

BlueCat Linux Board Support Guide

BlueCat Linux 4.0

DOC-0531-01

For MIPS Malta Boards

Product names mentioned in *BlueCat Linux Board Support Guide for MIPS Malta Boards* are trademarks of their respective manufacturers and are used here for identification purposes only.

Copyright ©1987 - 2002, LynuxWorks, Inc. All rights reserved.
U.S. Patents 5,469,571; 5,594,903

Printed in the United States of America.

All rights reserved. No part of *BlueCat Linux Board Support Guide for MIPS Malta Boards* may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photographic, magnetic, or otherwise, without the prior written permission of LynuxWorks, Inc.

LynuxWorks, Inc. makes no representations, express or implied, with respect to this documentation or the software it describes, including (with no limitation) any implied warranties of utility or fitness for any particular purpose; all such warranties are expressly disclaimed. Neither LynuxWorks, Inc., nor its distributors, nor its dealers shall be liable for any indirect, incidental, or consequential damages under any circumstances.

(The exclusion of implied warranties may not apply in all cases under some statutes, and thus the above exclusion may not apply. This warranty provides the purchaser with specific legal rights. There may be other purchaser rights which vary from state to state within the United States of America.)

Contents

PREFACE	V
For More Information	v
Typographical Conventions	vi
Special Notes	vii
Technical Support	vii
CHAPTER 1 OVERVIEW	1
CHAPTER 2 DOWNLOADING AND BOOTING BLUECAT LINUX ON THE TARGET	3
Prerequisites	3
Hardware Setup	4
Download and Boot Procedure Overview	4
Downloading BlueCat Linux into Flash via Network	4
Downloading a Demo System into Target Flash Memory	5
Booting BlueCat Linux from Flash	7
Starting a Demo System Using go	7
Starting a Demo System in Headless Mode	7
Activating Headless Boot	7
Deactivating Headless Boot	8
Booting BlueCat Linux via Network	9
Starting a Demo System Using the go Command	9
CHAPTER 3 DEFAULT KERNEL CONFIGURATION PARAMETERS	11
CHAPTER 4 SUPPORTED DEMO SYSTEMS	31
Demo Systems	31

	developer Demo System	31
	showcase Demo System	31
	Using Selected RPM Packages	32
	Using BusyBox	32
	Creating a BlueCat Linux System for BusyBox	32
	Booting BusyBox Images from a Network	34
	Using BusyBox Utilities	36
	Using TinyLogin RPM Package	37
	Creating a BlueCat Linux System for TinyLogin	37
	Booting the TinyLogin Images from a Network	40
	Using TinyLogin Utility	40
	Using Zebra RPM Package	41
	Creating a BlueCat Linux System for Zebra	42
	Booting the Zebra Images from a Network	45
	Using Zebra Utility	46
<hr/>	CHAPTER 5	
	SUPPORTED DEVICE DRIVERS	49
<hr/>	CHAPTER 6	
	BLUECAT LINUX API.....	51
	ASCII LED Display Control	51
<hr/>	CHAPTER 7	
	DEFECT FIXES AND KNOWN LIMITATIONS.....	53
	Defect Fixes for the MIPS Malta Board	53
	MIPS Malta Known Limitations and Workarounds	53

— *Preface*

For More Information

For more information on the features of BlueCat Linux, refer to the following printed and online documentation.

- *BlueCat Linux Release Notes*

This printed document contains late-breaking information about the current release of BlueCat.

- *BlueCat Linux User's Guide*

This document contains information about installing, configuring and using BlueCat Linux.

- Online information

The complete BlueCat Linux documentation set is available on the BlueCat Documentation CD-ROM. Books are provided in both HTML and PDF formats.

Updates to these documents are available online at the LynuxWorks Website: <http://www.lynxworks.com>.

Additional information about commands and utilities is provided online with the **man** command. For example, to find information about the GNU GCC compiler, use the following syntax:

```
man gcc
```

Typographical Conventions

The typefaces used in this manual, summarized below, emphasize important concepts. All references to file names and commands are case sensitive and should be typed accurately.

Kind of Text	Examples
Body text; <i>italicized</i> for emphasis, new terms, and book titles	Refer to the <i>BlueCat Linux User's Guide</i> .
Environment variables, file names, functions, methods, options, parameter names, path names, commands, and computer data	<code>ls</code> <code>-l</code> <code>myprog.c</code> <code>/dev/null</code>
Commands that need to be highlighted within body text, or commands that must be typed as is by the user are bolded .	<code>login: myname</code> <code># cd /usr/home</code>
Text that represents a variable, such as a file name or a value that must be entered by the user	<code>cat <filename></code> <code>mv <file1> <file2></code>
Blocks of text that appear on the display screen after entering instructions or commands	Linux version 2.4.10-1 (bin@build1) (gcc version 2.95.3 20010315 (release)) #5 Tue Dec 18 13:33:08 MSK 2001 Processor: Intel StrongARM-IXP1200 revision 3 Architecture: Intel IXP1200 On node 0 totalpages: 32768 zone(0): 32768 pages. zone(1): 0 pages. zone(2): 0 pages.
Keyboard options, button names, and menu sequences	Enter , Ctrl-C

Special Notes

The following notations highlight any key points and cautionary notes that may appear in this manual.

NOTE: These callouts note important or useful points in the text.



CAUTION! Used for situations that present minor hazards that may interfere with or threaten equipment/performance.

Technical Support

LynuxWorks Technical Support is available Monday through Friday (holidays excluded) between 8:00 AM and 5:00 PM Pacific Time (U.S. Headquarters) or between 9:00 AM and 6:00 PM Central European Time (Europe).

LynuxWorks U.S. Headquarters

email support@lnxw.com

phone (408) 979-3940
800-327-5969

fax (408) 979-3945

LynuxWorks Europe

email tech_europe@lnxw.com

phone (+33) 1 30 85 06 00

fax (+33) 1 30 85 06 06

World Wide Web

website <http://www.linuxworks.com>

customer support <http://www.linuxworks.com/support/custhelp.php3>

The *BlueCat Linux Board Support Guide for MIPS Malta Boards* provides information about the BlueCat Linux Board Support Package (BSP) for MIPS Malta boards, based on the MIPS32 R4000 processor.

Throughout this Board Support Guide (BSG), the BSP is referred to as the “*malta*” and the board as the “*MIPS Malta*”.

The chapters of this BSG provide the information listed below:

- *Chapter 1* is an overview of this BSG’s individual chapters.
- *Chapter 2* describes BlueCat Linux downloading and booting procedures for the MIPS Malta target.
- *Chapter 3* details configuration of the prebuilt BlueCat Linux kernel contained in the malta BSP.
- *Chapter 4* in this guide lists the BlueCat Linux demo systems included in the malta BSP.
- *Chapter 5* lists the device drivers supported by the MIPS Malta BSP.
- *Chapter 6* describes the BlueCat Linux Application Interface specific to the MIPS Malta board.

Downloading and Booting BlueCat Linux on the Target

This chapter provides instructions for downloading a BlueCat Linux demo system from a cross development host onto the target, and then booting the demo system on the target board.

Prerequisites

This document is a guide to downloading and booting BlueCat Linux systems onto the user's target board. Scenarios that use demo systems included in the BlueCat Linux distribution are presented. A basic familiarity with the target board hardware and operation is required to use this guide. The user must also have an understanding of system administration for the particular cross development host on which BlueCat Linux and the Board Support Package (BSP) are installed. It is assumed that the user has the manufacturer's documentation for the target board as well as system administration reference material for the cross development host.

Before downloading and booting BlueCat Linux on the target board, it is assumed that the default BlueCat Linux configuration and the malta BSP are installed on the development host. The user must:

1. Install the BlueCat Linux MIPS Core onto the cross development host, as described in the "Installing the Default Configuration" section in Chapter 1, "Introduction and Installation" of the *BlueCat Linux User's Guide*.
2. Install the malta BSP onto the cross development host as detailed in the "Installing Target Board Support" section of Chapter 1, "Introduction and Installation" in the *BlueCat Linux User's Guide*.
3. Activate support for the malta BSP as detailed in the "Activating Support for a Target Board" section of Chapter 1, "Introduction and Installation" in the *BlueCat Linux User's Guide*.

Hardware Setup

The target board is connected via a null modem serial cable to a terminal emulator on the host COM port. On the target, the serial port is located on the left connector (`tṭy0`) of J6. Ethernet is connected from the target J5 connector to a hub. The hub, in turn, is connected to the cross development host.

NOTE: The baud rate and parity settings for terminal emulator should be set to 38,400 bps, 8 bis/char, no parity, 1 stopbit, and RTS/CTS hardware flow control.

Download and Boot Procedure Overview

The procedure for downloading and booting BlueCat Linux onto the MIPS Malta target consists of the following main steps:

- Downloading a BlueCat Linux embedded system into the MIPS Malta flash memory using the YAMON firmware
- Booting the BlueCat Linux system programmed into Flash in one of the following ways:
 - Manually, with the `go` command
 - Automatically, in a “headless” mode

Downloading BlueCat Linux into Flash via Network

This section provides an example of downloading the `showcase` demo system supported by the malta BSP into the target’s flash memory using the YAMON firmware.

These instructions are applicable to any demo system provided in the BlueCat Linux distribution. This chapter uses the `showcase` demo system as an example.

Downloading a Demo System into Target Flash Memory

Use the following procedure to download the `showcase` demo system into target flash memory:

1. Copy the `showcase.srec` file from the `$BLUECAT_PREFIX/demo/showcase` directory to the TFTP host.

The `showcase.srec` file is the `showcase.kdi` file converted to the S-records format. For the `showcase` demo system, this conversion is performed automatically when building the image from the BlueCat Linux environment. To convert a custom demo system to the S-records format for downloading into target flash memory, use the following command:

```
BlueCat:$ mips-lynx-linux-bluecat-objcopy -I \
binary -O srec --change-address=0x9e100000 \
<KDI_image> <S-records_image>
```

To convert a custom demo system to the S-records format for the booting via network, use the following command:

```
BlueCat:$ mips-lynx-linux-bluecat -objcopy -I \
binary -O srec --change-address=0xa1000000 \
<KDI_image> <S-records_image>
```

where `<KDI_image>` and `<S-records_image>` correspond to the custom demo system files in the KDI and S-records formats, respectively.

2. To configure the host as a TFTP server do either of the following depending on the operating system running on the host:

A) For Red Hat Linux version 7.1 or 7.2:

- Install the `tftp-server` RPM package from the Red Hat distribution, if this package has not been installed already:

```
# rpm -i tftp-server-<version>-<release>.i386.rpm
```

where `<version>` and `<release>` are the version number and release of the `tftp-server` RPM package.

- Restart the `xinetd` daemon by typing:

```
# /etc/rc.d/init.d/xinetd restart
```

B)For SuSE version 7.2 or 7.3:

- Install the `tftp` RPM package from the SuSE distribution, if this package has not been installed already:

```
# rpm -i tftp-<version>-<release>.i386.rpm
```

where `<version>` and `<release>` are the version and release numbers of the `tftp` RPM package.

- Uncomment the following line in the `/etc/inetd.conf` configuration file:

```
tftp dgram udp wait root /usr/sbin/tcpd in.tftpd /tftpboot
```

- Send the `SIGHUP` signal to force the `inetd` process to re-read the configuration file:

```
BlueCat:# kill -HUP <inetd_pid>
```

where `<inetd_pid>` is the process identification number of the `inetd` process.

3. Copy the `showcase.flash.srec` or the `showcase.srec` file from the `$BLUECAT_PREFIX/demo/showcase` directory to the TFTP host.
4. Power up the target board by connecting the target's power supply to an outlet. Hit the **RESET** switch on the board. At this point, the `YAMON>` prompt appears on the terminal emulator connected to the host.
5. Type the following commands at the `YAMON>` prompt to install BlueCat Linux into target flash memory. (Type `help` at the `YAMON>` prompt for a list of available commands. Typing `help` followed by a specific command name brings up a synopsis of the command.)

```
YAMON> setenv ipaddr <target_ip>
```

```
YAMON> setenv gateway <gateway_ip>
```

```
YAMON> setenv subnetmask <netmask>
```

```
YAMON> erase 0x1e100000 0x002e0000
```

```
The following area will be erased:
```

```
Start address = 0x1e100000
```

```
Size          = 0x002e0000
```

```
Confirm ? (y/n) y
```

```
Erasing...Done
```

```
YAMON> load tftp://<server_ip>/showcase.flash.srec
```

```
About to load tftp://172.17.0.1/showcase.flash.srec
```

```
Press Ctrl-C to break
```

```
.....
```


Upon successful completion of this command, the BlueCat Linux system programmed into target flash memory automatically start son every board power-up/reset.

Deactivating Headless Boot

Use the following procedure to deactivate headless boot of BlueCat Linux:

1. Reset the board.
2. Wait until the following message appears on the serial console:

```
Environment variable 'start' exists.  
After 2 seconds it will be interpreted as  
a YAMON command and executed.  
Press Ctrl-C to bypass this.
```

3. Press the **Ctrl-C** keys to get the `YAMON>` prompt.
4. Type the following command at the `YAMON>` prompt:

```
YAMON> unsetenv start
```

Default Kernel Configuration Parameters

The malta BSP comes with a default BlueCat Linux kernel. This kernel has a number of configuration parameters with options set to default values. These parameters are detailed in the tables listed below:

Table 3-1: Kernel Configuration Options

Parameters	Table Number
Code Maturity Level Options	Table 3-2
Machine Selection	Table 3-3
Loadable Module Support	Table 3-4
CPU Selection	Table 3-5
General Setup	Table 3-6
Plug and Play Configuration	Table 3-7
Memory Technology Devices (MTD)	Table 3-8
User Modules and Translation Layers	Table 3-9
RAM/ROM/Flash Chip Drivers	Table 3-10
Mapping Drivers for Chip Access	Table 3-11
Self-Contained MTD Device Drivers	Table 3-12
Disk-On-Chip Device Drivers	Table 3-13
NAND Flash Device Drivers	Table 3-14
Parallel Port Support	Table 3-15
Block Devices	Table 3-16
Multi-Device Support (RAID and LVM)	Table 3-17
Networking Options	Table 3-18

Table 3-1: Kernel Configuration Options (Continued)

Parameters	Table Number
QoS and/or Fair Queuing	Table 3-19
Telephony Support	Table 3-20
ATA/IDE/MFM/RLL Support	Table 3-21
SCSI Support	Table 3-22
Network Device Support	Table 3-23
ARCnet Devices	Table 3-24
Ethernet (10 or 100 Mbit)	Table 3-25
Ethernet (1000 Mbit)	Table 3-26
Wireless LAN (Non-Ham Radio)	Table 3-27
Token Ring Devices	Table 3-28
WAN Interfaces	Table 3-29
Amateur Radio Support	Table 3-30
IrDA (Infrared) Support	Table 3-31
ISDN Subsystem	Table 3-32
Old CD-ROM Drivers (not SCSI, not IDE)	Table 3-33
Character Devices	Table 3-34
Serial Drivers	Table 3-35
I2C Support	Table 3-36
L3 Serial Bus Support	Table 3-37
Mice	Table 3-38
Joysticks	Table 3-39
Wtachdog Cards	Table 3-40
Ftape, the Floppy Tape Device	Table 3-41
Multimedia Devices	Table 3-42
File Systems	Table 3-43
Network File Systems	Table 3-44
Partition Types	Table 3-45

Table 3-1: Kernel Configuration Options (Continued)

Parameters	Table Number
Sound	Table 3-46
USB Support	Table 3-47
Input Core Support	Table 3-48
Kernel Hacking	Table 3-49

Table 3-2: Code Maturity Level Options

Option	Value	Description
CONFIG_EXPERIMENTAL	Y	Prompt for development and/or incomplete code/drivers

Table 3-3: Machine Selection

Option	Value	Description
CONFIG_ACER_PICA_61	N	Support for Acer PICA 1 chipset
CONFIG_ALGOR_P4032	N	Support for Algorithmics P4032 (Experimental)
CONFIG_BAGET_MIPS	N	Support for BAGET MIPS series (Experimental)
CONFIG_DECSTATION	N	Support for DECstations (Experimental)
CONFIG_DDB5074	N	Support for NEC DDB Vrc-5074 (Experimental)
CONFIG_MIPS_EV96100	N	Support for Galileo EV96100 Evaluation board
CONFIG_MIPS_EV64120	N	Support for Galileo EV64120 Evaluation board
CONFIG_MIPS_ATLAS	N	Support for MIPS Atlas board
CONFIG_MIPS_MALTA	Y	Support for MIPS Malta board
CONFIG_NINO	N	Support for Philips Nino (Experimental)
CONFIG_MIPS_MAGNUM_4000	N	Support for Mips Magnum 4000
CONFIG_MOMENCO_OCELOT	N	Support for Momentum Ocelot board
CONFIG_DDB5476	N	Support for NEC DDB Vrc 5476
CONFIG_DDB5477	N	Support for NEC DDB Vrc 5477

Table 3-3: Machine Selection (Continued)

Option	Value	Description
CONFIG_OLIVETTI_M700	N	Support for Olivetti M700-10
CONFIG_SGI_IP22	N	Support for SGI IP22
CONFIG_SNI_RM200_PCI	N	Support for SNI RM200 PCI
CONFIG_MIPS_ITE8172	N	Support for ITE 8172G
CONFIG_MIPS_IVR	N	Support for Globespan IVR board
CONFIG_MIPS_PB1000	N	Support for Alchemy Semi PB1000 board

Table 3-4: Loadable Module Support

Option	Value	Description
CONFIG_MODULES	Y	Enable loadable module support
CONFIG_MODVERSIONS	Y	Set version information on all module symbols
CONFIG_KMOD	Y	Kernel module loader support

Table 3-5: CPU Selection

Option	Value	Description
CONFIG_CPU_MIPS32	Y	CPU type
CONFIG_CPU_ADVANCED	N	Override CPU options

Table 3-6: General Setup

Option	Value	Description
CONFIG_CPU_LITTLE_ENDIAN	Y	Generate little endian code
CONFIG_BLUECAT_LOADER	N	BlueCat Linux OS Loader
CONFIG_BLUECAT_IGNORE_PRINTK	N	BlueCat Linux ignore printk
CONFIG_BLUECAT_SMALL_FOOTPRINT	N	BlueCat Linux small memory footprint

Table 3-6: General Setup (Continued)

Option	Value	Description
CONFIG_BLUECAT_MEMSIZE	N	Memory sizing benchmark
CONFIG_BLUECAT_MALTA_ASCII_LED	N	BlueCat Linux Malta ASCII LED control
CONFIG_BINFMT_MISC	N	Kernel support for MISC binaries
CONFIG_NET	Y	Networking support
CONFIG_PCI_NAMES	N	PCI device name database
CONFIG_HOTPLUG	N	Support for hot-pluggable devices
CONFIG_SYSVIPC	Y	System V IPC
CONFIG_BSD_PROCESS_ACCT	N	BSD Process Accounting
CONFIG_SYSCTL	N	Sysctl support

Table 3-7: Plug and Play Configuration

Option	Value	Description
CONFIG_PNP	N	Plug and play support

Table 3-8: Memory Technology Devices (MTD)

Option	Value	Description
CONFIG_MTD	Y	Memory Technology Device (MTD) support
CONFIG_MTD_DEBUG	N	Debugging
CONFIG_MTD_PARTITIONS	Y	MTD partitioning support
CONFIG_MTD_REDBOOT_PARTS	N	RedBoot partition table parsing

Table 3-9: User Modules And Translation Layers

Option	Value	Description
CONFIG_MTD_CHAR	Y	Direct character device access to MTD devices
CONFIG_MTD_BLOCK	N	Caching block character device access to MTD devices
CONFIG_MTD_BLOCK_RO	N	Read-only block device access to MTD devices
CONFIG_FTL	N	FTL (Flash Translation Layer) support
CONFIG_NFTL	N	NFTL (NAND Flash Translation Layer) support

Table 3-10: RAM/ROM/Flash Chip Drivers

Option	Value	Description
CONFIG_MTD_CFI	Y	Detect flash chips by Common Flash Interface (CFI) probe
CONFIG_MTD_JEDEC	N	Detect non-CFI AMD/JEDEC-compatible flash chips
CONFIG_MTD_CFI_ADV_OPTIONS	N	Flash chip driver advanced configuration options
CONFIG_MTD_CFI_INTELEXT	Y	Support for Intel/Sharp flash chips
CONFIG_MTD_CFI_AMDSTD	N	Support for AMD/Fujitsu flash chips
CONFIG_MTD_RAM	N	Support for RAM chips in bus mapping
CONFIG_MTD_ROM	N	Support for ROM chips in bus mapping
CONFIG_MTD_ABSENT	N	Support for absent chips in bus mapping
CONFIG_MTD_OBSOLETE_CHIPS	N	Older drivers for non-CFI chips

Table 3-11: Mapping Drivers for Chip Access

Option	Value	Description
CONFIG_MTD_PHYSMAP	N	CFI Flash device in physical memory map
CONFIG_MTD_MALTA	Y	Flash chip mapping on MIPS Malta board

Table 3-11: Mapping Drivers for Chip Access (Continued)

Option	Value	Description
CONFIG_MTD_MALTA_PART	:	Partitions layout
CONFIG_MTD_PCI	N	PCI MTD driver

Table 3-12: Self-Contained MTD Device Drivers

Option	Value	Description
CONFIG_MTD_PMC551	N	Ramix PMC551 PCI Mezzanine RAM card support
CONFIG_MTD_SLRAM	N	Uncached system RAM
CONFIG_MTD_MTDDRAM	N	Test driver using RAM
CONFIG_MTD_BLKMTD	N	MTD emulation using block device

Table 3-13: Disk-On-Chip Device Drivers

Option	Value	Description
CONFIG_MTD_DOC1000	N	M-Systems Disk-On-Chip 1000 support
CONFIG_MTD_DOC2000	N	M-Systems Disk-On-Chip 2000 and Millenium support
CONFIG_MTD_DOC2001	N	M-Systems Disk-On-Chip Millennium-only alternative driver support

Table 3-14: NAND Flash Device Drivers

Option	Value	Description
CONFIG_MTD_NAND	N	NAND Device Support

Table 3-15: Parallel Port Support

Option	Value	Description
CONFIG_PARPORT	N	Parallel port support

Table 3-16: Block Devices

Option	Value	Description
CONFIG_BLK_DEV_FD	N	Normal PC floppy disk support
CONFIG_BLK_CPQ_DA	N	Compaq SMART2 support
CONFIG_BLK_CPQ_CISS_DA	N	Compaq Smart Array 5xxxx support
CONFIG_BLK_DEV_DAC960	N	Mylex DAC960/DAC1100 PCI RAID controller support
CONFIG_BLK_DEV_LOOP	N	Loopback device support
CONFIG_BLK_DEV_NBD	N	Network block device support
CONFIG_BLK_DEV_RAM	Y	RAM disk support
CONFIG_BLK_DEV_RAM_SIZE	4096	Default RAM disk size
CONFIG_BLK_DEV_INITRD	N	Initial RAM disk (<code>initrd</code>) support
CONFIG_BLUECAT_RFS	Y	BlueCat Linux RFS support

Table 3-17: Multi-Device Support (RAID and LVM)

Option	Value	Description
CONFIG_MD	N	Multiple devices driver support (RAID and LVM)

Table 3-18: Networking Options

Option	Value	Description
CONFIG_PACKET	N	Packet socket
CONFIG_NETLINK	N	Kernel/User netlink socket
CONFIG_NETFILTER	N	Network packet filtering
CONFIG_FILTER	N	Socket Filtering
CONFIG_UNIX	Y	UNIX domain sockets
CONFIG_INET	Y	TCP/IP networking
CONFIG_IP_MULTICAST	N	IP: Multicasting
CONFIG_IP_ADVANCED_ROUTER	N	IP: Advanced router
CONFIG_IP_PNP	Y	IP: Kernel-level autoconfiguration
CONFIG_NET_IPIP	N	IP: Tunneling
CONFIG_NET_IPGRE	N	IP: GRE tunnels over IP
CONFIG_INET_ECN	N	IP: TCP Explicit Congestion Notification support
CONFIG_SYN_COOKIES	N	IP: TCP syncookie support (not enabled per default)
CONFIG_IPV6	N	The IPv6 protocol (Experimental)
CONFIG_KHTTPD	N	Kernel httpd acceleration (Experimental)
CONFIG_ATM	N	Asynchronous Transfer Mode (ATM) (Experimental)
CONFIG_IPX	N	The IPX protocol
CONFIG_ATALK	N	Appletalk protocol support
CONFIG_DECNET	N	DECnet support
CONFIG_BRIDGE	N	802.1d Ethernet Bridging
CONFIG_X25	N	CCITT X.25 Packet Layer (Experimental)
CONFIG_LAPB	N	LAPB Data Link Driver (Experimental)
CONFIG_LLC	N	802.2 LLC (Experimental)
CONFIG_NET_DIVERT	N	Frame Diverter (Experimental)
CONFIG_ECONET	N	Acorn Econet/AUN protocols (Experimental)
CONFIG_WAN_ROUTER	N	WAN router

Table 3-18: Networking Options (Continued)

Option	Value	Description
CONFIG_NET_FASTROUTE	N	Fast switching
CONFIG_NET_HW_FLOWCONTROL	N	Forwarding between high speed interfaces

Table 3-19: QoS and/or Fair Queueing

Option	Value	Description
CONFIG_NET_SCHED	N	QoS and/or fair queueing

Table 3-20: Telephony Support

Option	Value	Description
CONFIG_PHONE	N	Linux telephony support

Table 3-21: ATA/IDE/MFM/RLL Support

Option	Value	Description
CONFIG_IDE	Y	ATA/IDE/MFM/RLL support
CONFIG_BLK_DEV_IDE	Y	Enhanced IDE/MFM/RLL disk/CD-ROM/tape/floppy support
CONFIG_BLK_DEV_HD_IDE	N	Use old disk-only driver on primary interface
CONFIG_BLK_DEV_IDEDISK	N	Include IDE/ATA-2 DISK support
CONFIG_BLK_DEV_IDECD	N	Include IDE/ATAPI CD-ROM support
CONFIG_BLK_DEV_IDETAPE	N	Include IDE/ATAPI TAPE support
CONFIG_BLK_DEV_IDEFLOPPY	N	Include IDE/ATAPI FLOPPY support
CONFIG_BLK_DEV_CMD640	N	CMD640 chipset bugfix/support
CONFIG_BLK_DEV_RZ1000	N	RZ1000 chipset bugfix/support
CONFIG_BLK_DEV_IDEPCI	Y	Generic PCI IDE chipset support

Table 3-21: ATA/IDE/MFM/RLL Support (Continued)

Option	Value	Description
CONFIG_IDEPCI_SHARE_IRQ	N	Sharing PCI IDE interrupts support
CONFIG_BLK_DEV_IDEDMA_PCI	N	Generic PCI bus-master DMA support
CONFIG_BLK_DEV_OFFBOARD	N	Boot off-board chipsets first support
CONFIG_BLK_DEV_OPTI621	N	OPTi 82C621 chipset enhanced support (Experimental)
CONFIG_IDE_CHIPSET	N	Other IDE chipset support
CONFIG_BLK_DEV_ATA RAID	N	Support for IDE RAID controllers

Table 3-22: SCSI support

Option	Value	Description
CONFIG_SCSI	N	SCSI support

Table 3-23: Network Device Support

Option	Value	Description
CONFIG_NETDEVICES	Y	Network device support
CONFIG_DUMMY	N	Dummy net driver support
CONFIG_BONDING	N	Bonding driver support
CONFIG_EQUALIZER	N	EQL (serial line load balancing) support
CONFIG_TUN	N	Universal TUN/TAP device driver support
CONFIG_FDDI	N	FDDI driver support
CONFIG_HIPPI	N	HIPPI driver support (Experimental)
CONFIG_PPP	N	PPP support
CONFIG_SLIP	N	SLIP support
CONFIG_NET_FC	N	Fibre Channel driver support
CONFIG_RCPCI	N	Red Creek Hardware VPN (Experimental)
CONFIG_SHAPER	N	Traffic Shaper (Experimental)

Table 3-24: ARCnet Devices

Option	Value	Description
CONFIG_ARCNET	N	ARCnet support

Table 3-25: Ethernet (10 or 100 Mbit)

Option	Value	Description
CONFIG_NET_ETHERNET	Y	Ethernet (10 or 100 Mbit)
CONFIG_HAPPYMEAL	N	Sun Happy Meal 10/100baseT support
CONFIG_SUNGEM	N	Sun GEM support
CONFIG_NET_VENDOR_3COM	N	3COM cards
CONFIG_NET_VENDOR_SMC	N	Western Digital/SMC cards
CONFIG_NET_VENDOR_RACAL	N	Racal-Interlan (micom) NI cards
CONFIG_HP100	N	HP 10/100VG PCLAN (ISA, EISA, PCI) support
CONFIG_NET_PCI	Y	EISA, VLB, PCI and on board controllers
CONFIG_PCNET32	Y	AMD PCnet32 support
CONFIG_ADAPTEC_STARFIRE	N	Adaptec Starfire support (Experimental)
CONFIG_TULIP	N	DECchip Tulip (dc21x4x) PCI support
CONFIG_DE4X5	N	Generic DECchip & DIGITAL EtherWORKS PCI/EISA
CONFIG_DGRS	N	Digi Intl. RightSwitch SE-Xsupport
CONFIG_DM9102	N	DM9102 PCI Fast Ethernet Adapter support (Experimental)
CONFIG_EEPRO100	N	EtherExpress PRO/100 support
CONFIG_FEALNX	N	Myson MTD-8xx PCI Ethernet support
CONFIG_NATSEMI	N	National Semiconductor DP83810 series PCI Ethernet support
CONFIG_NE2K_PCI	N	PCI NE2000 and clones support
CONFIG_8139TOO	N	RealTek RTL-8139 PCI Fast Ethernet Adapter support
CONFIG_SIS900	N	Sis 900 PCI Fast Ethernet Adapter support

Table 3-25: Ethernet (10 or 100 Mbit) (Continued)

Option	Value	Description
CONFIG_EPIC100	N	SMC EtherPower II (Experimental)
CONFIG_SUNDANCE	N	Sundance Alta support
CONFIG_TLAN	N	TI ThunderLAN support
CONFIG_VIA_RHINE	N	VIA Rhine support
CONFIG_WINBOND_840	N	Winbond W89c840 PCI Ethernet support
CONFIG_LAN_SAA9730	N	Philips SAA9730 Ethernet support
CONFIG_NET_POCKET	N	Pocket and portable adapters

Table 3-26: Ethernet (1000 Mbit)

Option	Value	Description
CONFIG_ACENIC	N	Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support
CONFIG_DL2K	N	D-Link DL2000-based Gigabit Ethernet support
CONFIG_NS83820	N	National Semiconductor DP83820 support
CONFIG_HAMACHI	N	Packet Engines Hamachi GNIC-II support
CONFIG_YELLOWFIN	N	Packet Engines Yellowfin Gigabit-NIC support
CONFIG_SK98LIN	N	SysKconnect SK-98xx support

Table 3-27: Wireless LAN (Non-Ham Radio)

Option	Value	Description
CONFIG_NET_RADIO	N	Wireless LAN (non-ham radio)

Table 3-28: Token Ring Devices

Option	Value	Description
CONFIG_TR	N	Token Ring driver support

Table 3-29: WAN Interfaces

Option	Value	Description
CONFIG_WAN	N	WAN Interfaces support

Table 3-30: Amateur Radio Support

Option	Value	Description
CONFIG_HAMRADIO	N	Amateur radio support

Table 3-31: IrDA (Infrared) Support

Option	Value	Description
CONFIG_IRDA	N	IrDA subsystem support

Table 3-32: ISDN Subsystem

Option	Value	Description
CONFIG_ISDN	N	ISDN support

Table 3-33: Old CD-ROM Drivers (not SCSI, not IDE)

Option	Value	Description
CONFIG_CD_NO_IDESCSI	N	Support non-SCSI/IDE/ATAPI CD-ROM drivers

Table 3-34: Character Devices

Option	Value	Description
CONFIG_VT	N	Virtual terminal
CONFIG_SERIAL	Y	Standard/generic (8250/16550 and compatible UART) serial support
CONFIG_SERIAL_CONSOLE	Y	Support for console on serial port
CONFIG_SERIAL_EXTENDED	N	Extended dumb serial driver options
CONFIG_DRM	N	Direct Rendering Manager (XFree86 DRI support)
CONFIG_MWAVE	N	ACP Modem (Mwave) support

Table 3-35: Serial Drivers

Option	Value	Description
CONFIG_SERIAL_8250	N	8250/16550 and compatible serial support (Experimental)

Table 3-36: I2C Support

Option	Value	Description
CONFIG_I2C	N	I2C support

Table 3-37: L3 Serial Bus Support

Option	Value	Description
CONFIG_L3	N	L3 support

Table 3-38: Mice

Option	Value	Description
CONFIG_BUSMOUSE	N	Bus Mouse support
CONFIG_MOUSE	N	Mouse support (not serial)

Table 3-39: Joysticks

Option	Value	Description
CONFIG_INPUT_GAMEPORT	N	Game port support

Table 3-40: Watchdog Cards

Option	Value	Description
CONFIG_WATCHDOG	N	Watchdog Timer support

Table 3-41: Ftape, the Floppy Tape Device Driver

Option	Value	Description
CONFIG_FTAPPE	N	Ftape (QIC-80/Travan) support

Table 3-42: Multimedia Devices

Option	Value	Description
CONFIG_VIDEO_DEV	N	Video for Linux

Table 3-43: File Systems

Option	Value	Description
CONFIG_QUOTA	N	Quota support
CONFIG_AUTofs_FS	Y	Kernel automounter support
CONFIG_AUTofs4_FS	N	Kernel automounter version 4 support
CONFIG_ReiserFS_FS	N	Reiserfs support
CONFIG_ADFS_FS	N	ADFS file system support (read-only) (Experimental)
CONFIG_AFFS_FS	N	Amiga FFS file system support
CONFIG_HFS_FS	N	Apple Macintosh file system support (Experimental)
CONFIG_BFS_FS	N	BFS file system support (Experimental)
CONFIG_CMS_FS	N	CMS file system support
CONFIG_EXT3_FS	N	Ext3 file system support
CONFIG_FAT_FS	N	FAT file system support
CONFIG_EFS_FS	N	EFS file system support
CONFIG_JFFS_FS	N	Journalling Flash File System (JFFS) support
CONFIG_JFFS2_FS	N	Journalling Flash File System (JFFS2) support
CONFIG_CRAMFS	N	Compressed ROM file system support
CONFIG_TMPFS	N	Virtual memory file system support
CONFIG_RAMFS	N	Simple RAM-based file system support
CONFIG_ISO9660_FS	N	ISO 9660 CD-ROM file system support
CONFIG_MINIX_FS	N	Minix file system support
CONFIG_FREEVxFS_FS	N	FreeVxFs file system support
CONFIG_NTFS_FS	N	Windows NT NTFS support (read-only)
CONFIG_HPFS_FS	N	OS/2 HPFS file system support (read-only)
CONFIG_PROC_FS	Y	/proc file system support
CONFIG_DEVFS_FS	N	/dev file system support (Experimental)
CONFIG_QNX4FS_FS	N	QNX file system support (Experimental)
CONFIG_ROMFS_FS	N	ROM file system support

Table 3-43: File Systems (Continued)

Option	Value	Description
CONFIG_EXT2_FS	Y	Second extended file system support
CONFIG_SYSV_FS	N	System V, Version 7 and Coherent file system support
CONFIG_UDF_FS	N	UDF File System support (read-only)
CONFIG_UFS_FS	N	UFS file system support (read-only)

Table 3-44: Network File Systems

Option	Value	Description
CONFIG_CODA_FS	N	Coda file system support (advanced network file system)
CONFIG_INTERMEZZO_FS	N	InterMezzo file system support (Experimental, replicating file system)
CONFIG_NFS_FS	Y	NFS file system support
CONFIG_NFS_V3	N	Provide NFSv3 client support
CONFIG_NFSD	N	NFS server support
CONFIG_SMB_FS	N	SMB file system support (to mount WfW shares, etc.)
CONFIG_NCP_FS	N	NCP file system support (to mount NetWare volumes)

Table 3-45: Partition Types

Option	Value	Description
CONFIG_PARTITION_ADVANCED	N	Advanced partition selection

Table 3-46: Sound

Option	Value	Description
CONFIG_SOUND	N	Sound card support

Table 3-47: USB Support

Option	Value	Description
CONFIG_USB	N	Support for USB

Table 3-48: Input Core Support

Option	Value	Description
CONFIG_INPUT	N	Input core support

Table 3-49: Kernel Hacking

Option	Value	Description
CONFIG_CROSSCOMPILE	N	Are you using a cross compiler
CONFIG_REMOTE_DEBUG	N	Remote GDB kernel debugging
CONFIG_LL_DEBUG	N	Low-level debugging
CONFIG_MAGIC_SYSRQ	N	Magic SysRq key
CONFIG_MIPS_UNCACHED	N	Run uncached
CONFIG_BLUECAT_KDBG	N	Include <code>kdbg</code> kernel debugger

This chapter provides information about BlueCat Linux demo systems supported by the malta Board Support Package (BSP).

Demo Systems

The table below lists the demo systems supported in the malta BSP, the boot devices supported by each demo system, and their respective RAM and ROM requirements.

Table 4-1: Demo Systems Supported by malta BSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
developer	Network using firmware	3797 KB	22016 KB
showcase	Flash Network using firmware	2613 KB	16384 KB

developer Demo System

The `developer` demo system is a package consisting of the functionalities of the `shell`, `ftp`, `ping`, `gdb`, and `vl_demo` systems. Refer to Chapter 4 of the *BlueCat Linux User's Guide* for descriptions of `developer` and its components.

showcase Demo System

The `showcase` demo system starts and configures the Apache HTTP daemon, turning the target board into a Web server. Refer to Chapter 4 of the *BlueCat Linux User's Guide* for details.

Using Selected RPM Packages

This section describes how to use selected RPM packages that are frequently deployed in the embedded systems environment.

Using BusyBox

The BusyBox RPM package combines tiny versions of many common UNIX utilities into a single small executable. It provides minimalist replacements for most utilities found usually in `fileutils`, `shellutils`, `findutils`, `textutils`, `grep`, `gzip`, `tar`, etc. BusyBox provides a fairly complete POSIX environment for any small or embedded system.

The utilities in BusyBox generally have fewer options than their full-featured GNU counterparts; however, the options included provide the expected functionality and behave much like their GNU counterparts.

Creating a BlueCat Linux System for BusyBox

This section describes the steps for creating and booting a BlueCat Linux system containing BusyBox, and demonstrates the use of BusyBox utilities.

1. Create a new directory by typing

```
BlueCat:$ mkdir -p \  
$BLUECAT_PREFIX/demo/busybox/local
```

2. Set up the BlueCat Linux kernel configuration using the standard kernel configuration tools, and copy the kernel configuration file to the `$BLUECAT_PREFIX/demo/busybox` directory.

```
BlueCat:$ cd $BLUECAT_PREFIX/usr/src/linux  
  
BlueCat:$ make xconfig  
  
BlueCat:$ cp .config \  
$BLUECAT_PREFIX/demo/busybox/busybox.config
```

NOTE: The kernel configuration file for the `developer demo` (`$BLUECAT_PREFIX/demo/developer/developer.config`) is also recommended as a starting point.

3. Create a BlueCat Linux Kernel Downloadable Image, `busybox.kernel`.

```
BlueCat:$ cd $BLUECAT_PREFIX/demo/busybox
BlueCat:$ mkkernel ./busybox.config \
./busybox.kernel ./busybox.disk
```

4. Create a specification file (`busybox.spec`) with the following minimal directives:

```
strip on

mkdir /dev
mknod /dev/console c 5 1

mkdir /lib
mkdir -p /usr/lib
mkdir /bin
mkdir /sbin
mkdir -p /etc/rc.d
mkdir /proc

cp ./local/fstab ./local/inittab /etc
cp ./local/rc.sysinit /etc/rc.d

lcd ${BLUECAT_PREFIX}/sbin
cp reboot busybox /sbin

ln -s /sbin/busybox /sbin/init
ln -s /sbin/busybox /sbin/ifconfig
ln -s /sbin/busybox /sbin/route
ln -s /sbin/busybox /bin/mount
ln -s /sbin/busybox /bin/sh
ln -s /sbin/busybox /bin/ping

chmod 711 /etc/rc.d/rc.sys.init
chmod 755 /bin /sbin
# End of File
```

5. Create the `local/fstab` file with the following contents:

```
proc /proc proc defaults 0 0
```

6. Create the `local/inittab` file with the following contents:

```
# System initialization.
::sysinit:/etc/rc.d/rc.sysinit

::respawn:/bin/sh
```

NOTE: The first two fields in every record of the inittab file are ignored by the BusyBox init, so they must be empty. For example, the line `1:12345:respawn:/bin/sh` is not valid.

7. Create the `local/rc.sysinit` file with the following contents:

```
#!/bin/sh

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
mount -a
```

8. Create a root file system image (`busybox.rfs`) by entering the following command:

```
BlueCat:$ mkrootfs -lv ./busybox.spec ./busybox.rfs
```

NOTE: Makefile for the developer demo system can be used as a starting point to produce the BusyBox kernel and root file system images.

Booting BusyBox Images from a Network

Use the following procedure to boot the BlueCat Linux with the BusyBox utility from a network using the YAMON firmware. Refer to Chapter 2, “Downloading and Booting BlueCat Linux on the Target” for a detailed description of the BlueCat Linux booting procedure.

1. At the YAMON prompt, type the following commands:

```
YAMON> load tftp://<host_IP>/busybox.srec
About to load tftp://172.17.0.1/busybox.srec
Press Ctrl-C to break
.....
.....
.....
.....
.....
Start = 0xa1000000, range = (0xa1000000,0xa116bbff), format = SREC
YAMON> go
```

where `<host_IP>` is the IP address of the host.

2. The following sample output appears:

```
Compressed kernel : A0220014-A028C764
RFS                : A1F04000-A1FFEC00
Heap               : A02A8BD4-A02B8BD4
Command line       : ramdisk_size=28472 root=101
Uncompressing Linux... done.
Now booting the kernel

Linux started...
Loading MIPS32 MMU routines.
CPU revision is: 00018001
Primary instruction cache 16kb, linesize 16 bytes (4 ways)
Primary data cache 16kb, linesize 16 bytes (4 ways)
Number of TLB entries 16.
```

```

Linux version 2.4.10-1 (root@alger) (gcc version 2.95.3 20010315
(release)) #1 Tue Mar 5 15:28:22 2002
Determined physical RAM map:
 memory: 00001000 @ 00000000 (reserved)
 memory: 000ef000 @ 00001000 (ROM data)
 memory: 00010000 @ 000f0000 (reserved)
 memory: 00147000 @ 00100000 (reserved)
 memory: 01db9000 @ 00247000 (usable)
Initial ramdisk at: 0x01f04000 (1027072 bytes)
On node 0 totalpages: 8192
zone(0): 4096 pages.
zone(1): 4096 pages.
zone(2): 0 pages.
Kernel command line: ramdisk_size=28472 root=101
console=ttyS0,38400
calculating r4koff... 00061a87(400007)
CPU frequency 80.00 MHz
Calibrating delay loop... 79.87 BogoMIPS
Memory: 28880k/30436k available (995k kernel code, 1556k reserved,
69k data, 64k init)
Dentry-cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode-cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer-cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
Checking for 'wait' instruction... unavailable.
POSIX conformance testing by UNIFIX
PCI: Probing PCI hardware on host bus 0.
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Starting kswapd v1.8
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with MANY_PORTS SHARE_IRQ
SERIAL_PCI enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
block: queued sectors max/low 19101kB/6367kB, 64 slots per queue
RAMDISK driver initialized: 16 RAM disks of 28472K size 1024
blocksize
pnet32_probe_pci: found device 0x001022.0x002000
 iaddr=0x001200 resource_flags=0x000101
eth0: PCnet/FAST III 79C973 at 0x1200, 00 d0 a0 00 00 8a
pnet32: pnet32_private lp=a10a0000 lp_dma_addr=0x10a0000 assigned
IRQ 10.
pnet32.c:v1.25kf 26.9.1999 tsbogend@alpha.franken.de
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 31760
Freeing BlueCat RFS memory: 1003k freed
VFS: Mounted root (ext2 filesystem).
Malta ASCII LED display driver: Ok
Freeing prom memory: 956kb freed
Freeing unused kernel memory: 64k freed
Algorithmics/MIPS FPU Emulator v1.4
serial console detected. Disabling virtual terminals.
init started: BusyBox v0.52pre (2002.02.28-13:19+0300) multi-call
binary

```

Using BusyBox Utilities

This section provides the examples of using the BusyBox utilities. Entering a command from the following list results in the respective output:

- `ls`

```
/ # ls /
bin      etc      lost+found  sbin
dev      lib      proc        usr
```

- `cat`

```
/ # cat /etc/inittab
# System initialization
::sysinit:/etc/rc.d/rc.sysinit

::respawn:/bin/sh
```

- `chmod`

```
/ # chmod a-x /sbin/reboot
/ # ls -la /sbin/reboot
-rw-r--r--  1 0      0              7812 Dec 18  2001 /sbin/reboot
/ # chmod 755 /sbin/reboot
/ # ls -la /sbin/reboot
-rwxr-xr-x  1 0      0              7916 Jan 30 12:29 /sbin/reboot
```

- `echo`

```
/ # echo !!!!
!!!!
```

- `date`

```
/ # date
Wed Jan 30 16:06:54 UTC 2002
```

- `uname`

```
/ # uname -a
Linux (none) 2.4.10-1 #4 sRD qNW 30 15:29:52 MSK 2002 i586 unknown
```

- `mount`

```
/ # mount
/dev/root on / type ext2 (rw)
proc on /proc type proc (rw)
```

- `ifconfig`

```
/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:C0:DF:AD:EA:FF
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

```

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:10 Base address:0x320

/ # ifconfig eth0 172.17.1.3
/ # ping -c 5 172.17.0.1

PING 172.17.0.1 (172.17.0.1): 56 data bytes
64 bytes from 172.17.0.1: icmp_seq=0 ttl=255 time=1.0 ms
64 bytes from 172.17.0.1: icmp_seq=1 ttl=255 time=0.5 ms
64 bytes from 172.17.0.1: icmp_seq=2 ttl=255 time=0.5 ms
64 bytes from 172.17.0.1: icmp_seq=3 ttl=255 time=0.5 ms
64 bytes from 172.17.0.1: icmp_seq=4 ttl=255 time=0.5 ms

--- 172.17.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.6/1.0 ms

```

Using TinyLogin RPM Package

The TinyLogin RPM package is a suite of tiny UNIX utilities for handling logging into, being authenticated by, changing one's password for, and otherwise maintaining users and groups on an embedded system. It also provides shadow password support to enhance system security.

This section describes the steps necessary for creating and booting a BlueCat Linux system containing TinyLogin and demonstrates the use of the utility.

Creating a BlueCat Linux System for TinyLogin

Use the following procedure to create a BlueCat Linux image for TinyLogin:

1. Create a new directory:

```

BlueCat:$ mkdir -p \
$BLUECAT_PREFIX/demo/tinylogin/local

```

2. Set up the BlueCat Linux kernel configuration by using the standard kernel configuration tools, and copy the kernel configuration file to the \$BLUECAT_PREFIX/demo/tinylogin directory.

```

BlueCat:$ cd $BLUECAT_PREFIX/usr/src/linux
BlueCat:$ make xconfig
BlueCat:$ cp .config \
$BLUECAT_PREFIX/demo/tinylogin/tinylogin.config

```

NOTE: The kernel config file for the `developer` demo (`$(BLUECAT_PREFIX)/demo/developer/developer.config`) is also recommended as a starting point.

3. Create the BlueCat kernel downloadable image (`tinylogin.kernel`):

```
BlueCat:$ cd $BLUECAT_PREFIX/demo/tinylogin
BlueCat:$ mkkernel ./tinylogin.config \
./tinylogin.kernel ./tinylogin.disk
```

4. Create a specification file (`tinylogin.spec`) that contains the following minimal directives:

```
strip on

mkdir /dev
mknod /dev/console c 5 1
ln -s /dev/console /dev/tty
ln -s /dev/console /dev/tty1

mkdir /bin
mkdir /sbin
mkdir -p /etc/rc.d
mkdir /proc
mkdir /tmp
mkdir -p /usr/bin

mkdir /root

mkdir /dev/pts
mknod /dev/ptmx c 5 2

chmod 0666 /dev/ptmx

cp ./local/fstab ./local/passwd ./local/inittab /etc
cp ./local/securetty ./local/shadow /etc
cp ./local/rc.sysinit /etc/rc.d
cp ${BLUECAT_PREFIX}/etc/shells /etc
chmod 644 /etc/shells
cp ${BLUECAT_PREFIX}/etc/group /etc

lcd ${BLUECAT_PREFIX}/sbin
cp reboot init mingetty /sbin

cp ${BLUECAT_PREFIX}/usr/bin/tinylogin /usr/bin
ln -s /usr/bin/tinylogin /usr/bin/passwd
ln -s /usr/bin/tinylogin /bin/login

lcd ${BLUECAT_PREFIX}/bin
cp mount bash ls cat hostname /bin
ln -s /bin/bash /bin/sh

chmod 711 /etc/rc.d/rc.sysinit
```

```
chmod 755 /bin /sbin /usr/bin
chmod 04755 /usr/bin/tinylogin
# End of File
```

NOTE: In this `.spec` file, the `/bin/login` and `/usr/bin/passwd` symbolic links point to `/usr/bin/tinylogin`. This allows the user to change his/her password by simply typing `passwd`.

5. Create the `local/fstab` file with the following contents:

```
none /proc    proc
none /dev/pts devpts
```

6. Create the `local/inittab` file with the following contents:

```
id:1:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l:12345:respawn:/sbin/mingetty tty1
```

7. Create the `local/securetty` file with the following contents:

```
console
tty1
```

8. Create the `local/passwd` file with the following contents:

```
root:x:0:0:/root:/bin/bash
guest:x:500:10:0:/bin/bash
```

9. Create the `local/shadow` file:

```
root::10942:0:99999:7:::
guest::500:10:99999:7:::
```

10. Create the `local/rc.sysinit` file with the following contents:

```
#!/bin/sh

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

mount -a
hostname myhostname
```

11. Create a root file system image (`tinylogin.rfs`) by entering the following command:

```
BlueCat:$ mkrootfs -lv ./tinylogin.spec \
./tinylogin.rfs
```

NOTE: Makefile for the developer demo system can be used as a starting point to produce the TinyLogin kernel and root file system images.

Booting the TinyLogin Images from a Network

Use the following procedure to boot BlueCat Linux with the TinyLogin utility from a network using the YAMON firmware. Refer to Chapter 2, “Downloading and Booting BlueCat Linux on the Target” for a detailed description of the BlueCat Linux booting procedure.

1. At the YAMON prompt, type the following commands:

```
YAMON> load tftp://<host_IP>/ tinylogin.srec
About to load tftp://172.17.0.1/tinylogin.srec
Press Ctrl-C to break
.....
.....
.....
.....
.....
.....
Start = 0xa1000000, range = (0xa1000000,0xa116d1ff), format = SREC
YAMON> go
```

where `<host_IP>` is an IP address of the host.

The TinyLogin utility is loaded onto the target board and started automatically.

Using TinyLogin Utility

This section provides the examples of using the TinyLogin utility:

- Changing the guest password:

```
myhostname login: guest
bash-2.04$ passwd
Changing password for guest
Enter the new password (minimum of 5, maximum of 8 characters)

Please use a combination of upper and lower
case letters and numbers.
Enter new password: <new_guest_password>
Re-enter new password: <new_guest_password>
passwd[14]: password for 'guest' changed by user 'guest'
Password changed.
bash-2.04$ exit
```

- Changing the root password:

```
myhostname login: root
login[16]: root login on 'console'

bash-2.04# passwd
Changing password for root
Enter the new password (minimum of 5, maximum of 8 characters)

Please use a combination of upper and lower
case letters and numbers.
Enter new password: <new_root_password>
Re-enter new password: <new_root_password>
passwd[17]: password for 'root' changed by user 'root'
Password changed.
bash-2.04# exit
```

- Getting the root permissions:

```
myhostname login: guest
Password: <guest_password>
bash-2.04$ tinylogin su
Password: <guest_password>
login[20]: root login on 'console'

bash-2.04#
```

Using Zebra RPM Package

GNU Zebra is a free software that manages a TCP/IP-based routing protocol. It takes multi-server and multi-thread approach to resolve the complexity of the Internet.

GNU Zebra supports BGP4, BGP4+, OSPFv2, OSPFv3, RIPv1, RIPv2, and RIPng.

GNU Zebra is intended to be used as a Route Server and a Route Reflector. It is not a toolkit; it provides full routing power under a new architecture. GNU Zebra is unique in design in that it has a process for each protocol.

This section describes the steps necessary for creating and booting a BlueCat Linux system containing Zebra and demonstrates use of the Zebra utility.

Creating a BlueCat Linux System for Zebra

The user must step through the following procedure to create a BlueCat Linux image for Zebra:

1. Create a new directory by typing:

```
BlueCat:$ mkdir -p $BLUECAT_PREFIX/demo/zebra/local
```

2. Set up the BlueCat Linux kernel configuration by using the standard kernel configuration tools and copy kernel configuration file to the `$BLUECAT_PREFIX/demo/zebra` directory. For instance, type the following commands:

```
BlueCat:$ cd $BLUECAT_PREFIX/usr/src/linux
```

```
BlueCat:$ make xconfig
```

```
BlueCat:$ cp .config \  
$BLUECAT_PREFIX/demo/zebra/zebra.config
```

NOTE: In the kernel configuration the following options must be set to **Y**:

```
CONFIG_NETLINK=Y
```

```
CONFIG_RTNETLINK=Y
```

By default Zebra is configured to communicate with the kernel via the netlink socket.

3. Create the BlueCat kernel downloadable image (`zebra.kernel`):

```
BlueCat:$ cd $BLUECAT_PREFIX/demo/zebra
```

```
BlueCat:$ mkkernel ./zebra.config ./zebra.kernel \  
./zebra.disk
```

4. Create a specification file (`zebra.spec`) that contains the following minimal directives:

```
strip on  
  
mkdir /dev  
mknod /dev/console c 5 1  
ln -s /dev/console /dev/tty  
ln -s /dev/console /dev/tty1  
# Standard 16550 serial driver device  
mknod /dev/ttyS0 c 4 64  
mknod /dev/ttyS1 c 4 65  
  
mkdir -p /lib/security  
mkdir -p /usr/lib  
mkdir /bin
```

```

mkdir    /sbin
mkdir -p /etc/rc.d
mkdir -p /etc/pam.d
mkdir -p /etc/xinetd.d
mkdir -p /etc/zebra
mkdir    /proc
mkdir    /tmp
mkdir -p /usr/bin
mkdir -p /usr/sbin
mkdir -p /var/run
mkdir -p /usr/libexec

mkdir -p /var/log/zebra

mkdir /root

mkdir /dev/pts
mknod /dev/ptmx c 5 2

chmod 0666 /dev/ptmx

cp ./local/fstab ./local/passwd ./local/inittab ./local/mtab /etc
cp ./local/other                               /etc/pam.d
cp ./local/rc.sysinit                          /etc/rc.d
cp ./local/hosts                               /etc
cp ./local/protocols                          /etc
cp ./local/resolv.conf                        /etc
cp ${BLUECAT_PREFIX}/etc/pwdb.conf           /etc
cp ${BLUECAT_PREFIX}/etc/nsswitch.conf       /etc
cp ${BLUECAT_PREFIX}/etc/services            /etc

cp ${BLUECAT_PREFIX}/etc/security /etc

cp ./local/shadow                             /etc
cp ./local/pam.d                             /etc
cp ./local/xinetd.d/*                         /etc/xinetd.d
cp ./local/zebra.conf                         /etc/zebra/

cp ${BLUECAT_PREFIX}/lib/libnss_files-*.so   /lib
cp ${BLUECAT_PREFIX}/lib/libnss_dns-*.so     /lib
cp ${BLUECAT_PREFIX}/lib/libpwdb.so         /lib
cp ${BLUECAT_PREFIX}/lib/security           /lib

cp ./local/empty    /var/log/wtmp

lcd ${BLUECAT_PREFIX}/sbin
cp reboot init mingetty ifconfig /sbin

cp ${BLUECAT_PREFIX}/lib/security/pam_permit.so /lib/security

cp ${BLUECAT_PREFIX}/etc/xinetd.conf          /etc

cp ${BLUECAT_PREFIX}/usr/bin/telnet           /usr/bin

cp ${BLUECAT_PREFIX}/etc/shells              /etc
chmod 644 /etc/shells

cp ${BLUECAT_PREFIX}/etc/group               /etc

#
# General Binaries
#
lcd ${BLUECAT_PREFIX}/bin

```

```
cp ping mount bash cat ls hostname ps /bin
cp login /bin
ln -s /bin/bash /bin/sh

cp ${BLUECAT_PREFIX}/usr/bin/vtysch /usr/bin

# internet services utils
cp ${BLUECAT_PREFIX}/usr/sbin/xinetd /usr/sbin
cp ${BLUECAT_PREFIX}/usr/sbin/in.telnetd /usr/sbin
cp ${BLUECAT_PREFIX}/usr/sbin/zebra /usr/sbin

chmod 711 /etc/rc.d/rc.sysinit

chmod 755 /bin /sbin /usr/bin /usr/sbin

# End of File
```

5. Create the `local/inittab` file with the following contents:

```
id:1:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/sbin/halt
16:6:wait:/sbin/reboot

ca::ctrlaltdel:/sbin/shutdown -t3 -r now

pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System
Shutting Down"

pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown
Cancelled"

1:12345:respawn:/sbin/mingetty tty1
```

6. Create the `local/rc.sysinit` file with the following contents:

```
#!/bin/sh

PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

mount -a
xinetd -stayalive -reuse

hostname myhostname

zebra -d
```

7. Create the `local/zebra.conf` file with the following contents:

```
!
! zebra configuration file
!
hostname Router
password zebra
enable password zebra
!
! Interface's description.
```



```
.....  
.....  
.....  
.....  
.....  
Start = 0xa1000000, range = (0xa1000000,0xa13a35ff), format = SREC  
YAMON> go
```

where *<host_IP>* is an IP address of the host.

As a result, the Zebra utility is loaded onto the target and automatically started.

Using Zebra Utility

This section provides an examples of using the Zebra utility:

```
myhostname login: root  
bash-2.04# ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:D0:A0:00:00:8A  
          inet addr:172.17.3.9  Bcast:172.17.255.255  Mask:255.255.0.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:100  
          Interrupt:10 Base address:0x1200  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
  
bash-2.04# ping -c 2 172.17.0.1  
PING 172.17.0.1 (172.17.0.1) from 172.17.1.3 : 56(84) bytes of data.  
64 bytes from 172.17.0.1: icmp_seq=0 ttl=255 time=535 usec  
64 bytes from 172.17.0.1: icmp_seq=1 ttl=255 time=470 usec  
  
--- 172.17.0.1 ping statistics ---  
2 packets transmitted, 2 packets received, 0% packet loss  
round-trip min/avg/max/mdev = 0.470/0.502/0.535/0.039 ms  
bash-2.04#  
bash-2.04# telnet localhost 2601  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
  
Hello, this is zebra (version 0.91a).  
Copyright 1996-2001 Kunihiro Ishiguro.  
  
User Access Verification  
  
Password: zebra  
Router> enable  
Password: zebra  
Router# show ip route  
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
```

```
B - BGP, > - selected route, * - FIB route
S>* 0.0.0.0/0 [1/0] via 172.17.0.1, eth0
C>* 127.0.0.0/8 is directly connected, lo
C>* 172.17.0.0/16 is directly connected, eth0
Router#
```


The following table shows the device drivers supported by the malta BSP:

Table 5-1: Device Drivers Supported by the malta BSP

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
UART 16550-compatible	serial.c	drivers/char	CONFIG_SERIAL CONFIG_SERIAL_CONSOLE	
Real Time Clock	rtc.c mcl46818rtc.h mcl46818rtc.h	drivers/char include/linux include/asm-mips	CONFIG_RTC	
ASCII LED display	malta_ascii_led.c malta_ascii_led.h	drivers/char include/asm-mips	CONFIG_BLUECAT_MALTA_ASCII_LED	
IDE Controller	Generic IDE drivers	drivers/ide/ ide-*.c	CONFIG_IDE CONFIG_BLK_DEV_IDE CONFIG_BLK_DEV_IDEPCI	
Ethernet Controller AMD Am79C973	pcnet32.c	drivers/net	CONFIG_NETDEVICES CONFIG_NET_ETHERNET CONFIG_NET_PCI CONFIG_PCNET32	
NOR Flash Intel FlashFile	malta.c	drivers/mtd/maps	CONFIG_MTD CONFIG_MTD_PARTITIONS CONFIG_MTD_CHAR CONFIG_MTD_BLOCK CONFIG_MTD_MALTA CONFIG_MTD_MALTA_PART CONFIG_MTD_CFI CONFIG_MTD_CFI_INTELEXT	Supported via JFFS

This chapter describes the BlueCat Linux Application Interface specific to the MIPS Malta board.

ASCII LED Display Control

BlueCat Linux for the MIPS Malta board has a device driver for on-board ASCII LED display. The driver is configured by the `CONFIG_BLUECAT_MALTA_ASCII_LED` kernel configuration option. This option is located in the General Setup submenu of the `make xconfig` user interface. The driver provides an `ioctl()` interface for the user-space applications, and defined in the `$BLUECAT_PREFIX/usr/include/asm/malta_ascii_led.h` file.

The driver is accompanied by the `$BLUECAT_PREFIX/sbin/malta_led` utility to provide control of the ASCII LED display from the shell command line or shell scripts. The utility is included in the `showcase` package. Its sources are in the `$BLUECAT_PREFIX/demo/showcase/src.malta` directory.

The utility syntax is as follows:

```
malta_led <string>
```

where `<string>` is an array of 8 symbols to be displayed on the ASCII LED display.

For example:

```
# malta_led Hello!  
# malta_led 01234567
```

The ASCII LED driver automatically creates a special device file (`/dev/asciiled`) in the root file system. The major number of this device is 223.

Defect Fixes and Known Limitations

Defect Fixes for the MIPS Malta Board

The table below shows defect fixes in this release of BlueCat Linux:

Table 7-1: Defect Fixes in BlueCat Linux

Platform	Subcomponent	ID	Summary
All	BlueCat Linux Misc	16057	Ctrl-C, Ctrl-Z etc., do not work from shell.
All	BlueCat Linux OS Loader	16358	The BLOSH <code>ntar</code> command hangs the system.
All	BlueCat Linux Misc	17308	Certain sequence of file updates sometimes causes FFS to crash after reboot.
All	BlueCat Linux Debuggers	18404	Update BlueCat Linux <code>gdbserver</code> to process signals to <code>gdb</code> on the host correctly.
All	BlueCat Linux Linux Misc	18696	Provide support for NFS in BlueCat Linux.
Windows (cross)	BlueCat Linux installation procedure	16130	Windows install should support non-C drives.

MIPS Malta Known Limitations and Workarounds

- Even though the `developer.flash.srec` boot image is created while building the `developer demo` system, it cannot be programmed into Flash memory, because the size of this `srec` image exceeds the Flash capacity (4 MB).

- Since the YAMON boot loader supports only the SREC format as a boot image format, direct booting of BlueCat Linux KDIs from YAMON is not possible. Before booting, BlueCat Linux KDIs should be converted to the SREC format using the `objcopy` command.
- If `mkrootfs` is terminated (either by an error or by a signal), it tries to clean all its temporary files before exiting. However, due to certain features of the Cygwin execution environment, such temporary files can remain uncleaned in the `/tmp` directory on a Windows host. It is recommended that the `/tmp` directory be regularly checked and cleaned.
- The `tc1x` RPM package is not included in the Windows-hosted distribution.
- Debugging of multithreaded applications via GDB is not supported.
- On Windows hosts, some file permissions (including `r` and `s`) always have default values. To set permissions different from the default values, the `chmod` command should be used in the `.spec` file.