
Copyright © 2005 by Thorcom Systems Limited.
ALL RIGHTS RESERVED

This document may not be copied in part or in whole, stored on magnetic tape or disc or in an information retrieval system, or used in any way without permission from the publisher.

The information contained in this document has been carefully checked and is believed to be accurate and reliable, however, no responsibility is assumed for inaccuracies. Thorcom Systems Limited reserves the right to make changes to this document and the associated product in line with its policy of continued improvement. Thorcom Systems Limited does not assume any responsibility for direct or consequential damage to, or loss of, property or income through the use or inability to use this product.

The equipment described in this document must be installed, operated and maintained in accordance with the written procedures described herein, and should not be used for any purpose that involves the safety of life or limb, and should not form any part of a life support system.

Details and descriptions of the product, methodology and design supplied in this documentation are considered part of the product and are covered by the manufacturer's Intellectual Property Rights. Information provided in this documentation is to allow the customer to use, install and maintain the equipment described and does not transfer any rights to such Intellectual Property Rights, Patents or Copyrights to the customer or any third party.

For further information on this product please contact Technical Support at:

Thorcom Systems Limited
Unit 4
96B Blackpole Trading Estate West
Worcester WR3 8TJ
England, U. K.

Tel: 01 905 756700 (international: +44 1 905 756700)
Fax: 01 905 755777 (international: +44 1 905 755777)

Email: support@thorcom.co.uk
Web Site: <http://www.thorcom.co.uk>

Part Number: 100-00319
Revision: 1.00
Date: 12/08/2005
Path: \\Pendragon\Documents\Manuals\Products\vr1000\100-00319 VR1000 Reference Manual Rev 1.0 Final.doc

Document Revision History

Date	Rev	Initials	Changes/Amendments
15/05/2005	0.01	MJT	Initial draft version
25/07/05	0.90	MJT	Final draft
28/07/05	0.91	MJT	Minor layout corrections. First Release
11/08/05	1.00	SPC	First release

Contents

1. Introduction	7
1.1 Intended audience	7
1.2 Related documentation.....	7
1.3 Trademarks and acknowledgements.....	7
2. Product description	8
3. Hardware Reference	10
3.1 Front and rear panel connector arrangements	10
3.1.1 Front panel	10
3.1.2 Rear panel	10
3.2 Connector pin assignments	11
3.2.1 Power and digital I/O connector	11
3.2.2 Console port	12
3.2.3 Serial ports	13
3.2.4 Ethernet ports	13
3.2.5 USB ports.....	14
3.2.6 GPS receiver antenna port	15
3.2.4 Auxiliary interface (Audio, I/O and POCSAG).....	15
3.3 VR1000 PCB layout and jumper options	16
3.3.1 Input mode selectors.....	16
3.3.2 Power supply mode	17
3.3.3 Clear SRAM/GPS.....	17
3.3.4 Hard disk drive set up and LED	17
4. Installation and commissioning	18
4.1 Maintained power supply.....	18
4.2 Power supply requirements.....	18
4.3 Installation position	19
4.4 Power supply wiring.....	19
4.4.1 Switching external devices.....	20
4.4.2 Standard power lead	20
4.5 GPS antenna and cables.....	20
5. The VR1000 boot environment (ABLE)	22
5.1 ABLE operation.....	22
5.2 Naming conventions for mass storage and memory	22
5.3 ABLE boot messages.....	23
5.3.1 Interrupting the boot sequence	24
5.3.2 Manually booting from an alternative location.....	24
5.3.3 Transfer of control to Linux.....	25
5.4 ABLE command language	25
5.4.1 Print version of ABLE ("version")	25
5.4.2 Reboot VR1000 ("reset").....	26
5.4.3 List file-systems on VR1000 ("lsfs")	26
5.4.4 List files ("ls")	26
5.4.5 Show settings in non-volatile memory ("nvshow").....	26
5.4.6 Change settings in non-volatile memory ("nvset")	27

5.4.7 Write changed settings to non-volatile memory (“nvsave”)	27
5.4.8 Clear a setting in non-volatile memory (“nvunset”)	27
5.4.9 Clear all settings in non-volatile memory (“nvclear”)	27
5.4.10 Display CPU speed (“sysspeed”)	28
5.4.11 Display Ethernet interface configuration (“ifconfig”)	28
5.4.12 Memory test (“memtest”)	28
5.5 Manually booting the VR1000 system	29
5.5.1 Boot from network	29
5.5.2 Boot from disk partition	29
5.5.3 Boot from CDROM file-system	30
5.5.4 Boot from network	30
5.6 Upgrading the ABLE firmware	30
6. The VR1000 Linux operating system environment	33
6.1 VR1000 boot-up sequence	33
6.1.1 Linux kernel boot messages	33
6.2 VR1000 file-system layout	35
6.2.1 Read-only root file-system	36
6.2.2 File-systems that must be read-write	36
6.2.3 The data partition on the Compact FLASH	37
6.2.4 The Static RAM file-system	37
6.3 Modified start-up routines	37
6.3.1 boot-misc	37
6.3.2 urandom	38
6.4 Standard applications/packages installed	38
6.5 Thorcom applications installed	38
7. System configuration settings and procedures	39
7.1 User accounts and passwords	39
7.2 Logging in as the super-user	39
7.2.1 Login from the Console port	39
7.2.2 Login via SSH over LAN or WAN connection	40
7.3 Changing the root file-system to read-write	40
7.4 Changing the root password	40
7.5 Changing operating system settings	41
7.6 Enabling VR1000 full read-write file-system mode	41
7.6.1 Change to full read-write mode	42
7.6.2 Change to read-only mode	42
7.7 Installation or removal of software packages	43
7.8 Recovery from lost system password	44
8. Installed Thorcom software	46
8.1 VR1000 power management daemon	46
8.1.1 Default powerd configuration file	46
8.1.2 Configuration parameters:	48
9. VR1000 software development environment	49
10. Technical Specifications	50
11. Approvals and Certifications	53

11.1 Home/Office use	53
11.2 Vehicle use	53
12. Software licensing and Terms & Conditions	54
12.1 GNU Public License.....	54
12.2 Thorcom Systems Software Licence (“End User Licence Agreement”)	57
12.3 Export Controls affecting equipment incorporating Data Encryption software	59

1. Introduction

This document describes Thorcom Systems VR1000 family of in-vehicle routers/vehicle locator/application hosts and documents all aspects of the product hardware, connections and interfaces, installation instructions and specifications.

This document also provides details of the software environment – including the boot loader (ABLE) and the Debian Linux™ operating system and programming environment, but does not document customer specific applications or configurations.

For details of system configurations for specific projects, customers or applications please consult the separate project-specific documentation.

1.1 Intended audience

This document is intended to be read by systems designers, systems integrators, installation staff, support staff and software application developers.

A working understanding of the Linux (Unix) operating system is assumed throughout this document.

1.2 Related documentation

This manual should be read in conjunction with the online resources available at:

<http://www.thorcom.co.uk/products/vr1000>

and:

<http://www.debian.org>

1.3 Trademarks and acknowledgements

Linux™ is a trade mark of Linus Torvalds

ARM™ is a trade mark of Acorn RISC Machines Ltd

UNIX is a registered trademark of The Open Group

I²C- bus™ and I²S- bus™ are registered trademarks of Philips Electronics NV

Cisco is a registered trademark of Cisco Systems Inc.

MSDOS™ is a registered trademark of Microsoft Corp.

2. Product description

The VR1000 is a Linux™ based in-vehicle router, GPS based vehicle locator and application server in a small foot-print providing a complete Linux environment for in-vehicle computing and communications applications including, but not limited to:

- In-vehicle IP routing (GPRS/GSM, TETRA, LAN, WiFi, etc.)
- Firewall and Virtual Private Networking
- Message switching/message routing applications (such as Thorcom's SMG)
- GPS based vehicle tracking
- Mobile data/job dispatch applications
- Data logging and recording
- Vehicle telematics and telecontrol

The VR1000 incorporate a broad range of communications interfaces including:

- 2 x 10/100Mbps fast Ethernet LAN ports
- 4 x RS232 serial communications ports
- 2 x USB 1.1 host ports
- 1 x Integral Trimble GPS receiver for high performance AVL applications
- 1 x POCSAG paging encoder for transmission of paging messages
- 1 x Stereo audio interface (line level inputs/outputs)
- 1 x RS232 console port (for diagnostics/configuration)

allowing connection to radio and communications networks including (GSM, GPRS, TETRA, Ram/Mobitex, etc.) as well as both legacy devices via RS232 and future devices via LAN and/or USB.

The VR1000 is a complete computing platform including 128Mb SDRAM main memory (standard version, 256MB on XM version), IDE or Compact FLASH based mass storage (various sizes and options) and 1MB of battery backed high speed Static RAM which can be used with 'security seal' (tamper proof alarm) for storage rapidly changing data or for storage of sensitive information such as network encryption keys.

The VR1000 includes general purpose telemetry inputs/outputs and can power on as a result of the vehicle being started or an alarm condition. Also included are power monitoring and management facilities that allow the unit to power off in adverse conditions such as when the vehicle power supply is failing.

The VR1000 employs a high performance 32-bit ARM920T derived RISC CPU manufactured by SAMSUNG operating at 266MHz while consuming typically less than 4 watts of power – this processing power brings the ability to implement high performance firewall, security and networking applications to the vehicle environment.

The VR1000 operating system is Debian GNU Linux 3.1 and runs a Linux 2.6 kernel. The VR1000 is a recognised Linux hardware platform having its own unique Linux hardware ID (0x01DB).

By employing a full implementation of the Linux operating system the VR1000 achieves a high performance, stable and reliable computing platform for in-vehicle use while allowing rapid development of in-vehicle communications and mobile data applications including:

- IP routing
- Encryption
- Security
- IPSEC VPN
- Firewall
- Name and Address Translation (NAT)
- Quality of Service (QoS)

The Debian GNU Linux 3.1 distribution for ARM CPU includes a wealth of programming languages, applications, utilities and development tools which include, but are not limited to:

- GNU C/C++ compiler (gcc) version 3.4
- PERL 5.8
- Python
- PHP
- Apache web server
- Email MTAs including Sendmail and Exim
- IP firewall using IPtables/Netfilter and customise scripting
- OpenSwan IPSEC/VPN
- Dialup networking with PPP
- IP routing and load balancing with RIP, OSPF, IGRP, BGP
- Network debugging tools: traceroute, tcpdump, ethereal, etc.

In all over 300 high quality open source projects are available to install on VR1000.

The VR1000 enjoys the same rich software development environment as larger i386 based PC systems – you can develop applications on PC based hardware running Linux and port them to the VR1000 with ease.

VR1000 hardware-specific features such as digital inputs and outputs are available via a programming API and library routines to allow you to develop your own applications.

3. Hardware Reference

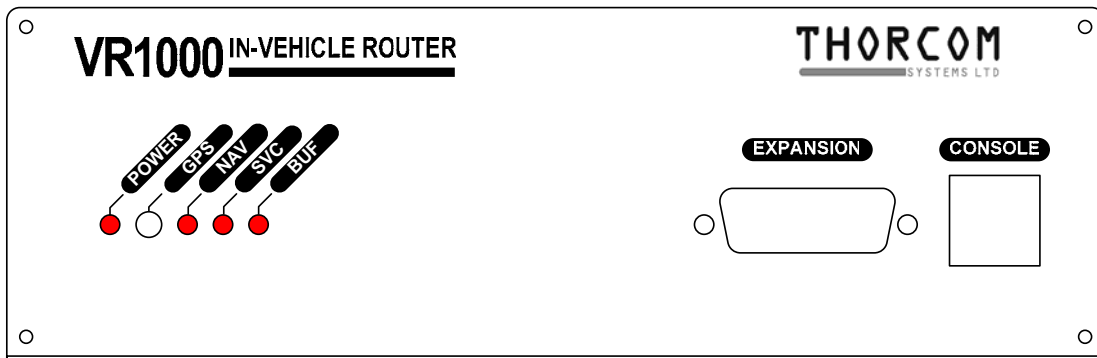
The VR1000 hardware is built on a multi-layer printed circuit board 160mm x 160mm in size which is fitted in an extruded Aluminum case with the connectors, indicators and interfaces on the two opposing faces. The main board (motherboard) hosts the Trimble GPS receiver “core module” (daughter board).

3.1 Front and rear panel connector arrangements

This section details the layout of the connectors on the front and rear panels of the VR1000.

3.1.1 Front panel

The VR1000 front panel is laid out as shown:

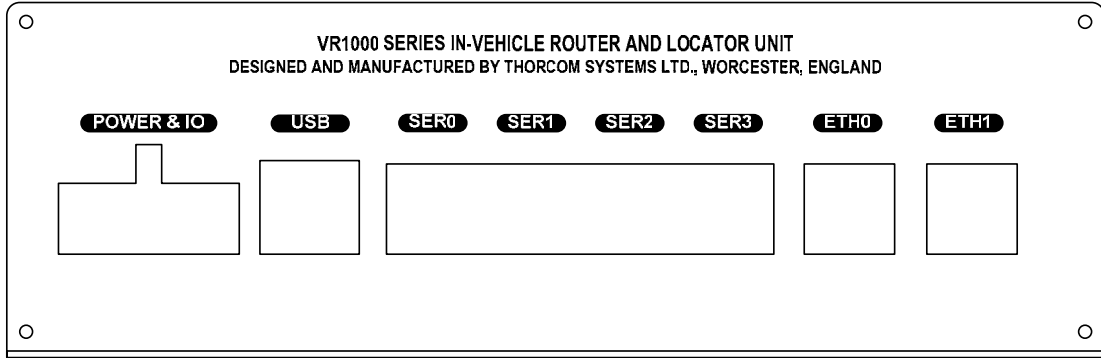


and contains the following connectors (left to right):

- Power on LED indicator
- GPS antenna connector (SMB coax connector)
- NAV (Navigation) LED indicator
- BUF (Message buffered) LED indicator
- SVC (In Service) LED indicator
- Auxiliary interface connectors (sound, digital inputs and POCSAG)
- Console port (RS232) RJ45 connector

3.1.2 Rear panel

The VR1000 rear panel is arranged as shown:



and contains the following connectors (left to right):

- a) Power and I/O connector
- b) Dual USB connector
- c) Four RS232 ports (SER0 through to SER3)
- d) First Ethernet port (Eth0)
- e) Second Ethernet Port (Eth1)

3.2 Connector pin assignments

This section describes the interface connectors, their pin assignments and uses.

3.2.1 Power and digital I/O connector

The power and digital input/output connector is a 12-way Molex Mini-Fit Junior 0.093" family (Molex part number 39-29-1128) with the following pin-out:



View looking in from outside product

Pin	Name	Description
1	GND	Ground
2	PWR	Nominal +12V DC power input: range 10.0 to 18.0 volts (unswitched)
3	IGN	Ignition Sense – input
4	ALM	Alarm Sense – input
5	IN1	General purpose input #1
6	IN2	General purpose input #2
7	RLNC	Output #1 – Relay contact (Normally closed)
8	RLCM	Output #1 – Relay contact (Common)
9	RLNO	Output #1 – Relay contact (Normally open)
10	OUT2	Output #2 – Open collector
11	IN3	General purpose input #3
12	IN4	General purpose input #4

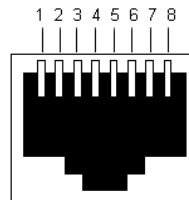
The VR1000 is designed to be connected to a maintained +12V DC power supply. Do not connect the unit directly to the vehicle engine battery – when the engine is started (“engine cranking”) the supply voltage will fluctuate wildly resulting in system re-boots and unpredictable behavior.

Please consult Section 4 (Installation & Commissioning) regarding recommended arrangements for provision of a maintained power source to the unit.

3.2.2 Console port

The Console port is the system setup and diagnostic port. The ABLE bootstrap loader and operating system messages are displayed on the console port during boot-up. After the operating system has loaded the port can be used to login and perform diagnostics and/or system configuration.

The console port has an RJ45 connector with the following pin out:



Pin	Name	Direction	Description
1	RTS	Output	Flow Control
2	DTR	Passive	Connected to DCD (pin 7)
3	TXD	Output	Transmit Data
4	GND	-	Ground
5	GND	-	Ground
6	RXD	Input	Receive Data
7	DCD	Passive	Connected to DTR (Pin 2)
8	CTS	Input	Flow Control

The Console port has the following Unix/Linux device name:

`/dev/ttySAC0`

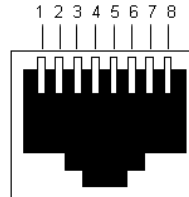
and, unless configured otherwise, operates with the following serial parameters:

Parameter	Value
Speed	115,200 bps
Start bits	1
Data bits	8
Stop bits	1
Parity	None
Flow Control	None

3.2.3 Serial ports

The rear panel provides a group of four RS232 serial communications ports named SER0 to SER3 each of which can be used for RS232 asynchronous serial communications a speeds from 300bps to 115200bps with 5, 6, 7 or 8 data bits, 1 start bit, 1 or 2 stop bits, odd/even/mar/space or no parity and hardware (CTS/RTS) or software (XON/XOFF) flow control.

The serial ports have the following pin outs:



Pin	Name	Direction	Description
1	RTS	Output	Ready To Send (flow control)
2	DTR	Output	Data Terminal Ready (general purpose)
3	TXD	Output	Transmit Data
4	GND	-	Ground
5	GND	-	Ground
6	RXD	Input	Receive Data
7	DCD	Input	Data Carrier Detect (general purpose)
8	CTS	Input	Flow Control (flow control)

The serial ports have the following Unix/Linux device names:

Port name	Unix/Linux Device name
SER0	/dev/ttyS0
SER1	/dev/ttyS1
SER2	/dev/ttyS2
SER3	/dev/ttyS3

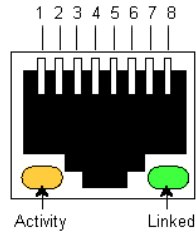
When specifying the serial ports in configuration files or applications use the Unix/Linux device names.

WARNING: When connecting external devices to the serial ports of the VR1000 please note that you should avoid connecting voltages from low source impedances to the output drivers (pins 1, 2 and 3) – such as a direct connection from 12V - as this can “over drive” the RS232 device drivers can cause permanent damage to the serial port.

3.2.4 Ethernet ports

The VR1000 contains two 10/100Mbps Ethernet LAN connections which are presented on the rear panel, called “Eth0” (Ethernet 0, first Ethernet) and “Eth1” (Ethernet 1, second Ethernet)

The Ethernet ports employ the industry standard pin-outs for 10/100Mbps Ethernet, as follows:



Pin	Name	Description
1	TXD+	Transmit Data +
2	TXD-	Transmit Data -
3	RXD+	Receive Data -
4	n/c	Not connected (reserved)
5	n/c	Not connected (reserved)
6	RXD-	Receive Data -
7	n/c	Not connected (reserved)
8	n/c	Not connected (reserved)

Each Ethernet LAN connector incorporates two LED indicators that provide status information as follows:

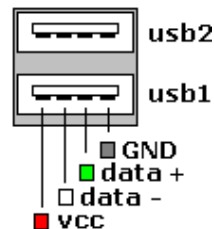
Led	State	Description
Green	Off	Not linked to an Ethernet device, switch or hub
Green	On	Linked to an Ethernet device at 10 or 100Mbps
Yellow	Off	No activity on port
Yellow	Flashing	Link activity (transmitting or receiving)

The Ethernet sockets include integral isolation transformers and can be used with standard CAT-5E unshielded twisted pair (UTP) cables up to 100m long in the home/office environment.

In the vehicle environment shielded twisted pair (STP) must be used with screened connectors to achieve the required level of EMC performance for the system and the cables must have a maximum length of 3.0m.

3.2.5 USB ports

The VR1000 provides two USB 1.1 OHCI interfaces with standard USB type "A" interface connectors:



with the following pin outs:

Pin	Name	Description
1	VCC	+5V DC supply (500mA maximum) via Polyfuse
2	D-	Data – (differential data bus)
3	D+	Data + (differential data bus)
4	GND	Ground

The VR1000 can provide 5V DC power on the USB connectors to power external devices subject to a limit of 500mA maximum consumption across both connectors. This is a limitation of the 5V power supply in the VR1000.

3.2.6 GPS receiver antenna port

The integrated Trimble ACE-III GPS receiver “core module” presents an SMB RF socket for connection of an external GPS antenna. The GPS antenna should be of the “active” type incorporating an internal 26dB gain pre-amplifier operating in the GPS L1 frequency band (1575.42MHz) and powered from the VR1000 at 4.5V DC.

Please refer to Section 4.5 for details of recommended GPS antennas and cables.

WARNING: Connection of an inappropriate GPS antenna can damage the GPS receiver and invalidate the product warranty.

3.2.4 Auxiliary interface (Audio, I/O and POCSAG)

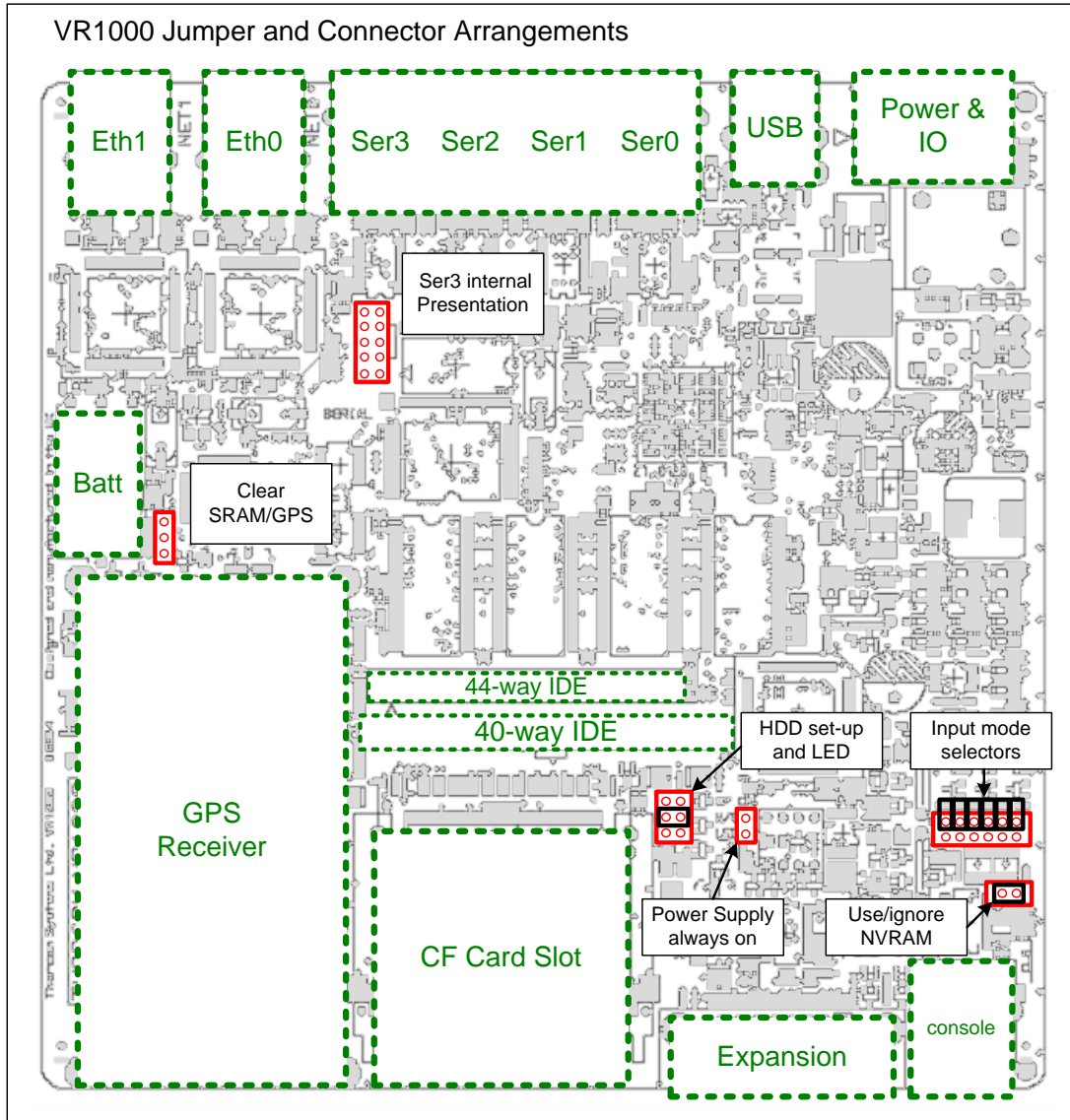
The VR1000 incorporates an Auxiliary Interface for audio, digital interfacing and POCSAG paging transmissions, allowing it to interface to radio/communications equipment, to be used as an enunciator or for multi-media playback – for example playing MP3 files.

A 15-way D-type connector is used with the following pin out:

Pin	Name	Direction	Description
1	GND	-	Ground
2	LEFT OUT	Output	1V peak-to-peak typical (nominal 600R source impedance)
3	RIGHT OUT	Output	1V peak-to-peak typical (nominal 600R source impedance)
4	GND	-	Ground
5	LEFT IN	Input	1V peak-to-peak typical (input impedance 47K)
6	RIGHT IN	Input	1V peak-to-peak typical (input impedance 47K)
7	GND	-	Ground
8	INPUT1	Input	General purpose open collector output (external switching)
9	INPUT2	Input	General purpose open collector output (external switching)
10	OUTPUT1	Output	General purpose input (external switching)
11	OUTPUT2	Output	General purpose input (external switching)
12	GND	-	Ground
13	PKEY	Output	POCSAG transmit key (transmitted on) signal. Open drain switching FET (Imax=50mA)
14	PTXD	Output	POCSAG transmit data signal. 5V TTL/CMOS compatible (buffered)
15	+12V	Output	12V supply to external equipment (via 0.5A polyfuse)

3.3 VR1000 PCB layout and jumper options

The VR1000 incorporates a number of push-fit PCB jumpers that characterise the VR1000's operation. The VR1000 circuit board layout is show below with position of the various jumpers:



3.3.1 Input mode selectors

The VR1000 digital inputs can be configured for “voltage driven” or “grounding” operation using the appropriate jumpers for Input #1 to Input #6 respectively. Input #1 is closest to the edge of the board, input #6 is furthest from the edge.

When the appropriate input jumper is not fitted, the input is configured for “voltage driven” operation – input impedance is approximately 22K ohms and the voltage applied to the pin has to be greater than 2.5V to indicate an “on” signal.

When the appropriate input jumper is fitted the input is pulled up to +5V DC by a 10K ohm resistor and the logic polarity is reversed – the input indicates “on” when the pin is open and “off” when the external device grounds the input.

WARNING: The VR1000 digital inputs incorporate EMC and spike protection diodes. Do not drive the inputs with a voltage source that is higher than the primary supply voltage or the protection diodes may become forward biased and damaged. If there is any chance that the external input voltage may be higher than the VR1000 supply voltage then an appropriate series resistor should be used – consult Thorcom for further advice.

3.3.2 Power supply mode

In normal operation VR1000 powers on when either the Ignition Sense or the Alarm Input lines are driven true (more than 2.5V DC with respect to ground) and the VR1000 powers off under a variety of conditions which are software controlled.

If the VR1000 is to remain powered on all of the time then jumper PL15 (“power always on”) can be fitted to disable the power control logic.

3.3.3 Clear SRAM/GPS

The on-board back-up battery is routed to the 1Mb system static RAM (SRAM) and to the GPS receiver to maintain in-memory data when the primary power is removed from VR1000. If the data in system SRAM or the GPS receiver is to be erased or has become corrupt, this jumper can be used to clear the contents of the appropriate memory.

This is a 3x1 jumper with ground on the middle pin (pin 2). Using a standard 2-pin jumper cap short pins:

- 1-2 to clear the contents off the system SRAM
- 2-3 to clear the contents of the GPS receiver’s internal SRAM

3.3.4 Hard disk drive set up and LED

When using a hard disk with the VR1000, the drive activity LED can be connected to the link furthest away from the IDE connectors. Pin 1 is +ve, nearest the Compact FLASH socket.

The middle jumper sets the Compact FLASH socket as slave.

The jumper nearest to the IDE connectors is not used.

4. Installation and commissioning

This section describes the installation and commissioning of the VR1000 hardware and should be read in conjunction with the documentation for the application software to be used on the product. The associated documentation will provide details on how to test and commission the installation.

4.1 Maintained power supply

The VR1000 is intended to be used with a maintained primary power supply, i.e. a power source that is “always on”. The VR1000 has its own intelligent power-up/power-down regime controlled from the “ignition sense” input in conjunction with the power management software module “powerd”.

The VR1000 should be considered as a miniaturised, but fully fledged, computer system with its own operating system. Unplanned interruptions to the primary power supply, such as the large voltage fluctuations caused by starting the vehicle engine (“engine cranking”) will result in the system rebooting and may in extreme conditions result in data corruption or loss of data.

In many applications the VR1000 will be providing message routing/switching, vehicle location operation and acting as a data logger and will be considered “mission critical” – normally important computer systems in companies or corporations are run from an Uninterruptible Power Supply (UPS) – we recommend that the VR1000 be considered equally important in terms of provision of vehicle communications.

WARNING: It is strongly recommended that the VR1000 is powered from a maintained power source in the vehicle, i.e. what is often referred to as a “technical battery” or “comms battery” and configured with appropriate power-up and power-down rules.

4.2 Power supply requirements

The VR1000 operates from a nominal 12V negative earth vehicle supply – the supply range is specified as 10.0 to 18.0 volts for normal operation. In practice VR1000 will operate over a wider supply range but this is not recommended and should not be relied upon.

Depending on the mass-storage options installed (Compact FLASH and/or 2.5” notebook IDE disk drive) the power consumption is in the range 2.5 to 4.0 watts. The USB port may power external USB devices drawing up to 0.5A at +5V resulting in a further 2.5 watts of power consumption.

A standard VR1000 as shipped from the factory with 512MB Compact FLASH card consumes 170mA at 13.5 volts resulting in a power consumption of 2.3 watts.

When the VR1000 is powered down (in standby mode) it draws under 1mA from the primary supply – used to maintain the system clock, battery backup and operate the power supply logic.

4.3 Installation position

The VR1000 can be installed in any orientation and position in the vehicle that is convenient and safe. Locations include under a passenger seat, in the boot/trunk or in vans/lorries bolted to the bulkhead between the passenger compartment and the load carrying area.

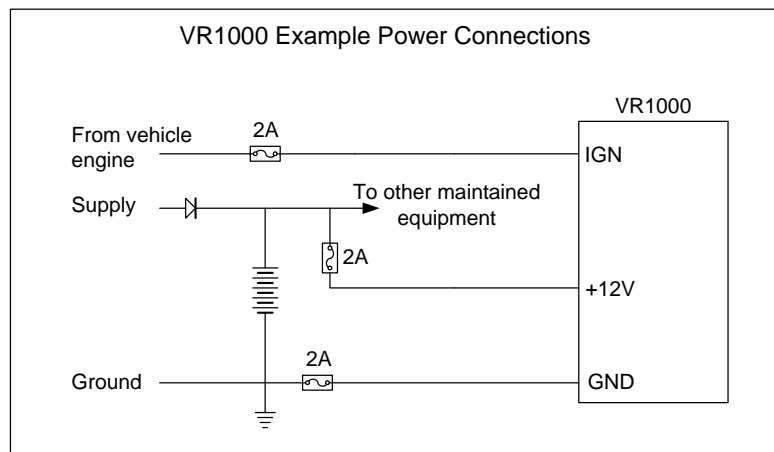
The VR1000 enclosure incorporates four oval hole fixing points in the bracket feet on 140 mm x 186 mm centers for use with M3.5 pan-head self-tap screws or similar fixings.

If the VR1000 has a 2.5" hard disk drive installed then operation in any orientation except upside-down is recommended (inverted operation is not recommended as it can cause premature wear on the bearings in hard disk drive unit).

WARNING: The VR1000 should be permanently installed using self-tap screws in all available fixing points – the units must not be mounted in such a fashion that it can break free and move around in the passenger compartment, for example under extreme braking conditions.

4.4 Power supply wiring

The VR1000 should be connected to the “technical battery” or “communications battery” as shown in the diagram below:



Note that the positive and negative leads should be connected to the battery and fused locally at the battery with a 2A automotive style fuse in both the positive and negative leads to protect the wiring against faults.

The ignition sense wire, if used, should also be protected with a 2A fuse connected at the source of the ignition sense signal.

4.4.1 Switching external devices

The VR1000 has an integral single pole change-over (SPDT) automotive relay rated at 10A for resistive loads – all three contacts (normally open, normally closed and common) are presented on the main power connector. This relay can be used to control the power to external devices such as radio/communications equipment, printers, bar code readers etc. in the vehicle.

For safe and reliable operation loads of up to 6A (resistive) should be controlled from the VR1000 to allow for the contact ratings of the connector. To switch larger loads (higher current) an external “booster” relay should be used.

The operation of the relay is software controlled by the VR1000’s power management daemon (“powerd”). See Section 8.1 for further information on the powerd software and configuration options.

4.4.2 Standard power lead

Thorcom provides a standard power lead (part number 70-00206) with VR1000 which has the following wires and colour assignments:

Pin	Name	Colour	Current Rating	Free pre-crimped cable recommendation
1	GND	Black	3 Amp	
2	PWR	Red	3 Amp	
3	IGN	Yellow	3 Amp	
4	ALM	n/c	-	
5	IN1	n/c	3 Amp	Blue pre-crimped core
6	IN2	n/c	-	
7	RLNO	n/c	3 Amp	Green pre-crimped core
8	RLCM	n/c	6 Amp	Red pre-crimped core
9	RLNC	n/c	-	
10	OUT2	n/c	-	
11	IN3	n/c	-	
12	IN4	n/c	-	

The lead is supplied partially loaded with pins 1, 2 and 3 installed in the plug and the other wires bundle but not loaded into the connector – if these are not required they can be discarded. If these are required, they can be fitted in the appropriate plug receptacle (see above table for colour code recommendations). Please refer to Thorcom if additional cable cores are required for a particular installation arrangement.

In some cases, non-Thorcom provided power wiring arrangements may be preferable. In this case, the VR1000 uses a Molex Minifit Jr 0.093 series 12-pin connector for which the mating part number is 39-29-1128.

4.5 GPS antenna and cables

The VR1000 uses an external GPS antenna for vehicle location applications. The antenna must be for operation on the standard GPS L1 frequency (1575.42MHz) and incorporate a 26dB active pre-amplifier that operates from 4.5V DC supplied on the centre pin of the antenna connector.

The antenna connector is a standard 50 ohm SMB RF connector.

Thorcom supplies the following antennas and cables for use with the VR1000:

Name	Thorcom P/N	Description
Hardmount	050-00001	Through-hole fixing antenna designed for permanent installation on the roof of vehicles. 22mm diameter hole required, TNC RF connection. Colour white.
Cable 2m	070-00103	2m long RG188 coaxial cable TNC plug to SMB plug for use with Hardmount antenna
Cable 5m	070-00122	5m long RG188 coaxial cable TNC plug to SMB plug for use with Hardmount antenna
Magmount	050-00004	Magnetic mount antenna with integral RG174 coaxial cable 5m long and SMB RF connector. Intended for temporary installation on vehicles. Colour black
Combi	See description	Through-hole fixing GPS antenna with combined PMR/TETRA/GSM whip antenna designed for permanent installation on the vehicle. 18mm diameter hole required. 5m long RG174 cable to SMB plug supplied for GPS connection and 5m long RG174 cable to BNC for PMR/TETRA/GSM connection. Refer directly to Thorcom for assistance with part numbers, cables and connectors

5. The VR1000 boot environment (ABLE)

The VR1000 incorporates a system bootstrap environment broadly equivalent in functionality to the BIOS on an Intel 386 architecture PC or the ROMMON environment on a Cisco Systems router.

The VR1000 boot environment is called the Advanced Boot Loader Environment (ABLE). ABLE is responsible for the following:

- initial power up control
- setting the CPU in to the correct operating mode such that it can boot the ARM Linux kernel and operating system
- detecting memory size, hardware options and peripherals
- storing boot parameters in non-volatile memory (I²C-bus NVRAM)
- reading the Linux operating system kernel in to memory from disk/Compact FLASH card and transferring control to the kernel

5.1 ABLE operation

The ABLE software resides in on-board (non-removable) NOR FLASH memory (39LV160) and is executed each time the VR1000 power is applied to the unit/the unit is switched on (typically then the ignition sense line is driven true).

ABLE runs immediately when the VR1000 powers up and:

- diagnoses the hardware (locates and sizes memory, locates non-volatile memory, diagnoses hard disk drives/Compact FLASH cards, diagnoses Ethernet LAN connections)
- reads the system settings and boot options non-volatile memory (I²C bus)
- loads the Linux operating system kernel in to memory and transfers control to it

In normal operation interaction with ABLE is not required as the boot process is automated, however ABLE has a command language of its own and its own operating environment.

5.2 Naming conventions for mass storage and memory

Memory devices and mass storage devices use the following naming convention:

dddn

where 'ddd' is the device name or technology type and 'n' is the unit number, for example:

rom0	for the system FLASH memory
nvrAm0	for the system non-volatile IIC bus memory

hd0 for the first hard disk or Compact FLASH drive/partition

ABLE diagnoses each IDE/ATA disk drive and/or CFA/Compact FLASH device that it finds connected to the IDE interface at system start-up and assigns each partition on each physical device a logical name in the form:

hdn

where 'n' is a number starting at zero. A Compact FLASH device in the CF Card slot with two partitions on it (standard VR1000 layout) will have them assigned as follows:

hd0 : first partition on CF card
hd1 : second partition on CF card

if a CDROM drive or hard disk drive is connected to the second IDE interface (40 pin or 44 pin connector) then these are also enumerated and assigned logical device names, for example:

hd2, hd3, hd4 ...

It should be noted that the logical names are assigned in order as disk drives and partitions are detected starting from the CF card slot on the first IDE channel and moving to the second IDE channel (40/44 pin interfaces) — as a result removing a device from the bus and rebooting the VR1000 will cause a change in the assigned logical device names.

When using device names in ABLÉ they are surrounded by round brackets, for example:

(hd0)

5.3 ABLÉ boot messages

Messages similar to the following are produced by ABLÉ on the Console Port when the VR1000 boots:

```
Initialising
Detecting SDRAM size
SDRAM: BANK6 size 04000000
SDRAM: BANK7 size 04000000
ABLE: 2.08 (s3c2410x) (vince@gerald) Fri Apr 8 16:35:26 BST 2005
Processor: Samsung S3C2410A (arm920)
System: Machine vr1000/s3c2410x, Linux id 0x01db
EEPROM: 24cXX, 1024 bytes, single byte addressed
(nvram0) on (24cxx)
sys.speed is unset, Setting CPU Speed to 266MHz
selected all-wr for console write stream
selected all-rd for console read stream
DRAM: 128 Mb (134217728 bytes)
hda: Hitachi XX.V.3.4.0.0: CFA PIO mode 4
hdc: FUJITSU MHT2030AT: ATA PIO mode 4
(hda) 488MB
(hd0) on ((hda1):ext2)
(hd1) on (hda2)
hdc:Diagnosing disc drive: ok
(hdc) 27GB
(hd2) on ((hdc1):ext2)
(hd3) on (hdc2)
(hd4) on ((hdc3):ext2)
DM9000: dm0: r1, 00:11:ac:00:04:94 int phy, link ok, 100Mbit full duplex
DM9000: dm1: r1, 00:11:ac:00:04:95 int phy, link down
TMP101: 23 degC, 9 bits accuracy (cfg 0x80, tval 0x1780)

sys.autoshadow unset, automatically shadowing
```

```
Autoboot attempt 1, Press any key to abort
Autobooting in 4...
```

If left un-interrupted ABLE will count down the auto-boot time and then execute the command stored in the non-volatile memory setting called:

```
boot.cmd
```

as if it had been types at the command prompt by the user.

The boot.cmd parameter is usually configured to load the Linux kernel into memory from the first hard disk (or Compact FLASH) partition, specifying the partition as the root file-system and specifying the VR1000 physical Console Port (ttySAC0) as the Linux Console Port.

The standard boot entry is as follows:

```
(hd0)vmlinux root=/dev/hda1 console=ttySAC0,115200
```

5.3.1 Interrupting the boot sequence

The normal VR1000/ABLE boot sequence can be interrupted at the point the system displays:

```
Auto-boot in n
```

and is counting down by pressing the [ENTER] or [RETURN] key on the terminal connected to the console port. When interrupted ABLE will display its command prompt:

```
>
```

allowing commands to be entered, settings to be changed or the system to be booted from an alternative source or with non-standard kernel parameters.

5.3.2 Manually booting from an alternative location

At the ABLE command prompt the VR1000 can be booted from an alternative location using a command in the form:

```
(device)filename [options]
```

Where 'device' is the name of the storage device, like (hd3) or for network boot (tftpboot), 'filename' is the name of the file to be loaded in to memory and executed, and '[options]' are any additional command line options to be passed to the program.

Example:

To boot from a laptop/notebook hard disk configured with three partitions, the appropriate command might be:

```
(hd3)vmlinux root=/dev/hdc3 console=ttySAC0,115200
```

assuming that Linux was on the third disk drive/partition.

Alternatively, it might be necessary to boot from the first device but force the Linux Kernel in to single user mode to perform maintenance on the file system or to rescue the system from some form of crash/damage/loss of configuration, in which case the command:

```
(hd0)vmlinux root=/dev/hda1 console=ttySAC0,115200 single
```

would be used – the additional argument ‘single’ tells the Linux Kernel to bring up Init (process 1) in single user mode and drop to a shell prompt rather than go multi-user.

5.3.3 Transfer of control to Linux

At the point a valid boot command issued, either manually, or by automatically as a result of ABLE executing the boot.cmd directive stored in non-volatile memory, ABLE loads the specified Linux kernel in to memory and executes it.

Note that the VR1000 does not require the use of a Linux “boot loader” such as LILO, GRUB or SYSLINUX because ABLE has native support for the ext2 file-system and can read ext3 file-systems in ext2 compatibility mode. ABLE is able to read the compressed kernel imaged directly from disk to memory and execute it.

When ABLE loads the Linux kernel it generates messages similar to the following:

```
loaded (hd4)vmlinux, 0x17e160 bytes at 0x00008000
boot: booting 'linux'
Booting Linux ...
```

At which point ABLE has transferred control to the binary executable image in memory. Further messages are produced by the Linux operating system kernel start-up procedures rather than ABLE. These are shown in Section 6.1.

5.4 ABLE command language

ABLE has a command language which is similar to a simplified version of the Unix/Linux command language. The full vocabulary includes many commands that are used during manufacturing setup/testing/diagnostics. A sub-set of the commonly used commands are documented here to allow the VR1000 to be configured or booted from alternative mass storage.

5.4.1 Print version of ABLE (“version”)

The “version” command prints the version/name of ABLE installed in the VR1000, usage:

```
> version
```

Example output:

```
ABLE: 2.08 (s3c2410x) (mike@pendragon) Fri Apr 8 16:35:26 BST 2005
```

5.4.2 Reboot VR1000 (“reset”)

The “reset” command terminates execution of ABLE and restarts the VR1000. Any changes made to non-volatile memory settings that have not been saved via the “nvsave” command are lost.

The VR1000 reboots and returns to ABLE, reads the non-volatile memory and obeys settings in non-volatile memory (when jumper PL1 is installed). If the jumper on PL1 is removed the contents of the NVRAM are ignored at boot time.

5.4.3 List file-systems on VR1000 (“lsfs”)

The command “lsfs” lists the file systems installed on the VR1000. ABLE enumerates each storage device and prints a list of logical devices and physical device/partition names, for example:

```
>lsfs
(hd4) (aliases to (hdc3):ext2)
(hd3) (aliases to hdc2)
(hd2) (aliases to (hdc1):ext2)
(hd1) (aliases to hda2)
(hd0) (aliases to (hda1):ext2)
(nvram0) (aliases to 24cxx)
```

This shows five logical disk drives (drive hda with two partitions and drive hdc with three partitions) and the non-volatile memory.

5.4.4 List files (“ls”)

The command “ls” lists devices and files on the VR1000 system. ABLE has a native implementation of the Linux ‘ext2’ file-system – the most common file system supporting long filenames, directories and attributes used in the Linux world and can display the files directly, for example:

```
>ls (hd4)/boot
System.map-2.6.11-bk10-audio9b-mtd20050312-thorcom5
vmlinux-2.6.11-bk10-audio9b-mtd20050312-thorcom5
```

5.4.5 Show settings in non-volatile memory (“nvshow”)

The command “nvshow” displays the ABLE settings stored in the non-volatile memory device, for example:

```
>nvshow
shell-hist (is unset)
cons-write (is unset)
cons-read (is unset)
boot.fs (is unset)
boot.auto (is unset)
boot.cmd (is unset)
boot.timeout (is unset)
ide.multi-limit (is unset)
sys.autoshadow (is unset)
sys.speed (is unset)
```

Parameters displayed as “(unset)” take sensible defaults and do not generally need to be specifically set to a value.

5.4.6 Change settings in non-volatile memory (“nvset”)

The “nvset” command is used to change one, or more, of the settings stored in non-volatile memory. The syntax of the command is:

```
nvset [item] [value]
```

where [item] is one of the tagged parameters names storable in non-volatile memory and [value] is the value to be assigned. Note that if the value is in multiple parts, i.e. includes spaces, then the whole value string must be enclosed between double quotes, for example:

```
nvset boot.cmd "(hd3)vmlinux root=/dev/hdc3 console=ttySAC0,115200"
```

Changed parameters are not stored unless the “nvsave” command (see below) is used.

5.4.7 Write changed settings to non-volatile memory (“nvsave”)

ABLE boot parameters in main memory are not written back to the non-volatile memory until the “nvsave” command is used. After changing a setting in ABLE use the “nvsave” command to ensure that the parameter has been saved and then “reset” to re-start the VR1000 and use the settings, for example:

```
> nvset boot.timeout 4
> nvsave
> reset
```

which changes the boot-up delay to 4 seconds.

5.4.8 Clear a setting in non-volatile memory (“nvunset”)

The “nvunset” command changes a parameter in non-volatile memory from the user provided values back to the default, or ‘unset’ state, for example:

```
> nvunset boot.timeout
```

will set the boot timeout back to the ‘unset’ time – defaults to 12 second when unset.

Changes need to be written back to memory using the “nvwrite” command.

5.4.9 Clear all settings in non-volatile memory (“nvclear”)

The command “nvclear” returns all of the non-volatile parameters in VR1000 back to the factory defaults stored in ABLE.

Changes need to be written back to memory using the “nvwrite” command.

5.4.10 Display CPU speed (“sys speed”)

The “sys speed” command displays the CPU clock frequency (speed) options and displays the currently selected value by placing a star (“*”) next to the selected speed, for example:

```
>sys speed
 34 MHz      45 MHz      51 MHz      48 MHz      56 MHz
 68 MHz      79 MHz      85 MHz      90 MHz      101 MHz
113 MHz      118 MHz      124 MHz      135 MHz      147 MHz
152 MHz      158 MHz      170 MHz      180 MHz      186 MHz
192 MHz      203 MHz      210 MHz      226 MHz      * 266 MHz
268 MHz      270 MHz
```

The VR1000 employs a CPU clock synthesiser that generates the clock frequencies for the various parts of the VR1000 system. The CPU operates at the selected frequency while the SDRAM (main memory) runs at half the CPU speed and the memory mapped I/O devices run at ¼ of the CPU speed.

WARNING: Thorcom recommends that the customer does not attempt to fine tune or “over clock” the VR1000 as this may result in instability.

5.4.11 Display Ethernet interface configuration (“ifconfig”)

The “ifconfig” command displays the state of the network interfaces prior to operating system boot up. This command can be used to check the state of the interface prior to using TFTP to network-boot the device.

```
>ifconfig
dm0      Link encap:Ethernet  HWaddr 00:11:ac:00:04:94
         inet addr:0.0.0.0  Mask:0.255.255.255
         gateway addr:0.0.0.0  tftpserver:0.0.0.0
         UP MTU:1500  Metric:1
```

5.4.12 Memory test (“memtest”)

The command “memtest” performs a suite of tests in the main system SDRAM memory and prints the outcome as a success/fail message. Example:

```
>memtest
Test 0 (simple alias)
Ok

Test 1 (simple address test)
.....
.....Ok
.....
.....Ok
.....
.....Ok
.....
.....Ok

Test 2 (simple address test (with pause))
.....Ok
```

5.5 Manually booting the VR1000 system

You can manually boot the VR1000 from mass storage or from the LAN/network using a command in the form:

```
(location)filename [options]
```

Where:

(location) is the location of the file to be booted, the location is the name of a logical disk or 'tftpboot' to boot from the network, for example:

```
(hd0)vmlinux
```

```
(tftpboot)able-romwrite-v208.bin
```

filename is the name of the file to be booted from the location

[options] are the optional command line arguments to be passed to the program that is being booted

5.5.1 Boot from network

ABLE supports boot from network using the BOOTP/DHCP/TFTP protocol family. The most common use of network boot is to perform an upgrade on the ABLE firmware in FLASH ROM, however network booting can be a valid method of testing new operating system kernel images.

To boot from network connect the VR1000's first Ethernet port (Eth0) to a LAN segment that has a BOOTP/DHCP and TFTP server with the file you wish to load available and use the following command:

```
>(tftpboot)filename
```

ABLE will issue BOOTP and DHCP probes on the network LAN segment and obtain an IP address, contact the TFTP server and download the file to memory and then begin execution – an example is shown in Section 5.6 (below).

5.5.2 Boot from disk partition

ABLE can boot from a file located on a hard disk or Compact FLASH card by reading the file in to memory and executing it. ABLE has native support for the 'ext2' file system and does not require an intermediate boot loader such as LILO and GRUB (used in i386 systems).

To boot from disk the command is in the form:

```
(partition)/path/filename
```

where:

(partition) is the logical partition detected by ABLE at start up, for example (hd0)

path	is an optional path to where the file is located
filename	is the name of the file to load and execute

Example:

```
> (hd0)/boot/vmlinux
```

Additional parameters may be passed on the command line, for example:

```
> (hd0)/boot/vmlinux root=/dev/hda1 console=ttySAC0,115200
```

Use the “ls” command to see a list of partitions and/or files on the partitions.

Note that the command used to boot the system is typed at the ABLE command line in the same syntax as stored in the non-volatile memory parameter boot.cmd.

5.5.3 Boot from CDROM file-system

ABLE supports boot from CDROM using ISO9660 (CDROM file systems) that are bootable (El Torito specification).

To boot from CDROM use the command:

```
> (cdrom)
```

5.5.4 Boot from network

ABLE supports boot from local area network using DHCP/TFTP/BOOTP.

The VR1000 will acquire an IP address from the DHCP server and use that address to request the file specified using BOOTP/TFTP from the server that answered the DHCP request.

To boot to a file called ‘filename’ from the network use the command:

```
> (tftpboot)filename
```

5.6 Upgrading the ABLE firmware

From time-to-time Thorcom releases an update to the ABLE firmware to support new features, new hardware or updates for bugs/issues. Thorcom provides new releases of ABLE in a form which can be BOOTP/DHCP/TFTP loaded from a suitable server.

ABLE images suitable for network boot have a name in the form:

able-romwrite-vnnn.bin

where 'vnnn' is a number like 'v208' representing ABLE version 2.08 firmware. The 'romwrite' means that the image has the FLASH Erase/Programmer/Verify code prepended to the start of the image so that the FLASH programmer is actually executed and it replaces the firmware in the onboard FLASH.

The following procedure is the recommended way to upgrade the ABLE firmware:

- configure a DHCP server to be authoritative on the network LAN segment and to hand out valid IP addresses for the LAN segment that can reach the TFTP server on the LAN segment
- configure a TFTP server to serve the ABLE firmware and place the ABLE firmware in the appropriate directory/folder and ensure that it has the correct file access permissions to be network readable. NB the TFTP server and BOOTP/DHCP server should reside on the same machine (same IP address).
- connect the VR1000 to a stable source of power – not that if the power fails during ABLE upgrade the unit can be rendered useless and would have to be returned to Thorcom for repair which is chargeable
- connect the VR1000's first Ethernet LAN port (Eth0) to the network via a suitable RJ45 cable
- connect a dumb terminal or PC running Hyperterm to the console port using a standard console cable
- start the VR1000 (by applying power, and/or asserting the ignition sense line)
- when the VR1000 displays the ABLE diagnostic messages wait until it displays:

```
Autoboot in n seconds
```

And press the [ENTER] or [RETURN] key to interrupt automatic boot-up which will yield a command prompt in the form:

```
>
```

- at the command prompt type the following:

```
(tftpboot)<filename>[enter]
```

```
(tftpboot)able-romwrite-v208.bin[enter]
```

- ABLE will issue BOOTP and DHCP probes on the network LAN segment and obtain an IP address, contact the TFTP server and download the new ABLE firmware image and FLASH loader program (romwrite), it will load the image into memory, check its integrity and execute it, displaying messages similar to the following:

```
>(tftpboot)able-romwrite-v208.bin
tftp: attempting bootp
bootp: sending request
bootp: serverip: 192.168.144.1
bootp: netmask: 255.255.255.0
bootp: serverip: 192.168.144.1
bootp: netmask: 255.255.255.0
bootp: address: 192.168.144.172
.....
.....loaded (tftpboot)able-romwrite-v208.bin, 0x58114 bytes
at 0x00008000
boot: booting 'able appl'
ROM Write: Version 1.25 (c) 2002, 2003 Thorcom Systems
Identify_machine: ableid is 6 (VR1000)
Replacing current version 207 with image version 208
Image release number is 2005041001
Image CRC passed
Machine is VR1000
Flash: SST 39VF1601 [0x00BF, 0x234B]
Initialising programmer:
Erasing device: ..... done
Writing data: ..... done
Verifying data: ..... done
Finishing operation: done
Done! - Please reset machine
>
```

- type "reset" to restart the VR1000

6. The VR1000 Linux operating system environment

This section describes the VR1000's Debian ARM Linux environment, normal system start-up and shut-down procedures, how to login, change system settings and install or remove software packages.

6.1 VR1000 boot-up sequence

The VR1000 Linux boot-up sequence is as follows:

- a) ABLE executes, initializes hardware, reads non-volatile memory and loads the Linux kernel into memory and transfers control to it
- b) the Linux kernel begins execution and starts all drivers, network devices etc. and loads process number 1 ("init") in to memory
- c) init reads the contents of /etc/inittab to work out what to do and:
 - executes all of the system start-up scripts specified in /etc/rcS.d
 - changes runlevel to that specified in /etc/inittab (runlevel 3 on the VR1000) and then executes start-up scripts specified in /etc/rc3.d

6.1.1 Linux kernel boot messages

Once ABLE has loaded the Linux kernel into memory and transferred control to it the Linux kernel decompresses itself, checks its CRC and begins execution, at which point it:

- identifies the CPU make, model and options, eg. ARM920T
- identifies the Linux machine type, in our case "Thorcom VR1000"
- detects available memory
- sets up IRQ handlers
- sets up the console
- initialises device drivers
- sets up networking
- mounts file-systems
- starts process number 1 ("init") which then boots the rest of the system

Messages similar to the following are generated by the Linux kernel on the VR1000 Console port:

```
Uncompressing Linux..... done,
booting the kernel.
Linux version 2.6.11-bk10-audio9b-mtd20050312-thorcom5 (root@pendragon.int.thorcom.com)
(gcc version 3.4.0) #1 Tue May 3 14:29:12 BST 2005
CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
Machine: Thorcom-VR1000
Memory policy: ECC disabled, Data cache writeback
CPU S3C2410A (id 0x32410002)
S3C2410: core 266.000 MHz, memory 133.000 MHz, peripheral 66.500 MHz
S3C2410 Clocks, (c) 2004 Simtec Electronics
```

```

USB Power Control, (c) 2004 Simtec Electronics
Built 1 zonelists
Kernel command line: root=/dev/hdc3 console=ttySAC0,115200
irq: clearing pending ext status 0003fe00
irq: clearing subpending status 00000092
PID hash table entries: 1024 (order: 10, 16384 bytes)
timer tcon=00500000, tcnt ea5f, tcfg 00000000,00040000, usec 00001555
Console: colour dummy device 80x30
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 128MB = 128MB total
Memory: 126208KB available (2745K code, 692K data, 96K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16
S3C2410: Initialising architecture
SCSI subsystem initialized
usbcore: registered new driver usbfs
usbcore: registered new driver hub
S3C2410 DMA Driver, (c) 2003-2004 Simtec Electronics
DMA channel 0 at c8800000, irq 33
DMA channel 1 at c8800040, irq 34
DMA channel 2 at c8800080, irq 35
DMA channel 3 at c88000c0, irq 36
Simtec TLV320AIC23 Audio platform driver
s3c2410-iis: failed to claim resource 0
NetWinder Floating Point Emulator V0.97 (double precision)
JFFS version 1.0, (C) 1999, 2000 Axis Communications AB
JFFS2 version 2.2. (C) 2001-2003 Red Hat, Inc.
Initializing Cryptographic API
S3C2410 RTC, (c) 2004 Simtec Electronics
S3C2410 Watchdog Timer, (c) 2004 Simtec Electronics
Serial: 8250/16550 driver $Revision: 1.90 $ 8 ports, IRQ sharing enabled
ttyS0 at MMIO 0x11800080 (irq = 56) is a 16550A
ttyS1 at MMIO 0x118000a0 (irq = 57) is a 16550A
ttyS2 at MMIO 0x118000c0 (irq = 58) is a 16550A
ttyS3 at MMIO 0x118000e0 (irq = 59) is a 16550A
s3c2410_serial0 at MMIO 0x50000000 (irq = 70) is a S3C2410
s3c2410_serial1 at MMIO 0x50004000 (irq = 73) is a S3C2410
s3c2410_serial2 at MMIO 0x50008000 (irq = 76) is a S3C2410
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered
loop: loaded (max 8 devices)
nbd: registered device at major 43
PPP generic driver version 2.4.2
PPP Deflate Compression module registered
PPP BSD Compression module registered
dm9000 Ethernet Driver
eth0: dm9000 at c8876000,c8878040 IRQ 54 MAC: 00:11:ac:00:04:94
eth1: dm9000 at c887a080,c887c0c0 IRQ 53 MAC: 00:11:ac:00:04:95
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
ide-floppy driver 0.99.newide
BAST: IDE driver, (c) 2003-2004 Simtec Electronics
hda: Hitachi XX.V.3.4.0.0, CFA DISK drive
ide0 at 0xee000000-0xee000007,0xee1000c0 on irq 60
hda: max request size: 128KiB
hda: 1000944 sectors (512 MB) w/1KiB Cache, CHS=993/16/63
hda: cache flushes not supported
hda: hda1 hda2
hdc: FUJITSU MHT2030AT, ATA DISK drive
ide1 at 0xee200000-0xee200007,0xee3000c0 on irq 61
hdc: max request size: 128KiB
hdc: 58605120 sectors (30005 MB) w/2048KiB Cache, CHS=58140/16/63
hdc: cache flushes supported
hdc: hdc1 hdc2 hdc3
BAST NOR-Flash Driver, (c) 2004 Simtec Electronics
Found: SST 39VF1601
bast-nor: Found 1 x16 devices at 0x0 in 16-bit bank
number of JEDEC chips: 1
cfi_cmdset_0002: Disabling erase-suspend-program due to code brokenness.
Searching for RedBoot partition table in bast-nor at offset 0x1fff000
No RedBoot partition table detected in bast-nor
Generic platform RAM MTD, (c) 2004 Simtec Electronics
vr1000_ram_setrw: dev c02d5454, to=1

```

```

mtd-ram mtd-ram.0: registered mtd device
usbmon: debugs is not available
s3c2410-ohci s3c2410-ohci: S3C24XX OHCI
s3c2410-ohci s3c2410-ohci: new USB bus registered, assigned bus number 1
s3c2410-ohci s3c2410-ohci: irq 42, io mem 0x49000000
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
drivers/usb/class/bluetty.c: USB Bluetooth support registered
usbcore: registered new driver bluetty
drivers/usb/class/bluetty.c: USB Bluetooth tty driver v0.13
usbcore: registered new driver usblp
drivers/usb/class/usblp.c: v0.13: USB Printer Device Class driver
Initializing USB Mass Storage driver...
usbcore: registered new driver usb-storage
USB Mass Storage support registered.
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.01:USB HID core driver
mice: PS/2 mouse device common for all mice
i2c /dev entries driver
s3c2410-i2c s3c2410-i2c: slave address 0x10
s3c2410-i2c s3c2410-i2c: bus frequency set to 377 KHz
s3c2410-i2c s3c2410-i2c: i2c-0: S3C I2C adapter
m41st87 0-0068: time halted, restarting
m41st87 0-0068: attached to /dev/rtc
Advanced Linux Sound Architecture Driver Version 1.0.8 (Thu Jan 13 09:39:32 2005 UTC).
i2c_adapter i2c-0: probing addr 0x1a
simtec_audio_matchdev: dev=c07cf91c
simtec_audio_matchdev: parent i2c-0
simtec_audio_matchdev: parent2 s3c2410-i2c
tlv320aic23 tlv320aic23.0: attached tlv320aic23 driver
tlv320aic23 tlv320aic23.0: clock clkout0, rate 12000000Hz
s3c24xx-tlv320aic23 s3c24xx-tlv320aic23: probe called
s3c24xx-tlv320aic23 s3c24xx-tlv320aic23: soundcard attached ok (c07cd5f0)
ALSA device list:
 #0: S3C24XX TLV320AIC23
NET: Registered protocol family 2
IP: routing cache hash table of 1024 buckets, 8Kbytes
TCP established hash table entries: 8192 (order: 4, 65536 bytes)
TCP bind hash table entries: 8192 (order: 3, 32768 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
IPv4 over IPv4 tunneling driver
GRE over IPv4 tunneling driver
ip_conntrack version 2.1 (1024 buckets, 8192 max) - 256 bytes per conntrack
ip_tables: (C) 2000-2002 Netfilter core team
ipt_recent v0.3.1: Stephen Frost <sfrost@snowman.net>.
http://snowman.net/projects/ipt_recent/
ClusterIP Version 0.6 loaded successfully
NET: Registered protocol family 1
NET: Registered protocol family 17
kjournald starting. Commit interval 5 seconds
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem) readonly.
Freeing init memory: 96K
INIT: version 2.86 booting

```

At which point control is transferred to init and the VR1000 is now running the kernel and starting all of the Debian Linux configurations and processes needed to bring the system up.

6.2 VR1000 file-system layout

This section describes the layout of the various file-systems in the standard VR1000 configuration and which data types are stored in which file-systems.

A standard VR1000 is shipped with a 512Mb Compact FLASH with two partitions set up as follows:

Partition	Primary	Size	Type	Mode	Description
1	Y	300Mb	0x82	Read-only	Linux 'ext2' file system, kernel, operating system, programs and application code
2	Y	212Mb	0x06	Read/Write	MSDOS FAT16 file system – user data

6.2.1 Read-only root file-system

The VR1000 operates from a read-only root file system – this is the first partition on the Compact FLASH card and generally known as:

```
/dev/hda1
```

It is unusual to run Unix/Linux machines with a read-only root file-system as normally systems employ a hard disk drive and the root file-system ("/") is mounted read-write. There are several reasons why having the root file-system read-write is undesirable in the harsh vehicle environment:

1. the power may go off at any time un-announced causing a 'crash' and possibly causing damage to the file-system and/or data loss.
2. the next time the system is booted after a 'crash' the operating system would have to run file-system consistency checks and this may detect errors that require manual intervention.
3. the Compact FLASH device has a limited write/re-write limit, and is subject to wear.

The last item is of particular importance – standard 'consumer' Compact FLASH units have a typical life-time of 100,000 write cycles. Unix/Linux file-systems like 'ext2' and 'ext3' maintain a significant amount of meta-data in the inodes that include things like "time last modified" and "time last accessed" – even when you read files from these file-systems the "time last accessed" (atime) is updated. In a multi-user, multi-process operating system this means that the Compact FLASH card would be wearing out while the product was in use.

6.2.2 File-systems that must be read-write

By forcing the root file-system to operate in read-only mode the drawbacks and limitations described above are avoided, however the way some of the standard directories work has to be adapted to compensate – in particular mount points:

```
    /tmp
and
    /var
```

are generally expected to be read-write as programs store temporary files, temporary data, log files and other program related state information in these areas.

The VR1000 satisfies the requirement for maintaining /tmp and /var read-write by mounting them as effectively RAMdisks using the "tmpfs" file system. The "tmpfs" file system allows the VR1000 to create a dynamic RAMdisk in main memory that works automatically.

6.2.3 The data partition on the Compact FLASH

The data partition is formatted using the FAT16 (MSDOS) file system because it is used read/write and FAT16 is more efficient in terms of number of writes/rewrites of the directory and file allocation table areas than say ext2, and we wish for the maximum lifetime from the Compact FLASH device (NB. Nearly all digital cameras use FAT16 for storing images for the same reason).

As the file system is FAT16 the usual 8.3 filename limitations apply.

The data partition is intended to be used by Thorcom supplied or customer developed applications that wish to store data.

6.2.4 The Static RAM file-system

The VR1000 incorporates a 1Mb (512Kb x 16) Static RAM (SRAM) memory device which is battery backed and write-protected except when being written to. This device appears as a raw block-mode device via the Thorcom supplied Memory Technology Driver (MTD) called:

```
/dev/mtd1
```

NB. The device `/dev/mtd0` is the on-board 39LV160 boot FLASH device.

While the SRAM can be accessed in block mode via the low-level block device `/dev/mtd1` it is intended to be formatted with a file system and accessed as a mounted device. By default the SRAM is formatted using FAT16 (MSDOS) and is mounted on `/sram`.

One use for the SRAM is for logging frequently changing data to a file and then flushing (copying) it from SRAM to the data partition on the Compact FLASH – this technique can be used, for example, with GPS data which may update once per second, and avoid having to write to the CF once-per-second.

6.3 Modified start-up routines

To support operation with a read-only root file system some of the Debian Linux start up routines have had to be modified.

6.3.1 boot-misc

At system start up the `/tmp` and `/var` mount points are mounted on the tmpfs file system in system RAM – while this achieves the requirement of having the read-write many programs expect the relevant directory structures like:

```
/var/log  
/var/lock  
/var/run  
/var/spool
```

etc. to be present and viable. The script `/etc/init.d/vr1000_bootmisc` recreates these directories each time the system starts-up.

6.3.2 urandom

Modern Linux systems have a high performance random number generator (“urandom”) which is used for a range of applications and includes things like being used to generate session keys for encrypted communications using SSL or VPN connections.

In order to maintain the quality of the random numbers generated by the system the random number generator’s entropy is saved and restored across system boot-up and shut-down. The entropy requires 4Kb (4096 bytes) to be saved from the kernel and restored on the subsequent start-up.

Thorcom has modified the `/etc/init.d/urandom` script to save the entropy to the file `/data/urandom` in the read-write area of the Compact FLASH (data partition). This file should not be deleted.

6.4 Standard applications/packages installed

The VR1000 is shipped with the base set of packages installed by the ‘debootstrap’ process for Debian GNU Linux 3.1 – the base operating system installation - plus the following packages used on the VR1000:

Package Name	Description
<code>module-init-tools</code>	Kernel module loader/unloader and tools for Linux 2.6 kernel
<code>xinetd</code>	The extended inet super server
<code>dhcp3-server</code>	The ISC DHCP server version 3
<code>joe</code>	A text editor with the look and feel of WordStar
<code>less</code>	File and stream viewing utility
<code>telnetd</code>	Telnet remote login facility
<code>ssh</code>	The Secure SHell client and server
<code>dosfstools</code>	Utilities for creating/manipulating MSDOS format data partitions
<code>pureftpd</code>	A high performance, secure, FTP server
<code>ftp</code>	FTP command line client
<code>file</code>	A utility for determining file types using magic
<code>psmisc</code>	Misc. tools for access to process information
<code>libdb4.3</code>	GNU/Berkely compatible binary database library

6.5 Thorcom applications installed

Thorcom supplies the power management daemon (“powerd”) with each VR1000 to manage correct operation in the vehicle environment. Please refer to section 8 of this manual for further information.

Other applications installed by Thorcom are on a project-by-project or customer-by-customer basis and the relevant project or customer documentation should be referred to.

7. System configuration settings and procedures

7.1 User accounts and passwords

The Linux operating system employs login/accounts in normal use for users and the system administrator – referred to as the “superuser”. On Unix/Linux systems the superuser’s login name is “root”

VR1000 is shipped with the root password set to:

```
bizzare
```

For standard VR1000 builds no user accounts are installed.

Note: Please check project-specific documentation for additional accounts that may have been added to the system as part of a system configuration or setup procedure.

7.2 Logging in as the super-user

To make changes to the operating system configuration, user accounts or application programs you have to login as the system administrator, i.e. log in to the root account.

Depending on how your VR1000 is configured you can login to the root account from the following sources:

- a) from the Console Port serial interface, using a dumb terminal
- b) from a LAN connection using the Secure Shell (SSH)
- c) over a wide area network connection (for example GPRS/GSM) via the Secure Shell (SSH)

7.2.1 Login from the Console port

Connect a router console cable between your PC/Laptop/dumb terminal and the VR1000 Console Port. Serial port settings are shown in Section 3.2.2.

You can use Thorcom cable part number 70-00212 or a standard Cisco router console port cable (Cisco part number 73-3383-01).

Press [ENTER] a couple of times and you should receive the login prompt similar to the following:

```
Thorcom Systems VR1000 In-Vehicle Mobile Data Router  
Copyright (C) 2004-2005 by Thorcom Systems Limited  
Documentation: http://www.thorcom.co.uk/products/vr1000  
  
vr1000 login:
```

You should be able to login using the username/password specified in Section 7.1 (above).

7.2.2 Login via SSH over LAN or WAN connection

The VR1000 includes a Secure Shell (SSH version 2) remote login server (“ssh”).

In order to connect to the VR1000 you need an SSH Client – most modern Unix implementations have a command line based SSH client installed. For Windows (32-bit) Thorcom recommends using “putty” – a freeware Windows SSH client which can be downloaded from the Internet, here:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Notes:

1. to connect to the VR1000 from a remote location you will need to know its IP address, or the domain name service (DNS) name of the machine
2. your SSH client must implement SSH version 2; SSH version 1 is deprecated due to security issues
3. the IP routing between you and the remote VR1000 must be established
4. any firewalls in your network must allow traffic on TCP port 22 (SSH) to be passed to the VR1000
5. the VR1000 firewall, if installed and configured, must allow traffic on TCP port 22 in from your address

7.3 Changing the root file-system to read-write

To change the root file-system from read-only to read-write mode use the following procedure:

1. login as the superuser (“root”)
2. Issue the following command:

```
mount -o remount,rw /
```

this uses the ‘mount’ program with options “remount” existing file-system and in “rw” (read-write) mode on the root file-system (“/”)

After this command is issued you have access to all the files in the directories /etc, /lib, /bin, /usr, /sbin, /boot etc. but mount points /var and /tmp are still mounted as RAM disks using /tmpfs.

7.4 Changing the root password

To change the root password use the following procedure:

1. login using the serial port or network connection as detailed in Section 7.2
2. change the root file-system to read-write mode using the procedure in 7.3

3. use the 'passwd' command to change the password. For example to change the root password to 'fred1234' you would do the following:

```
# passwd
Enter new root password: fred1234
Re-Enter new root password: fred1234
All authentication tokens updated
#
```

4. reboot the VR1000 to bring the system back in read-only mode:

```
# reboot
```

Note that this procedure requires that you have the existing root password in order to login to the VR1000 – if you have lost the root password or it has been changed by someone else and you do not know it then you will have to use the “Recovery from lost root password” procedure in Section 7.8

7.5 Changing operating system settings

In order to change operating configuration parameters, for example the host name or IP address of an interface the relevant configuration file needs to be changed in /etc.

Please refer to the Debian GNU Linux documentation at:

<http://www.debian.org/doc>

for documentation on the configuration files and their contents along with the appropriate Unix manual (“man”) pages, for example “man 5 interfaces” for details of the IP network interface setup.

Please remember that VR1000 mounts the root file system in read-only mode and that you must change from read-only to read-write mode in order to edit and change settings – please refer to the procedure in section 7.3 (above) if you want to simply edit/change a configuration setting and then reboot the unit.

7.6 Enabling VR1000 full read-write file-system mode

There may be occasions where you need to bring the VR1000 up with the Compact FLASH and root file-system in full read-write mode, rather than the read-only protected mode which VR1000 is shipped with. For example you might use this procedure to install new software to an existing VR1000 or update some operating system packages and/or files.

The procedure detailed below changes the way in which VR1000 operates from the normal read-only to full read-write root file system.

WARNING: Thorcom recommends that this procedure is used sparingly and only when necessary – running the root file system from Compact FLASH in read-write mode will accelerate the wear-out process of the FLASH and in addition may make the unit susceptible to corruption of files due to power interruptions, etc. After installing software packages or changing the configuration return the VR1000 to read-only operation as soon as practical.

7.6.1 Change to full read-write mode

To change the VR1000 to operate in full read-write file-system mode use the following procedure:

1. login as the superuser ("root")

2. change directory to /etc

```
# cd /etc
```

3. check the state of the file system table files:

```
# ls -l fstab*
```

which should show two files called 'fstab-ro' (read-only) and fstab-rw (read-write) and one symbolic link called 'fstab' which should point to 'fstab-ro', thus:

4. remove the symbolic link:

```
# rm fstab
```

5. re-create the symbolic link pointing to the read-write version of the fstab file:

```
# ln -s fstab-rw fstab
```

6. reboot the VR1000

```
# reboot
```

The VR1000 will shutdown and automatically reboot, at which point the file-systems will be re-mounted using the configuration in /etc/fstab-rw - all parts of the CF card will become writable and the partitions /tmp and /var will be on disk and not in tmpfs (RAMdisk).

WARNING: do not leave the VR1000 configured in read-write mode for extended periods as this can lead to premature failure (wear out) of the CF card. Perform whatever upgrades, modifications, installations etc. are necessary and return the system to read-only mode using the procedure in 7.7.2 below.

7.6.2 Change to read-only mode

To return the VR1000 to operating in read-only file-system mode use the following procedure:

1. login as the superuser ("root")

2. change directory to /etc

```
# cd /etc
```

3. check the state of the file system table files:

```
# ls -l fstab*
```

which should show two files called 'fstab-ro' (read-only) and fstab-rw (read-write) and one symbolic link called 'fstab' which should point to 'fstab-ro', thus:

4. remove the symbolic link:

```
# rm fstab
```

5. re-create the symbolic link pointing to the read-only version of the fstab file:

```
# ln -s fstab-ro fstab
```

6. reboot the VR1000

```
# reboot
```

The VR1000 will shutdown and automatically reboot, at which point the file systems will be re-mounted using the configuration in /etc/fstab-ro and the Compact FLASH root file-system will be mounted read-only again.

7.7 Installation or removal of software packages

Once your VR1000 is up and running you may wish to install additional software packages, upgrade versions of software or remove packages.

Debian has a low-level command-line tool called 'dpkg' which deals with the installation/removal of a single package that has a filename in the form:

```
packagename-version.arch.deb
```

where:

packagename	is the basename of the package, for example "exim4-common"
version	is a version number, for example "4.52"
arch	is the machine architecture ("arm" for the VR1000)
.deb	is the suffix for a Debian package

Debian also has a high-level toolset known collectively as the Advanced Package Tool (APT) which allows packages to be downloaded and installed across a network connection or from a local CDROM. APT is able to search packages for dependencies and resolve them such that, for example, the relevant libraries are also installed.

Documentation of the Debian package management systems is outside the scope of this document. Please refer to:

- a) the manual page for dpkg (“man dpkg”)
- b) the manual pages for apt (“man apt-search”, “man apt-get”)
- c) the Debian documentation at:

`http://www.debian.org/doc`

- d) the newbie’s guide to Debian at Sourceforge:

`http://newbiedoc.sourceforge.net/system/apt-get-intro.html`

7.8 Recovery from lost system password

The following procedure should be used to regain access to a VR1000 if the root password is lost. This procedure requires physical access to the VR1000 Console port and cannot be used over the LAN/WAN.

1. Connect a laptop, PC or dumb terminal to the VR1000 Console port, ensuring the serial parameters are as defined in Section 3.2.2
2. Connect the VR1000 to a good power source that will not be interrupted during the password recovery procedure
3. Start the VR1000 by applying an “on” signal to the Ignition Sense line (yellow wire)
4. Observe the ABLE boot messages, as described in section xx.xx
5. At the point where ABLE prints the message:

```
Auto-boot in n seconds
```

Press the [ENTER] or [RETURN] key on your terminal several times to interrupt the automatic boot sequence, you will receive the ABLE command prompt:

```
>
```

6. Manually boot the VR1000 to Linux in single user mode with the following command:

```
(hd0)vmlinux root=/dev/hda1 console=ttySAC0,115200 single
```

Note that spaces and upper/lower case matter. Copy the command exactly as shown above.

7. ABLE will load the kernel into memory. The Linux kernel will load and execute in the usual fashion and you will see the normal kernel messages. The keyword ‘single’ tells the kernel to start process #1 (“init”) in single-user mode. When init starts in single-user mode it loads a copy of the shell (command language) and just prints a prompt, thus:

```
#
```

At which point you have access to the system and can enter privileged commands.

8. Change or reset the root password using the “passwd” command as detailed in Section 7.4
9. Reboot the system using the “reboot” command

```
# reboot  
System rebooting...
```

WARNING: Do not cycle the power at this point as the changes may not be committed to disk and you may corrupt the disk file-system

8. Installed Thorcom software

The VR1000 is normally shipped with customer or project specific software pre-installed. For these items you should refer to the customer or project specific documentation.

All VR1000 units are shipped with the VR1000 power management d

8.1 VR1000 power management daemon

The VR1000 ships with a program known as the VR1000 power management daemon (“powerd”). A ‘daemon’ is in Unix/Linux what a ‘service’ is in Windows Server, i.e. a background task that performs a service.

The purpose of the VR1000 power management daemon is to:

- monitor the vehicle supply voltage
- monitor the vehicle’s ignition sense
- shut the VR1000 off when the supply is failing or, optionally, after ignition goes off
- optionally control the power supply to external devices
- broadcast status messages on the LAN network(s) to attached devices

There are three components to the power management daemon:

1. the program /sbin/powerd
2. the configuration file /etc/vr1000/powerd.conf
3. the system start script /etc/init.d/powerd

8.1.1 Default powerd configuration file

Below is a copy of the default /etc/vr1000/powerd.conf configuration file:

```
#
# /etc/vr1000/powerd.conf -- configuration file VR1000 power management
#
# This is the configuration file for the VR1000 power management system
# ("powerd"). Powerd does several things:
#
# 1. it monitors for low supply voltage conditions and if the supply
#    voltage remains low for more than a specified time starts a shutdown
#
# 2. it monitors the ignition sense line and can start a shutdown after
#    ignition has been off for a pre-determined time
#
# 3. it controls the output relay used to switch other equipment on/off
#
# 4. it provides notifications to other applications running in the VR1000
#    by sending a SIGPWR signal to them, allowing them to take appropriate
#    action
#
# 5. it provides status broadcast (unicast) messages on the LAN segment
#    to devices that are interested in the state of the VR1000
#
###
### supply voltage monitoring
###
```

```
#
# low voltage indication, below this level we have a failing supply
#
volts_low          11.2000

#
# good voltage indication, above this level to cancel the failing supply
#
volts_okay        11.4000

#
# time supply voltage low before system shutdown initiated (seconds)
#
volts_low_time    60

###
### ignition sense monitoring
###

#
# ignition off time - time after ignition goes off before we switch off (seconds)
#
ignition_off_time 14400

###
### telling other programs about power down
###

#
# we specify the names of other application's running in the VR1000 that
# require an early-warning that the system is shutting down. This is done
# by specifying the name of the application's pid file, powerd will then
# send the applications listed a SIGPWR signal before initiating the
# shutdown
#
signal_list        /var/run/avld.pid
signal_list        /var/run/smg.pid

#
# signal time is the time to wait (seconds) after telling the other programs
# via SIGPWR and actually starting the shutdown
#
signal_time       3

###
### relay output operation
###

#
# relay output operating mode:
#
#     0 = disabled (stays off)
#     1 = follows ignition sense input
#     2 = follows VR1000 power
#
relay_mode        1

###
### Network status broadcasts
###

#
# one or more destination_address entries can be specified to
# receive network status UDP/IP unicast messages. These messages
# are generated once per second and contain the supply voltage
# and status indicators.
#
# to disable this feature do not specify any 'destination_address'
# entries below.
#
#destination_port  702
#destination_address 192.168.1.102

# the end
```

8.1.2 Configuration parameters:

The powerd.conf file accepts the following configuration parameters:

Parameter	Units	Description
volts_low	Volts (3 decimal places)	Voltage below which the vehicle supply voltage is considered out-of-specification and failing and which will lead to the VR1000 shutting down
volts_okay	Volts (3 decimal places)	Voltage above which the vehicle supply voltages is considered in-specification and any pending shutdown can be cancelled. Note that there should be a gap where volts_okay is higher than volts_low to create hysteresis.
volts_low_time	Seconds	The time in seconds that the voltage has to remain below the volts_low threshold before the VR1000 starts an orderly shutdown
ignition_off_time	Seconds	Time after the Ignition Sense line goes from on->off before the VR1000 starts an orderly shutdown
signal_list	Path to PID file	Full filename/path to a PID file (filename that contains a process ID number) that should be notified of the start of a shutdown by receiving a SIGPWR message. This entry may be repeated as many times as necessary.
signal_time	Seconds	Time after SIGPWR sent to processes listed in the signal list before the actual shutdown starts
relay_mode	Number (0-2)	The operating mode for the power control relay: 0 = disabled 1 = follows the Ignition Sense line 2 = Always on when VR1000 on, off when VR1000 off
destination_port	Number (0-65535)	Specifies the UDP/IP port number on which status messages are broadcast once per second
destination_address	IP address (dotted-quad)	Specifies an IP (v4) address that is to receive status broadcast messages. This entry may be repeated as many times as necessary.

9. VR1000 software development environment

The VR1000 is normally supplied as a “runtime” system with a minimal Debian GNU Linux 3.1 (ARM) operating system distribution installed. Only the packages required to run the applications that the VR1000 was developed for are installed in order to save disk space.

In order to develop software and applications for the VR1000 language components, compilers, linkers, debuggers, library tools, development header files etc. are required.

Thorcom can supply a version of the VR1000 with integrated 30Gb notebook hard drive and all of the Debian ARM Linux tools installed, alternatively you can connect an IDE hard disk drive to either the 40-way or 44-way connectors on the VR1000 and install packages from CDROM or network – however this can be time consuming and is not recommended for Linux novice users.

Please contact the Thorcom Sales Department and ask about the VR1000-SDK.

10. Technical Specifications

VR1000 technical specification:

Item	Description
CPU	Samsung S3C2410X CPU (32-bit ARM920T RISC core) with memory management unit, USB, DMA, UARTs, I ² S, I ² C, RTC, operating at 266MHz
System memory	128MB SDRAM main memory (standard) 256MB (on Extended Memory "XM" version - ordering option) operating at 133MHz 2MB NOR FLASH (for ABLE system bootstrap) 39LV160 2KB I ² C non-volatile memory 1MB Static RAM (SRAM) battery backed non-volatile memory (for data logger function)
Mass storage	Dual-channel IDE/PATA/CF interface, with: 40-pin box header 0.1" pitch for connection of hard disk or CDROM. 44-pin header 2mm pitch for connection of 2.5" notebook type hard disk. Compact FLASH (CF) socket on board for operating system/application software storage (connected as IDE device) Maximum of four IDE/CF devices. Note that it may not be possible to support all combinations of four devices on the bus due to incompatibilities arising due to incomplete implantation of CFA/IDE devices by some manufacturers.
Serial interfaces (integrated with CPU)	Total of 7 serial interfaces, as follows: 3 serial ports provided by the CPU, as follows: a) console port, TXD/RXD at RS232/V.24 levels, presented on external RJ45 connector (for diagnostics/setup) – 5 wire interface TXD/RXD, RTX/CTS and GND at RS232/V.24 levels b) GPS receiver main port (TSIP protocol), 5V TTL/CMOS levels c) GPS receiver aux port (NMEA/RTCM), 5V TTL/CMOS levels
Serial interfaces (external)	4 serial ports using industry standard 16550A UART (quad device 16554A) each with a 7-wire interface TXD/RXD, RTS/CTS, DTR/DCD at RS232/V.24 levels on RJ45 connectors for connection to a wide range of RS232 devices.
Ethernet interfaces	2 x 10/100Mbps compatible Ethernet LAN interfaces (Davicom DM9000). Standard RJ45 connectors with integrated transformer isolation and two status LEDs. Green: Link state up/down Yellow: Activity
USB ports	Two USB 1.1 compatible OHCI host ports with 5V power source and fusing (Polyfuse). Note the VR1000 can supply a maximum of 500mA across both of the USB ports.

GPS receiver	Trimble ACE-III 8-channel parallel processing GPS receiver core module on board for vehicle location and system timing applications
Digital inputs	<p>Total of 8 low speed digital inputs, as follows:</p> <p>IGN : Ignition Sense – power up VR1000 ALM : Alarm Input – power up VR1000</p> <p>6 general purpose inputs (Input #1 to Input #6) individually jumper selectable between:</p> <p>a) external voltage driven (> 3v = on, < 3v = off) b) internally pulled up, externally grounded (open = off, grounded = on)</p> <p>Inputs #1 to Input #4 on Power & I/O connector Input #5 & Input #6 on Auxiliary/Audio connector</p>
Digital outputs	<p>Total of 4 digital outputs as follows:</p> <p>1 SPDT change-over relay on Power & I/O connector, max 6A resistive load.</p> <p>1 open-collector switching outputs (bipolar transisto) capable of switching 30V at 250mA to operate relays etc. on Power & I/O connector</p> <p>2 open-collector switching outputs (bipolar transistor or FET) capable of switching 30V at 250mA on Auxiliary/Audio interface</p> <p>Note: Open collector outputs over-current protected at 400mA</p>
Status LEDs	<p>4 front-panel LEDs (power + three status) 3mm low current CMOS:</p> <p>POWER: on when VR1000 powered NAV, SVC and BUF software controlled (by application software)</p>
Audio interface	Audio interface (Texas TLV320AIC23), line in/line out levels, 8000-441000 samples/second mono/stereo.
Paging interface	<p>POCSAG paging interface to drive external transmitter: 3-wire interface (PTXD, PKEY and GND).</p> <p>PTXD: 512bps NRZ POCSAG serial data (open collector) PKEY: Transmitter keying signal (open collector)</p>
Real time Clock	Hours/Minutes/Seconds Day/Month/Year from 32.768KHz crystal battery maintained
Power Supply	12V vehicle supply operation. Supply voltage range 10.0 to 18.0 volts.
Power Consumption	2.5 watts typical, 3.5 watts maximum (excluding hard disk drive and externally powered USB devices)
Power Management	<p>Power management sub-system to power unit up on Ignition Sense or Alarm input activation. Provide automatic shutdown after Ignition off time.</p> <p>Provide logic signals (ignition sense, alarm and volts low) to main CPU</p>
Environmental	<p>-10 to +55 C operating commercial range (required) -20 to +70 C operating industrial range (desirable)</p>
Size	<p>PCB size: 160mm x 160mm Product cased: 190mm x 190mm x 32mm (approximate)</p>

Weight	900g (2lbs) – without internal hard disk drive option
Enclosure	Aluminium extrusion. IP54 rated.
Approvals	Finished product must meet or exceed requirements of Class-B computing device under EMC directive 92/336/EEC (EN 301 489) and Vehicle EMC directive 95/54/EEC

11. Approvals and Certifications

The VR1000 has been tested for EMC compliance for use both in the Home/Office environment and for use in vehicles as follows:

11.1 Home/Office use

The VR1000 has been tested and complies with European Union Council Directive 89/336/EEC and has been tested to EN 301 489-1 for use in the home/office environment.

11.2 Vehicle use

The VR1000 has been tested under European Union Council Directive 95/54/EEC – Vehicle EMC Directive – as an Information Technology device designed for after-market installation and operation in vehicles and is approved with the following approvals number:

e11*72/245*95/54*3258*00

Vehicle installation and use is subject to the following notices and stipulations:

1. the device is not designed to control any part of the vehicle's operation including engine, propulsion, steering or braking
2. the device must not be installed in such a fashion that its operation could distract the driver of the vehicle
3. the device is connected to an auxiliary power source and NOT the main engine battery
4. all cables have a maximum length of 3.0m
5. cables connected to the Ethernet LAN ports are Shielded Twisted Pair (STP) types using high quality screened RJ45 connectors

12. Software licensing and Terms & Conditions

This section provides details of the various software license terms that apply to the VR1000 product as well as notices regarding export controls and use of the product.

12.1 GNU Public License

The VR1000 is based on Linux™ software supplied under the GNU Public License (GPL), reproduced in full, below.

Thorcom Systems has modified software obtained under the GPL in order to create the VR1000 product, copies of the software changes and configuration scripts are available on the Thorcom web site at the following URL:

<http://www.thorcom.co.uk/products/vr1000>

where you can find patches, script files and documentation necessary to recreate VR1000 operating system images.

The GNU General Public License can be found here:

<http://www.gnu.org/copyleft/gpl.html>

and is also reproduced below:

```
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA
```

```
Everyone is permitted to copy and distribute verbatim copies of this license
document, but changing it is not allowed.
```

Preamble

```
The licenses for most software are designed to take away your freedom to
share and change it. By contrast, the GNU General Public License is intended to
guarantee your freedom to share and change free software--to make sure the software
is free for all its users. This General Public License applies to most of the Free
Software Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by the GNU
Library General Public License instead.) You can apply it to your programs, too.
```

```
When we speak of free software, we are referring to freedom, not price. Our General
Public Licenses are designed to make sure that you have the freedom to distribute
copies of free software (and charge for this service if you wish), that you receive
source code or can get it if you want it, that you can change the software or use
pieces of it in new free programs; and that you know you can do these things.
```

```
To protect your rights, we need to make restrictions that forbid anyone to deny you
these rights or to ask you to surrender the rights. These restrictions translate to
certain responsibilities for you if you distribute copies of the software, or if you
modify it.
```

```
For example, if you distribute copies of such a program, whether gratis or for a
fee, you must give the recipients all the rights that you have. You must make sure
that they, too, receive or can get the source code. And you must show them these
terms so they know their rights.
```

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose

permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice. This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES

12.2 Thorcom Systems Software Licence ("End User Licence Agreement")

END-USER LICENSE AGREEMENT FOR THORCOM SOFTWARE

IMPORTANT: CAREFULLY READ THIS LICENCE BEFORE USING THIS PRODUCT. USING THIS PRODUCT INDICATES YOUR ACKNOWLEDGMENT THAT YOU HAVE READ THIS LICENCE AND AGREE TO ITS TERMS. THIS LICENCE AGREEMENT IS YOUR PROOF OF LICENCE. PLEASE TREAT IT AS VALUABLE PROPERTY.

This Thorcom End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Thorcom Systems Ltd. for a single-user

software product licence. The term "Software" as defined in this licence may include associated software components, media, printed materials, and "online" or electronic documentation. By installing, copying, or otherwise using the software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the software. Where the software is provided on a multi-user basis, each individual user will be subject to the terms of this EULA as though they were a single user.

The software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The software is licensed, not sold.

1. GRANT OF LICENSE. The software is licensed as follows:

- * Installation and Use. Thorcom grants you the right to install and use one copy of the software on your computer.
- * Backup Copies. You may also make one copy of the software as may be necessary for backup and archival purposes. You must reproduce any copyright or other notice marked on the software on the copy you make.

2. YOU MAY do the following with the software:

- * Use the software only on a single computer or network, and only by a single user at a time. If you wish to use the software for more users, you will need a further licence for each user;
- * Use a copy of the software on a home or portable computer if that copy is never loaded at the same time it is loaded on your primary computer.
- * If the software is an upgrade, whether from us or another supplier, you may use or transfer the software only in conjunction with the upgraded product, unless you destroy it. If the software is an upgrade from a Thorcom product, you may only use that upgraded product in accordance with this Licence.

3. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

- * Maintenance of Copyright Notices. You must not remove or alter any copyright notices on all copies of the software.
- * Distribution. You may not distribute copies of the software to third parties.
- * Prohibition on Reverse Engineering, Decompilation, and Disassembly. You may not reverse engineer, decompile, or disassemble the software, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
- * Rental. You may not rent, lease, or lend the software.
- * Transfer. You may permanently transfer all of your rights under this EULA, provided the recipient agrees to the terms of this EULA.
- * Support Services. Thorcom may provide you with support services related to the software ("Support Services"). Use of Support Services is governed by Thorcom. Any supplemental software code provided to you as part of the Support Services shall be considered part of the software and subject to the terms and conditions of this EULA. With respect to technical information you provide to Thorcom as part of the Support Services, Thorcom may use such information for its business purposes, including for product support and development. Thorcom will not utilise such technical information in a form that personally identifies you.
- * Compliance with Applicable Laws. You must comply with all applicable laws regarding use of the software.

4. TERM: This EULA shall continue for as long as you use the Product. However, it will terminate if you fail to comply with any of its terms or conditions. You agree, upon termination, to destroy all copies of the Product. The Limitations of Warranties and Liability set out in this EULA shall continue in force even after any termination.

5. COPYRIGHT. All title, including but not limited to copyrights, in and to the software and any copies thereof are owned by Thorcom or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the software is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by Thorcom.

6. NO WARRANTIES. Thorcom expressly disclaims any warranty for the software. THE SOFTWARE PRODUCT AND ANY RELATED DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE IMPLIED

WARRANTIES OR MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE REMAINS WITH YOU.

7.LIMITATION OF LIABILITY. To the maximum extent permitted by applicable law, in no event shall Thorcom or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the software or the provision of or failure to provide Support Services, even if Thorcom has been advised of the possibility of such damages. In any case, Thorcom's entire liability under any provision of this EULA shall be limited to the greater of the amount actually paid by you for the software or GBP£5.00; provided however, if you have entered into a Thorcom Support Contract, Thorcom's entire liability regarding Support Services shall be governed by the terms of that agreement.

8.MISCELLANEOUS. This EULA is governed by the laws of the United Kingdom.

This EULA is the entire agreement between us, supersedes any other agreement or discussions, oral or written and may not be changed except by a written signed agreement. If any provision of this EULA is declared by a Court of competent jurisdiction to be invalid, illegal, or unenforceable, such provision shall be severed from the EULA and the other provisions shall remain in full force and effect. The parties have requested that this Agreement and all documents contemplated hereby be drawn up in English.

Should you have any questions concerning this EULA, or if you desire to contact Thorcom for any reason, please contact Thorcom Systems Ltd. Unit 4, 96B Blackpole Trading Estate West, Worcester WR3 8TJ, United Kingdom

12.3 Export Controls affecting equipment incorporating Data Encryption software

The Linux operating system used in this product contains cryptographic software which is subject to the following legal notice:

This product includes publicly available encryption source code which, together with object code resulting from the compiling of publicly available source code, may be exported from the United States under License Exception "TSU" pursuant to 15 C.F.R. Section 740.13(e).

This legal notice applies to cryptographic software only.

The VR1000 product may be controlled by UK, European, US or UN laws regarding supply, transfer and use. It is your responsibility to ensure that you obey export regulations and controls and, if necessary obtain the appropriate authorisation and/or export/import licences.

One source of information is the U.S. Bureau of Industry and Security web site at:

<http://www.bis.doc.gov>

which has more information about current U.S. regulations regarding export controls and the Export Administration Regulations (EAR) program.

A second source of information is the UK Department of Trade & Industry Export Control Organisation website at:

<http://www.dti.gov.uk/export.control>

where the document entitled "UK Strategic Export Controls List" can be obtained which refers to Export of Goods, Transfer of Technology and Provision of Technical Assistance (Control) Order 2003 (EGTPTA(C)O 2003 and Council Regulation (EC) 1334/2000 as amended by regulation EC 1504/2004.