB of free space. This is about 4MB of free space, assuming the same compression ratio.

You can make additional space available by removing unused packages. For example, if you do not require video support, removing *xorg* and its components could release about 2MB. For more information on removing packages, refer to the AEL Embedded Linux Technical Manual, which is in the /manuals/ folder on the Development Kit CD.

The default Flash image is on the Development Kit CD, in the /images/ folder. For an explanation of how to reload a complete Flash image, see the AEL Embedded Linux Technical Manual.

Linux kernel

The version of the Linux kernel used on the VIPER / VIPER-Lite is a standard Linux 2.6 kernel tree with patches to provide support for Eurotech Ltd's boards.

Source and configuration

The kernel sources are installed in the host environment as /opt/arcom/src/linux-source-VERSION.tar.gz.

Once you have unpacked the kernel source you can configure the default VIPER / VIPER-Lite kernel with:

```
$ make ARCH=arm viper_defconfig
```

The kernel may now be built and installed as described in the AEL Embedded Linux Technical Manual.

Default command line

The default kernel command line is:

```
root=/dev/mtdblock2 rootfstype=jffs2 ro console=ttyS0,115200
```

Hardware configuration

PC/104 bus

The PC/104 bus on the standard VIPER option is implemented via the second PCMCIA controller built into the PXA255 processor.

The PC/104 I/O address space is mapped at physical address 0x30000000 and virtual address 0xF1000000. This means that a PC/104 card that has been configured at I/O address 0xXXX is available at virtual address 0xF1000XXX. This virtual address is the correct value to use when configuring a device driver for that card, rather than the associated physical address.



Previous versions of AEL Embedded Linux on the VIPER using version 2.4 of the Linux kernel used 0xF7000000 as the offset for the PC/104 I/O address space.

The PC/104 memory address space is available at physical address 0x3C000000. This means that a PC/104 card that has been configured for memory address 0xXXXX is available at physical address 0x3C00XXXX. It is the responsibility of the driver to map this address into the kernel's virtual address space before use.



The 'vip' example file on the CD (in /examples/c/) may be useful for very simple user space PC/104 tests. This accesses physical PC/104 addresses via /dev/mem.

PC/104 interrupts are mapped as shown in the following table:

PC/104 interrupt	VIPER interrupt
3	104
4	105
5	106
6	107
7	108
10	109
11	110
12	111
9	112
14	113
15	114

For more information, see the VIPER / VIPER-Lite Technical Manual and PXA255 Developer's Manual.

CPU clock frequency scaling

The PXA255 processor used on the VIPER / VIPER-Lite can scale the core clock frequency from 100 to 400MHz, allowing an application to dynamically trade off performance against power consumption. This functionality is exposed in Linux via special files in the /sys system. These can be manipulated directly (for example using **cat** and **echo**), but are better controlled using the **cpufreq-info** and **cpufreq-set** utilities. Both utilities are described below.

The processor's clock and power manager, including the available Core Clock Control Register (CCCR) multiplier configurations, are described in detail in the PXA255 Processor Developer's Manual, which is in the /references/ folder on the Development Kit CD. The CCCR consists of 3 multipliers, L, M and N:

- The L multiplier converts the basic 3.6864MHz crystal frequency into the memory clock frequency. Typical values of L are 27, 36, and 45.
- The M multiplier converts the memory clock frequency into the run-mode frequency. Typical values of M are 1, 2 and 4.
- The N multiplier converts the run-mode frequency into the turbo-mode frequency. Typical values of N are 1.0, 1.5, 2.0 and 3.0.



Not all combinations of L, M and N are valid. Some valid combinations are described below, and more information is available in the PXA255 Processor Developer's Manual.

The Linux kernel divides the available clock configurations into two complementary sets of available frequencies, corresponding to the run-mode and turbo-mode multipliers. The first set is the *performance* set. These use the run-mode multiplier, CCCR[M], to scale the clock frequency. Increasing the run-mode multiplier in this way provides the best performance but at the cost of increased power consumption.

The second set of frequencies comprise the *powersave* frequencies. These use the turbo-mode multiplier, CCCR[N], to scale the clock frequency. The turbo-mode multiplier uses less power than the run-mode multiplier but with a performance cost (especially if frequent memory accesses are required), since the core must wait for these while running in turbo mode.

The Linux kernel manages the current clock configuration using a software governor combined with user-specified minimum and maximum desired clock frequencies.




The Linux kernel modules for CPU frequency scaling override the settings made by the RedBoot **clock** command. If you don't need to alter or limit the CPU frequency at runtime and would like to keep the CPU frequency set by RedBoot, then you can reconfigure and rebuild the kernel to remove this feature.

The default VIPER / VIPER-Lite kernel provides three governors:



- The *performance* governor selects the maximum allowed frequency from the processor's available performance frequencies.
- The *powersave* governor selects the lowest available frequency from the available powersave frequencies.
- The *userspace* governor allows direct userspace selection of a frequency from the available performance frequencies.

The default on the VIPER is to use the performance governor with a maximum frequency of 400MHz. The maximum frequency on the VIPER-Lite is limited to 200MHz by passing a command line parameter to the Linux kernel. Please refer to the [Software configuration](#) section on page 28 for further information. The powersave and userspace governors are supplied as loadable kernel modules, called `cpufreq_powersave` and `cpufreq_userspace` respectively. They may be loaded using the **modprobe** command.

The available performance clock rates are listed in the following table. They correspond to the modes available when the turbo mode multiplier, `CCCR[N]` is 1.0.

Clock speed (KHz)	Core Clock Configuration Register(L, M and N multipliers)	Memory clock (MHz)	Run clock (MHz)	Turbo clock (MHz)	PX bus (MHz)
99532	27, 1, 1.0	99	99	99	50
199065	27, 2, 1.0	99	199	199	99
 398131*	27, 4, 1.0	99	398	398	196

The available powersave clock rates are listed in the following table:

Clock speed (KHz)	Core Clock Configuration Register (L, M and N multipliers)	Memory clock (MHz)	Run clock (MHz)	Turbo clock (MHz)	PX bus (MHz)
99532	27, 1, 1.0	99	99	99	50
199065	27, 1, 2.0	99	99	199	50
 298598	27, 1, 3.0	99	99	299	50
 398131	27, 2, 2.0	99	199	398	99

The **cpufreq-set** utility can be used to select a governor and to set the minimum and maximum allowable frequency. This is illustrated by the following examples:

- To select the performance governor and limit the maximum allowed speed to 200MHz:
cpufreq-set --governor performance --max 199065
- To select the powersave governor and set the minimum allowed frequency to 300MHz:
modprobe cpufreq_powersave
cpufreq-set --governor powersave --min 298598
- To use the userspace governor and set a frequency of 400MHz:
modprobe cpufreq_userspace
cpufreq-set --governor userspace --freq 398131



Changing the clock frequency automatically regulates the core voltage in order to maximise power savings.

The current settings can be examined using the **cpufreq-info** utility:

```
# cpufreq-info
...
analyzing CPU 0:
...
hardware limits: 99.5MHz - 398MHz
available cpufreq governors: performance
                        current policy: frequency should be within 398MHz
                        and 398MHz. The governor "performance" may decide
                        which speed to use within this range.
current CPU frequency is 398MHz (asserted by call to hardware).
```

Sleep and wake up support

The PXA255 processor supports a low power sleep mode in which the processor shuts down entirely until some external or internal stimulus causes it to wake up. This stimulus can be either an alarm generated by the processor's internal Real Time Clock (RTC) or a change in the state of one of the processor's GPIO lines. For an explanation of how to configure the wake up source and ensure that power consumption is reduced to the minimum, see the VIPER / VIPER-Lite Technical Manual.

Once a wake up source has been configured, the processor can be put into sleep mode by writing to the special file `/sys/power/state`.

The following command, for example, would not appear to return until after the processor has been woken up:

```
# echo -n "mem" > /sys/power/state
```

If you intend to use the PXA255's internal RTC as a wake up source, you must unload the driver for the external RTC (which cannot perform the wake up function) and load the driver for the internal RTC, as follows:

- 1 Remove the external DS1307 RTC driver:
rmmod ds1307
- 2 Install the SA1100/PXA255 RTC driver:
modprobe sa1100-rtc
- 3 Resynchronise the PXA255 RTC to the correct time from the system clock:
hwclock -systohc

The source for the arcom-utils package (which is on the Development Kit CD, in the `/packages/arcom-utils` folder) contains the script **viper-sleep.sh**, which is an example of how to put a VIPER / VIPER-Lite board to sleep and wake it up again.

Display controller

The display controller driver module (pxafb) is configured by default for a 320x240 TFT display. The driver module should be loaded using the normal **modprobe** command:

```
# modprobe pxafb
```

This Development Kit is supplied with a VIPER FPIF-CRT interface module. For best results the display controller can be configured for 640x480-16 VGA mode using:

```
modprobe pxafb options=mode:640x480-16
```


Correspondingly the Screen section from /etc/X11/xorg.conf file needs to be modified to:

```
Section "Screen"
    Identifier      "Default Screen"
    Device         "PXA255 integrated controller"
    Monitor        "Generic Monitor"
    DefaultDepth   16
    Subsection "Display"
        Depth       16
        Modes       "640x480" "320x240"
    EndSubsection
EndSection
```

In order to use a flat panel display, extra parameters must be passed to the **modprobe** command, either on the command line or using one of the methods described in the AEL Embedded Linux Technical Manual. The options are given as a comma separated list of parameters. Where a parameter requires a value, it is separated from the parameter name using a colon. The available parameters are listed in the following table:



The registers referred to are described in section 7.6.4 of the PXA255 Processor Developer's Manual, in the /references/ folder on the Development Kit CD.

Parameter	Description	Default
mode	<p>The required video mode has the following format: XRESxYRES-BPP</p> <p>Where XRES is the X-axis resolution, YRES is the Y-axis resolution and BPP is the desired bit depth.</p> <p>The hardware supports multiple resolutions (which include, but are not limited to, 320x240, 640-480 800x600) and bit depths of 1, 2, 4, 8 and 16 bits. The LCD driver is known to work well in 8- or 16-bit mode.</p>	mode: 320x240-16
	 <p>Resolutions above 640x480-16 may cause video artifacts on your screen image.</p>	
	<p>The mode parameter corresponds to the LCCR1[PPL], LCCR2[LPP] and LCCR3[BPP] registers.</p>	

continued...

Parameter	Description	Default
pixclock	The pixel clock, expressed in picoseconds (10^{-12} s). This value is used in conjunction with the current memory clock rate to calculate LCCR3[PCD].	pixclock:157500
left, right, hsynclen, upper, lower and vsynclen	<p>The timing parameters. All parameters are given as a number of pixel clock ticks. The upper and lower margins should be 0 for passive (STN) displays.</p> <p>The values correspond to LCCR as follows:</p> <ul style="list-style-type: none"> • 'left' corresponds to LCCR1[BLW] + 1. • 'right' corresponds to LCCR1[ELW] + 1. • 'hsynclen' corresponds to LCCR1[HSW] + 1. • 'upper' corresponds to LCCR2[BFW]. • 'lower' corresponds to LCCR2[EFR]. • 'vsynclen' corresponds to LCCR2[VSW] + 1. <p>Several of these values are modified by + 1 because the hardware expects the desired value – 1 to be programmed. The values given to this parameter therefore correspond to the desired value rather than the value programmed into the hardware.</p>	left:7, right:13, hsynclen:63, upper:1, lower:1, vsynclen:20.
colour or mono	Configures the LCD controller for colour or monochrome panels. These parameters correspond to the LCCR0[CMS] register.	colour
active or passive	Configures the LCD controller for active (TFT) or passive (STN) displays. These parameters correspond to the LCCR0[PAS] register.	active
single or dual	In passive mode configures the LCD controller for either single or dual panel displays. These parameters correspond to the LCCR0[SDS] register.	N/A
4pix or 8pix	In monochrome passive mode configures the LCD controller for either 4- or 8-pixel mode. These parameters correspond to the LCCR0[DPD] register.	N/A
hsync and vsync	Configures the polarity of the horizontal and vertical sync pulses. A value of 0 indicates an active low sync pulse, while 1 indicates active high. These parameters correspond to the LCCR3[HSP] and LCCR3[VSP] registers.	hsync:0,vsync:0

continued...

Parameter	Description	Default
outputen	Configures the polarity of the output enable signal. A value of 0 indicates an active low signal while 1 indicates active high. This parameter corresponds to the LCCR3[OEP] register.	outputen:1
pixclockpol	Configures the polarity of the pixel clock signal. A value of 0 indicates a falling edge while 1 indicates rising edge. This parameter corresponds to the LCCR3[PCP] register.	pixclockpol:0

The required values for all of the parameters passed to the **modprobe** command are generally defined in the datasheet for a given LCD panel. Often the timing parameters (**pixel clock**, **margins**, **sync pulse lengths**) can be derived from values in the datasheet along with a certain amount of trial and error. It is recommended that you read chapter 7 of the PXA255 Processor Developers manual since many of the parameters given to the driver correspond directly to settings in the LCD controller hardware.

The default parameter settings support a TFT type panel, so you can support a different TFT panel by passing only the **mode** and **timings** parameters. For example, to support a 640 x 480 panel such as the NL6448AC20 you would use the following command:

```
# modprobe pxafb options=mode:640x480-16,pixclock:39722,left:16,right:81,upper:12,lower:31,hsyncclen:63,vsyncclen:2
```



The options string must be a single line with no spaces or line-breaks.

There is much greater variation between STN type panels, so it is necessary to pass all parameters. For example, to support a Kyocera KCS057QV1AA-G00 320x240 single panel colour STN display you would use the following command:

```
# modprobe pxafb options=mode:320x240-8,pixclock:154000,left:1,right:1,upper:0,lower:0,hsyncclen:8,vsyncclen:32,passive,colour,single,outputen:1,pixclockpol:0,hsync:1,vsync:1
```

CompactFlash IEEE802.11b WiFi card

To install a CompactFlash card, insert the card on the given slot on the board.



Note that the card goes upside down.

Further information on Wi-Fi card compatibility and usage can be found in the section Wireless Network Configuration in the AEL Embedded Linux Technical Manual. The AEL Embedded Linux manual is on the Development Kit CD in the /manuals/ folder.

Software configuration

If you are using a VIPER-Lite you need to carry out some additional configuration. This primarily involves informing the Linux kernel of the reduced CPU speed.

The VIPER-Lite is a reduced power VIPER derivative based around the 200MHz PXA255 processor. You can inform the Linux kernel of the reduced maximum CPU speed by adding the following to the default kernel command line:

```
"cpu_pxa25x.max_frequency=199065"
```

For more information, see [Default command line](#) on page [20](#).

This limits the available CPU frequencies to 200MHz. For a discussion of CPU clock frequency scaling, see page [22](#).



The VIPER-Lite Flash image available in the /images/ folder on the Development Kit CD and the image loaded onto the board by default already contains the modified kernel command line.



Using software images for a VIPER on a VIPER-Lite causes a 200MHz processor to run at 400MHz, which may damage your board permanently.

Examples for VIPER / VIPER-Lite

The Development Kit includes an example program, which demonstrates how to use the VIPER I/O board. This can be found in `examples/c/` directory on the CD. It demonstrates how to access and manipulate many of the capabilities provided by the VIPER I/O board. These include functions to:

- Drive LEDs, relays and the sounder.
- Carry out operations using switches provided on the board.

Although this program is very simple, it can be used as a building block to write your own embedded applications using the VIPER I/O board.

You can also use the utilities in the `arcom-utils` package as a reference while developing your application. This package and its source can be found in `packages/arcom-utils/` directory on the CD. The utilities and the functionality they provide are:

- **exar-XR16C2852-control**, which configures an EXAR XR16C2852 UART. This chip provides COM4 and COM5 on the VIPER board.
- **devmem2**, which is a simple program that demonstrates how to read from, and write to, any location in memory.
- **rtc-alarm**, which provides a simple example of how to setup the RTC to generate an alarm at a particular time.
- **pxaregs**, a simple utility to read and modify the registers on a PXA255 XScale processor.
- **viper-sleep.sh**, which provides an example script to put a VIPER / VIPER-Lite board into sleep mode.
- **isp1161-powerdown**, which controls the power state of an isp1161 USB controller.
- **mtd-unlock** and **mtd-lock**, which lock or unlock an mtd device respectively.
- **i2c-regs**, which read from and write to the registers on devices on the I²C bus.

Further information on these utilities can be found in the README file in the package. A MAKEFILE is provided to assist in building them from the sources.

Target support software

When building applications for the VIPER / VIPER-Lite board, you must use the AEL Embedded Linux host environment. This consists of a set of utilities that provide a cross-compile environment targeting the VIPER / VIPER-Lite board. A cross-compile environment is required in order to produce binaries for the XScale processor.

Refer to the AEL Embedded Linux Technical Manual, on the accompanying CD, for an explanation of how to install the host environment. This CD also includes example source code for running the Eurotech Ltd specific device drivers and utility programs.

Appendix A – Contacting Eurotech Ltd

Eurotech Ltd sales

Eurotech Ltd's sales team is always available to assist you in choosing the board that best meets your requirements.

Eurotech Ltd
3 Clifton Court
Cambridge
CB1 7BN
UK

Tel: +44 (0)1223 403410

Fax: +44 (0)1223 410457

Email: sales@eurotech-ltd.co.uk

Comprehensive information about our products is also available at our web site:
www.eurotech-ltd.co.uk.



While Eurotech Ltd's sales team can assist you in making your decision, the final choice of boards or systems is solely and wholly the responsibility of the buyer. Eurotech Ltd's entire liability in respect of the boards or systems is as set out in Eurotech Ltd's standard terms and conditions of sale. If you intend to write your own low level software, you can start with the source code on the disk supplied. This is example code only to illustrate use on Eurotech Ltd's products. It has not been commercially tested. No warranty is made in respect of this code and Eurotech Ltd shall incur no liability whatsoever or howsoever arising from any use made of the code.

Eurotech Ltd technical support

Eurotech Ltd has a team of dedicated technical support engineers available to provide a quick response to your technical queries.

Tel: +44 (0)1223 412428

Fax: +44 (0)1223 410457

Email: support@eurotech-ltd.co.uk

Eurotech Ltd Group

Eurotech Ltd is a subsidiary of Eurotech Group. For further details see
www.eurotech.com

Index

A

AEL Embedded Linux · 16
applications, embedded · 14
audio · 9

B

board features · 4
boot scripts · 14
boot up · 16
bootstrap · 14

C

· 12
CCCR multiplier configurations · 22
installation · 7
clock
 command · 15
 frequency scaling · 22
 rates · 22
command line · 20
CompactFlash card · 28
contact details · 32
conventions · 6
copyright · 2
CPU clock · 22

D

DHCP · 18
 server · 16, 17
display controller · 25
DNS · 18
download · 14

E

embedded applications · 14
ethernet · 8, 9, 14
example
 programs · 30
 utilities · 30

F

file
 system · 19
 temporary · 19
Flash array · 9, 19
flat panel · 9

G

getting started · 7
GPIO · 9

H

handling · 4
host environment · 16, 31

I

interface · 14
IP address, static · 18

J

JFFS2 · 19
journaling · 19

K

kernel · 20
 command line · 20
 configuration · 20
 sources · 20

L

Linux · 8, 12, 16, 20
login · 17

M

modprobe · 25

N

nano · 18
networking · 18

P

packaging · 4
PC/104 · 9, 21
 interrupts · 21
performance clock rate · 23
peripherals · 9
power management · 9
power supply · 13
powersave clock rate · 23
prompts · 6

R

real time clock · 9
RedBoot · 6, 14

S

secure shell · 16
serial terminal
 configuring · 12
 emulator · 12
server
 DHCP · 16, 17
 FTP · 16
 web · 16
sleep · 24
source code · 32
SRAM · 9
static · 4
static IP address · 18
STUART · 9
support
 software · 31
 technical · 32
symbols · 6

T

TCP/IP networking · 18
technical manual · 14
technical support · 32
temporary files · 19
trademarks · 2

U

UART · 9
unpacking · 10
USB client · 9
USB host controller · 9
utilities · 30

V

vi · 18
VIPER-Lite · 29

W

wake up · 24
warranty · 2
WiFi · 28

X

X window · 16
XScale · 31
