

BlueCat Linux Board Support Guide

BlueCat Linux 4.0
DOC-0548-00

For Force PPMC260 Boards

Product names mentioned in *BlueCat Linux Board Support Guide for Force PPMC260 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 Force PPMC260 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

CHAPTER 1	OVERVIEW	1
CHAPTER 2	DOWNLOADING AND BOOTING BLUECAT LINUX ON THE TARGET	3
	Prerequisites	3
	Download and Boot Procedure Overview	4
	Downloading BlueCat Linux into Flash	4
	Booting BlueCat Linux from Flash	5
	Booting BlueCat Linux from a Network	6
CHAPTER 3	DEFAULT KERNEL CONFIGURATION PARAMETERS	7
CHAPTER 4	SUPPORTED DEMO SYSTEMS.....	33
	Demo Systems	33
	developer Demo System	33
	osloader Demo System	34
	showcase Demo System	34
	Using Selected RPM Packages	34
	Using BusyBox	34
	Creating a BlueCat Linux System for BusyBox	34
	Using TinyLogin RPM Package	40
	Using Zebra RPM Package	44
CHAPTER 5	SUPPORTED DEVICE DRIVERS	53
CHAPTER 6	DEFECT FIXES AND KNOWN LIMITATIONS.....	55
	Force PPMC260 Target Board Problems and Limitations	55

CHAPTER 7	DEFECT FIXES AND KNOWN LIMITATIONS	59
	Force PPMC260 Target Board Problems and Limitations	59

This *BlueCat Linux Board Support Guide for Force PPMC260 Boards* provides information about the BlueCat Linux Board Support Package (BSP) for the Force PPMC260 boards, which are based on the PowerPC 750 processor.

Throughout this Board Support Guide (BSG), the BSP is referred to as the “*ppmc260*” and the board as the “*Force PPMC260*”.

This BSG provides the following information:

- *Chapter 1* is an overview of this BSG’s individual chapters.
- *Chapter 2* describes BlueCat Linux downloading and booting procedures for the Force PPMC260 target.
- *Chapter 3* details configuration of the prebuilt BlueCat Linux kernel contained in the *ppmc260* BSP.
- *Chapter 4* in this guide lists the BlueCat Linux demo systems included in the *ppmc260* BSP.
- *Chapter 5* lists the device drivers supported by the Force PPMC260 BSP.



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

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 ppmc260 BSP have been installed on the development host. The user must:

1. Install the BlueCat Linux PowerPC Core onto the cross development host, as described in the “Installing the Default Configuration” section in Chapter 1, “Installation” of the *BlueCat Linux User's Guide*.
2. Install the ppmc260 BSP onto the cross development host as detailed in the “Installing Target Board Support” section of Chapter 1, “Installation” in the *BlueCat Linux User's Guide*.

-
3. Activate support for the `ppmc260` BSP as detailed in the “Activating Support for a Target Board” section of Chapter 1, “Installation” in the *BlueCat Linux User’s Guide*.

NOTE: The default BlueCat Linux kernel for Force PPMC260 is built for 512MB of RAM.

If the PPMC260 board uses a different amount of memory, the BlueCat Linux kernel must be recompiled with the appropriate RAM amount. The following kernel command line option must be used:

```
mem=<memory>M
```

where `<memory>` is the amount of memory on the board.

This command can be edited in the kernel Makefile or set manually from the BLOSH prompt. For example, from BLOSH:

```
> s CMD mem=256M
```

Download and Boot Procedure Overview

To provide a convenient boot mechanism, the procedure for downloading and booting BlueCat Linux onto a Force PPMC260 target consists of the following main steps:

- Downloading a BlueCat Linux embedded system into Flash
- Booting a BlueCat Linux embedded system.

Downloading BlueCat Linux into Flash

This section provides instructions on how a BlueCat Linux embedded system can be downloaded into the target flash memory using the BlueCat Linux OS loader. Refer also to the *BlueCat Linux User’s Guide* for details on BlueCat Linux OS loader.

These instructions are applicable to any of the demo systems. Specifically, this chapter uses the `showcase` demo system as an example.

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

- Boot the `i_osloader` demo system onto the target by typing the following commands at the `PowerBoot>` prompt:

```
PowerBoot> netload i_osloader.kdi 1000000 <client_ip> \  
<server_ip>
```

where `<client_ip>` is an IP address of the target, `<server_ip>` is an IP address of the development host.

- At the `PowerBoot>` prompt, enter:

```
PowerBoot> go 1007000
```

This command starts the BlueCat Linux OS loader.

- At the BlueCat OS loader prompt (`>`), type the following commands:

```
> set IF eth0  
> set IP <client_IP>  
> set HOST <server_ip>  
> set FILE tftp showcase.kdi  
> flash /dev/mtdchar0 erase
```

where `<client_ip>` is an IP address of the target, `<server_ip>` is an IP address of the development host.

After these commands have been performed, the `showcase` demo is programmed into Flash and can be booted as described in the “Booting BlueCat Linux from Flash” section.

Booting BlueCat Linux from Flash

The BlueCat Linux embedded system programmed into Flash can be started in the following way (suppose that the BlueCat Linux image `showcase.kdi` is located at the start of the target Flash memory).

- At the `PowerBoot` prompt, enter the following command:

```
PowerBoot> go 50007000
```

Booting BlueCat Linux from a Network

Type the following commands at the PowerBoot prompt to boot the BlueCat Linux showcase demo system onto a Force PPMC260 target from a network:

```
PowerBoot> netload showcase.kdi 1000000 <client_ip> \  
<server_ip>
```

```
PowerBoot> go 1007000
```

where *<client_ip>* is an IP address of the target, *<server_ip>* is an IP address of the development host.

Default Kernel Configuration Parameters

The ppmc260 BSP comes with a default BlueCat Linux kernel. This kernel has a number of default configuration parameters. These parameters are detailed in the tables listed in Table 3-1 "Kernel Configuration Options".

Table 3-1: Kernel Configuration Options

Parameters	Table Number
Code Maturity Level Options	Table 3-2
Loadable Module Support	Table 3-3
Platform Support	Table 3-4
Marvell/Galileo GT64260 Options	Table 3-5
General Setup	Table 3-6
Parallel Port Support	Table 3-7
Memory Technology Devices (MTD)	Table 3-8
RAM/ROM/Flash Chip Drivers	Table 3-9
Mapping Drivers for Chip Access	Table 3-10
Self-Contained MTD Device Drivers	Table 3-11
NAND Flash Device Drivers	Table 3-12
Plug and Play Configuration	Table 3-13
Block Devices	Table 3-14
Multi-Device Support (RAID and LVM)	Table 3-15
Networking Options	Table 3-16
IP: Netfilter Configuration	Table 3-17
QoS and/or Queuing	Table 3-18

Table 3-1: Kernel Configuration Options (Continued)

Parameters	Table Number
ATA/IDE/MFM/RLL Support	Table 3-19
IDE, ATA, and ATAPI Block Devices	Table 3-20
SCSI Support	Table 3-21
IEEE 1394 (FireWire) Support (Experimental)	Table 3-22
Network Device Support	Table 3-23
ARCnet Devices	Table 3-24
Appletalk Devices	Table 3-25
Ethernet (10 or 100 Mbit)	Table 3-26
Ethernet (1000 Mbit)	Table 3-27
Wireless LAN (Non-Hamradio)	Table 3-28
Token Ring Devices	Table 3-29
WAN Interfaces	Table 3-30
Amateur Radio Support	Table 3-31
IrDA (Infrared) Support	Table 3-32
ISDN Subsystem	Table 3-33
Old CD-ROM Drivers (not SCSI, not IDE)	Table 3-34
Console Drivers	Table 3-35
Frame-buffer support	Table 3-36
Input Core Support	Table 3-37
Character Devices	Table 3-38
Serial Drivers	Table 3-39
I2C Support	Table 3-40
L3 Serial Bus Support	Table 3-41
Mice	Table 3-42
Joysticks	Table 3-43
Watchdog Cards	Table 3-44
Ftape and Floppy Tape Device Driver	Table 3-45

Table 3-1: Kernel Configuration Options (Continued)

Parameters	Table Number
Multimedia Devices	Table 3-46
File Systems	Table 3-47
Network File Systems	Table 3-48
Partition Types	Table 3-49
Sound	Table 3-50
USB Support	Table 3-51
Bluetooth Support	Table 3-52
Kernel Hacking	Table 3-53
Modular Advanced Power Management	Table 3-54

Table 3-2: Code Maturity Level Options

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

Table 3-3: 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

Table 3-4: Platform Support

Option	Value	Description
CONFIG_6xx	Y	Processor Type
CONFIG_8260	N	MPC8260 CPM Support
CONFIG_PPMC260	Y	Machine Type
CONFIG_SMP	N	Symmetric multi-processing support
CONFIG_ALTIVEC	N	AltiVec Support
CONFIG_TAU	N	Thermal Management Support

Table 3-5: Marvell/Galileo GT64260 Options

Option	Value	Description
CONFIG_GT64260_BUS_CLOCK_133MHZ	Y	133 Mhz GT64260 memory bus clock(default 100 Mhz)
CONFIG_GT64260_ETH	Y	GT64260 Ethernet Ports
CONFIG_GT64260_ETH_0	Y	EVB64260 - Ethernet Port 0
CONFIG_GT64260_ETH_0_MACADDR	FEFFFF000000	MAC Address
CONFIG_GT64260_ETH_1	Y	EVB64260 - Ethernet Port 1
CONFIG_GT64260_ETH_1_MACADDR	FEFFFF000001	"MAC Address
CONFIG_GT64260_ETH_2	Y	EVB64260 - Ethernet Port 2
CONFIG_GT64260_ETH_2_MACADDR	FEFFFF000002	MAC Address
CONFIG_GT64260_MPSC	Y	GT64260 MPSC Serial Ports
CONFIG_GT64260_MPSC_0	Y	MPSC Port 0
CONFIG_GT64260_MPSC_1	N	MPSC Port 1
CONFIG_GT64260_CONSOLE	Y	MPSC Port 0 as system console

Table 3-6: General Setup

Option	Value	Description
CONFIG_HIGHMEM	N	High memory support (Experimental)
CONFIG_BLUECAT_IGNORE_PRINTK	N	BlueCat Ignore printk
CONFIG_BLUECAT_LOADER	N	BlueCat OS Loader
CONFIG_BLUECAT_SMALL_FOOTPRINT	N	BlueCat small memory footprint
CONFIG_NET	Y	Networking support
CONFIG_BLUECAT_MEMSIZE	N	Memory sizing benchmarks
CONFIG_SYSCTL	Y	Sysctl support
CONFIG_SYSVIPIC	Y	System V IPC
CONFIG_BSD_PROCESS_ACCT	N	BSD Process Accounting
CONFIG_BINFMT_MISC	N	Kernel support for MISC binaries
CONFIG_PCI_NAMES	Y	PCI device name database
CONFIG_HOTPLUG	N	Support for hot-pluggable devices
CONFIG_PPC_RTC	Y	Support for /dev/rtc
CONFIG_CMDLINE_BOOL	N	Default bootloader kernel arguments

Table 3-7: Parallel Port Support

Option	Value	Description
CONFIG_PARPORT	N	Parallel port 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

Table 3-8: Memory Technology Devices (MTD)(Continued)

Option	Value	Description
CONFIG_MTD_REDBOOT_PARTS	N	RedBoot partition table parsing
CONFIG_MTD_CHAR	Y	Direct char device access to MTD devices
CONFIG_MTD_BLOCK	Y	Caching block device access to MTD devices
CONFIG_FTL	N	FTL (Flash Translation Layer) support
CONFIG_NFTL	N	NFTL (NAND Flash Translation Layer) support

Table 3-9: 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 (theoretically obsoleted now) drivers for non-CFI chips

Table 3-10: Mapping Drivers for Chip Access

Option	Value	Description
CONFIG_MTD_PHYSMAP	N	CFI Flash device in physical memory map
CONFIG_MTD_TQM8XXL	N	CFI Flash device mapped on TQM8XXL

Table 3-10: Mapping Drivers for Chip Access(Continued)

Option	Value	Description
CONFIG_MTD_RPXLITE	N	CFI Flash device mapped on RPX Lite or CLLF
CONFIG_MTD_CFI_FLAGADM	N	CFI Flash device mapping on FlagaDM
CONFIG_MTD_PPMC260	Y	Flash chip mapping on PPMC260 board
CONFIG_MTD_PPMC260_PART	:	Partitions layout
CONFIG_MTD_PCI	N	PCI MTD driver

Table 3-11: 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
CONFIG_MTD_DOC1000	N	M-Systems Disk-On-Chip 1000
CONFIG_MTD_DOC2000	N	M-Systems Disk-On-Chip 2000 and Millennium
CONFIG_MTD_DOC2001	N	M-Systems Disk-On-Chip Millennium-only alternative driver (see Help)

Table 3-12: NAND Flash Device Drivers

Option	Value	Description
CONFIG_MTD_NAND	N	NAND Device Support

Table 3-13: Plug and Play Configuration

Option	Value	Description
CONFIG_PNP	N	Plug and Play support

Table 3-14: 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 5xxx 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 (initrd) support
CONFIG_BLUECAT_RFS	Y	BlueCat RFS support

Table 3-15: Multi-device Support (RAID and LVM)

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

Table 3-16: Networking Options

Option	Value	Description
CONFIG_PACKET	Y	Packet socket
CONFIG_PACKET_MMAP	N	Packet socket: mmapped IO
CONFIG_NETLINK	Y	Kernel/User netlink socket
CONFIG_RTNETLINK	N	Routing messages
CONFIG_NETLINK_DEV	N	Netlink device emulation
CONFIG_NETFILTER	Y	Network packet filtering (replaces ipchains)
CONFIG_NETFILTER_DEBUG	N	Network packet filtering debugging
CONFIG_FILTER	N	Socket Filtering
CONFIG_UNIX	Y	Unix domain sockets
CONFIG_INET	Y	TCP/IP networking
CONFIG_IP_MULTICAST	Y	IP: multicasting
CONFIG_IP_ADVANCED_ROUTER	N	IP: advanced router
CONFIG_IP_PNP	N	IP: kernel level autoconfiguration
CONFIG_NET_IPIP	N	IP: tunneling
CONFIG_NET_IPGRE	N	IP: GRE tunnels over IP
CONFIG_IP_MROUTE	N	IP: multicast routing
CONFIG_INET_ECN	N	IP: TCP Explicit Congestion Notification support
CONFIG_SYN_COOKIES	Y	IP: TCP syncookie support (disabled 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	M	Appletalk protocol support
CONFIG_DECNET	N	DECnet Support

Table 3-16: Networking Options(Continued)

Option	Value	Description
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
CONFIG_NET_FASTROUTE	N	Fast switching (read Help!)
CONFIG_NET_HW_FLOWCONTROL	N	Forwarding between high speed interfaces

Table 3-17: IP: Netfilter Configuration

Option	Value	Description
CONFIG_IP_NF_CONNTRACK	M	Connection tracking (required for masq/NAT)
CONFIG_IP_NF_FTP	M	FTP protocol support
CONFIG_IP_NF_IRC	N	IRC protocol support
CONFIG_IP_NF_QUEUE	N	Userspace queueing via NETLINK (Experimental)
CONFIG_IP_NF_IPTABLES	M	IP tables support (required for filtering/masq/NAT)
CONFIG_IP_NF_MATCH_LIMIT	M	limit match support
CONFIG_IP_NF_MATCH_MAC	M	MAC address match support
CONFIG_IP_NF_MATCH_MARK	M	netfilter MARK match support
CONFIG_IP_NF_MATCH_MULTIPORT	M	Multiple port match support
CONFIG_IP_NF_MATCH_TOS	M	TOS match support
CONFIG_IP_NF_MATCH_TCPMSS	M	tcpmss match support
CONFIG_IP_NF_MATCH_STATE	M	Connection state match support
CONFIG_IP_NF_MATCH_UNCLEAN	M	Unclean match support (Experimental)

Table 3-17: IP: Netfilter Configuration(Continued)

Option	Value	Description
CONFIG_IP_NF_MATCH_OWNER	M	Owner match support (Experimental)
CONFIG_IP_NF_FILTER	M	Packet filtering
CONFIG_IP_NF_TARGET_REJECT	M	REJECT target support
CONFIG_IP_NF_TARGET_MIRROR	M	MIRROR target support (Experimental)
CONFIG_IP_NF_NAT	M	Full NAT
CONFIG_IP_NF_TARGET_MASQUERADE	M	MASQUERADE target support
CONFIG_IP_NF_TARGET_REDIRECT	M	REDIRECT target support
CONFIG_IP_NF_MANGLE	N	Packet mangling
CONFIG_IP_NF_TARGET_LOG	N	LOG target support
CONFIG_IP_NF_TARGET_TCPMSS	M	TCPMSS target support
CONFIG_IP_NF_COMPAT_IPCHAINS	M	ipchains (2.2-style) support
CONFIG_IP_NF_COMPAT_IPFWADM	N	ipfwadm (2.0-style) support

Table 3-18: QoS and/or Fair Queueing

Option	Value	Description
CONFIG_NET_SCHED	N	QoS and/or fair queueing

Table 3-19: ATA/IDE/MFM/RLL Support

Option	Value	Description
CONFIG_IDE	Y	ATA/IDE/MFM/RLL support

Table 3-20: IDE, ATA and ATAPI Block Devices

Option	Value	Description
CONFIG_BLK_DEV_IDE	Y	Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
CONFIG_BLK_DEV_HD_IDE	N	Use old disk-only driver on primary interface
CONFIG_BLK_DEV_IDEDISK	Y	Include IDE/ATA-2 DISK support
CONFIG_IDEDISK_MULTI_MODE	N	Use multi-mode by default
CONFIG_BLK_DEV_IDECD	Y	Include IDE/ATAPI CDROM support
CONFIG_BLK_DEV_IDETAPE	N	Include IDE/ATAPI TAPE support
CONFIG_BLK_DEV_IDEFLOPPY	Y	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
CONFIG_IDEPCI_SHARE_IRQ	Y	Sharing PCI IDE interrupts support
CONFIG_BLK_DEV_IDEDMA_PCI	Y	Generic PCI bus-master DMA support
CONFIG_BLK_DEV_OFFBOARD	N	Boot off-board chipsets first support
CONFIG_IDEDMA_PCI_AUTO	Y	Use PCI DMA by default when available
CONFIG_IDEDMA_PCI_WIP	N	ATA Work(s) In Progress (Experimental)
CONFIG_BLK_DEV_AEC62XX	N	AEC62XX chipset support
CONFIG_BLK_DEV_ALI15X3	N	ALI M15x3 chipset support
CONFIG_BLK_DEV_AMD74XX	N	AMD Viper support
CONFIG_BLK_DEV_CMD64X	Y	CMD64X chipset support
CONFIG_BLK_DEV_CY82C693	N	CY82C693 chipset support
CONFIG_BLK_DEV_CS5530	N	Cyrix CS5530 MediaGX chipset support
CONFIG_BLK_DEV_HPT34X	N	HPT34X chipset support
CONFIG_BLK_DEV_HPT366	N	HPT366 chipset support
CONFIG_BLK_DEV_NS87415	N	NS87415 chipset support (Experimental)

Table 3-20: IDE, ATA and ATAPI Block Devices(Continued)

Option	Value	Description
CONFIG_BLK_DEV_OPTI621	N	OPTi 82C621 chipset enhanced support (Experimental)
CONFIG_BLK_DEV_PDC202XX	N	PROMISE PDC202{46 62 65 67 68} support
CONFIG_BLK_DEV_SVWKS	N	ServerWorks OSB4/CSB5 chipsets support
CONFIG_BLK_DEV_SIS5513	N	SiS5513 chipset support
CONFIG_BLK_DEV_SLC90E66	N	SLC90E66 chipset support
CONFIG_BLK_DEV_TRM290	N	Tekram TRM290 chipset support (Experimental)
CONFIG_BLK_DEV_VIA82CXXX	N	VIA82CXXX chipset support
CONFIG_BLK_DEV_SL82C105	Y	Winbond SL82c105 support
CONFIG_IDE_CHIPSETS	N	Other IDE chipset support
CONFIG_IDEDMA_IVB	N	IGNORE word93 Validation BITS
CONFIG_BLK_DEV_ATA RAID	N	Support for IDE Raid controllers

Table 3-21: SCSI Support

Option	Value	Description
CONFIG_SCSI	N	SCSI support

Table 3-22: IEEE 1394 (FireWire) Support (Experimental)

Option	Value	Description
CONFIG_IEEE1394	N	IEEE 1394 (FireWire) support (Experimental)

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	QL (serial line load balancing) support
CONFIG_TUN	N	Universal TUN/TAP device driver support
CONFIG_ETHERTAP	N	Ethertap network tap (Obsolete)
CONFIG_FDDI	N	FDDI driver support
CONFIG_HIPPI	N	HIPPI driver support (Experimental)
CONFIG_PPP	N	PPP (point-to-point protocol) support
CONFIG_SLIP	N	SLIP (serial line) 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: Appletalk Devices

Option	Value	Description
CONFIG_APPLETALK	N	Appletalk interfaces support

Table 3-26: Ethernet (10 or 100 Mbit)

Option	Value	Description
CONFIG_NET_ETHERNET	Y	Ethernet (10 or 100Mbit)
CONFIG_MACE	N	MACE (Power Mac ethernet) support
CONFIG_BMAC	N	BMAC (G3 ethernet) support
CONFIG_GMAC	N	GMAC (G4/iBook ethernet) support
CONFIG_OAKNET	N	National DP83902AV (Oak ethernet) support
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 PCI support
CONFIG_ADAPTEC_STARFIRE	N	Adaptec Starfire support (Experimental)
CONFIG_TULIP	N	DECchip Tulip (dc21x4x) PCI support
CONFIG_DE4X5	Y	Generic DECchip & DIGITAL EtherWORKS PCI/EISA
CONFIG_BLUECAT_DE4X5_WORKAROUND	N	Fix for buggy SROM on Motorola MCP(N)750 cPCI board
CONFIG_DGRS	N	Digi Intl. RightSwitch SE-X support
CONFIG_DM9102	N	Davicom DM910x/DM980x support
CONFIG_EEPRO100	Y	EtherExpressPro/100 support
CONFIG_FEALNX	N	Myson MTD-8xx PCI Ethernet support
CONFIG_NATSEMI	N	National Semiconductor DP8381x series PCI Ethernet support
CONFIG_NE2K_PCI	N	PCI NE2000 and clones support (see Help)

Table 3-26: Ethernet (10 or 100 Mbit)(Continued)

Option	Value	Description
CONFIG_8139TOO	N	RealTek RTL-8139 PCI Fast Ethernet Adapter support
CONFIG_SIS900	N	SiS 900/7016 PCI Fast Ethernet Adapter support
CONFIG_EPIC100	N	SMC EtherPower II
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 Ethernet support
CONFIG_NET_POCKET	N	Pocket and portable adapters

Table 3-27: 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 Semiconduct DP83820 support
CONFIG_HAMACHI	N	Packet Engines Hamachi GNIC-II support
CONFIG_YELLOWFIN	N	Packet Engines Yellowfin Gigabit-NIC support (Experimental)
CONFIG_SK98LIN	N	SysKonnnect SK-98xx support

Table 3-28: Wireless LAN (Non-hamradio)

Option	Value	Description
CONFIG_NET_RADIO	N	Wireless LAN (non-hamradio)

Table 3-29: Token Ring Devices

Option	Value	Description
CONFIG_TR	N	Token Ring driver support

Table 3-30: WAN Interfaces

Option	Value	Description
CONFIG_WAN	N	WAN interfaces support

Table 3-31: Amateur Radio Support

Option	Value	Description
CONFIG_HAMRADIO	N	Amateur Radio support

Table 3-32: IrDA (Infrared) Support

Option	Value	Description
CONFIG_IRDA	N	IrDA subsystem support

Table 3-33: ISDN Subsystem

Option	Value	Description
CONFIG_ISDN	N	ISDN support

Table 3-34: Old CD-ROM Drivers (not SCSI, not IDE)

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

Table 3-35: Console Drivers

Option	Value	Description
CONFIG_VGA_CONSOLE	N	Support for VGA Console

Table 3-36: Frame-buffer support

Option	Value	Description
CONFIG_FB	N	Support for frame buffer devices (Experimental)

Table 3-37: Input Core Support

Option	Value	Description
CONFIG_INPUT	N	Input core support

Table 3-38: 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

Table 3-38: Character Devices (Continued)

Option	Value	Description
CONFIG_SERIAL_EXTENDED	N	Extended dumb serial driver options
CONFIG_SERIAL_NONSTANDARD	N	Non-standard serial port support
CONFIG_UNIX98_PTYS	Y	Unix98 PTY support
CONFIG_UNIX98_PTY_COUNT	256	Maximum number of Unix98 PTYs in use (0-2048)
CONFIG_QIC02_TAPE	N	QIC-02 tape support
CONFIG_INTEL_RNG	N	Intel i8x0 Random Number Generator support
CONFIG_NVRAM	N	/dev/nvram support
CONFIG_RTC	N	Enhanced Real Time Clock Support
CONFIG_DTLK	N	Double Talk PC internal speech card support
CONFIG_R3964	N	Siemens R3964 line discipline
CONFIG_APPLICOM	N	Applicom intelligent fieldbus card support
CONFIG_AGP	N	/dev/agpgart (AGP Support) (Experimental)
CONFIG_DRM	N	Direct Rendering Manager (XFree86 DRI support)
CONFIG_MWAVE	N	ACP Modem (Mwave) support

Table 3-39: Serial Drivers

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

Table 3-40: I2C Support

Option	Value	Description
CONFIG_I2C	N	I2C support

Table 3-41: L3 Serial Bus Support

Option	Value	Description
CONFIG_L3	N	L3 support

Table 3-42: Mice

Option	Value	Description
CONFIG_BUSMOUSE	Y	Bus Mouse Support
CONFIG_ATIXL_BUSMOUSE	N	ATIXL busmouse support
CONFIG_LOGIBUSMOUSE	N	Logitech busmouse support
CONFIG_MS_BUSMOUSE	N	Microsoft busmouse support
CONFIG_MOUSE	Y	Mouse Support (not serial and bus mice)
CONFIG_PSMOUSE	Y	PS/2 mouse (aka "auxiliary device") support
CONFIG_82C710_MOUSE	N	C&T 82C710 mouse port support (as on TI Travelmate)
CONFIG_PC110_PAD	N	PC110 digitizer pad support

Table 3-43: Joysticks

Option	Value	Description
CONFIG_QIC02_TAPE	N	QIC-02 tape support

Table 3-44: Watchdog Cards

Option	Value	Description
CONFIG_WATCHDOG	N	Watchdog Timer Support

Table 3-45: Ftape, the Floppy Tape Device Driver

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

Table 3-46: Multimedia Devices

Option	Value	Description
CONFIG_VIDEO_DEV	N	Video for Linux

Table 3-47: FileSystems

Option	Value	Description
CONFIG_QUOTA	N	Quota support
CONFIG_AUTOFS_FS	N	Kernel automounter support
CONFIG_AUTOFS4_FS	N	Kernel automounter version 4 support (also supports v3)
CONFIG_REISERFS_FS	N	Reiserfs support
CONFIG_ADFS_FS	N	ADFS file system support
CONFIG_AFFS_FS	N	Amiga FFS file system support (Experimental)
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 (Experimental)
CONFIG_EXT3_FS	N	Ext3 journaling file system support (Experimental)
CONFIG_FAT_FS	N	DOS FAT fs support
CONFIG_EFS_FS	N	EFS file system support (read only) (Experimental)
CONFIG_JFFS_FS	N	Journaling Flash File System (JFFS) support

Table 3-47: FileSystems(Continued)

Option	Value	Description
CONFIG_JFFS2_FS	N	Journaling Flash File System v2 (JFFS2) support
CONFIG_CRAMFS	N	Compressed ROM file system support
CONFIG_TMPFS	N	Virtual memory file system support (former shm fs)
CONFIG_RAMFS	N	Simple RAM-based file system support
CONFIG_ISO9660_FS	N	ISO 9660 CDROM file system support
CONFIG_MINIX_FS	N	Minix fs support
CONFIG_FREEVXFS_FS	N	FreeVxFS file system support (VERITAS VxFS(TM) compatible)
CONFIG_NTFS_FS	N	NTFS file system support (read only)
CONFIG_HPFS_FS	N	OS/2 HPFS file system support
CONFIG_PROC_FS	Y	/proc file system support
CONFIG_DEVFS_FS	Y	/dev file system support (Experimental)
CONFIG_DEVFS_MOUNT	N	Automatically mount at boot
CONFIG_DEVFS_DEBUG	N	Debug devfs
CONFIG_DEVPTS_FS	Y	/dev/pts file system for Unix98 PTYs
CONFIG_QNX4FS_FS	N	QNX4 file system support (read only) (Experimental)
CONFIG_ROMFS_FS	N	ROM file system support
CONFIG_EXT2_FS	Y	Second extended fs support
CONFIG_SYSV_FS	N	System V/Xenix/V7/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 3-48: Network File Systems

Option	Value	Description
CONFIG_CODA_FS	N	Coda file system support (advanced network fs)
CONFIG_INTERMEZZO_FS	N	InterMezzo file system support (Experimental, replicating fs)
CONFIG_NFS_FS	Y	NFS file system support
CONFIG_NFS_V3	N	Provide NFSv3 client support
CONFIG_NFSD	Y	NFS server support
CONFIG_NFSD_V3	N	Provide NFSv3 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 3-49: Partition Types

Option	Value	Description
CONFIG_PARTITION_ADVANCED	Y	Advanced partition selection
CONFIG_ACORN_PARTITION	N	Acorn partition support
CONFIG_OSF_PARTITION	N	Alpha OSF partition support
CONFIG_AMIGA_PARTITION	N	Amiga partition table support
CONFIG_ATARI_PARTITION	N	Atari partition table support
CONFIG_MAC_PARTITION	Y	Macintosh partition map support
CONFIG_MSDOS_PARTITION	Y	PC BIOS (MSDOS partition tables) support
CONFIG_BSD_DISKLABEL	N	BSD disklabel (FreeBSD partition tables) support
CONFIG_MINIX_SUBPARTITION	N	Minix subpartition support
CONFIG_SOLARIS_X86_PARTITION	N	Solaris (x86) partition table support
CONFIG_UNIXWARE_DISKLABEL	N	Unixware slices support

Table 3-49: Partition Types(Continued)

Option	Value	Description
CONFIG_LDM_PARTITION	N	Windows Logical Disk Manager (Dynamic Disk) support
CONFIG_SGI_PARTITION	N	SGI partition support
CONFIG_ULTRIX_PARTITION	N	Ultrix partition table support
CONFIG_SUN_PARTITION	N	Sun partition tables support

Table 3-50: Sound

Option	Value	Description
CONFIG_SOUND	M	Sound card support
CONFIG_SOUND_BT878	N	BT878 audio dma
CONFIG_SOUND_CMPCI	N	C-Media PCI (CMI8338/8738)
CONFIG_SOUND_EMU10K1	N	Creative SBLive! (EMU10K1)
CONFIG_SOUND_FUSION	N	Crystal SoundFusion (CS4280/461x)
CONFIG_SOUND_CS4281	N	Crystal Sound CS4281
CONFIG_SOUND_ES1370	N	Ensoniq AudioPCI (ES1370)
CONFIG_SOUND_ES1371	N	Creative Ensoniq AudioPCI 97 (ES1371)
CONFIG_SOUND_ESSSOLO1	N	ESS Technology Solo1
CONFIG_SOUND_MAESTRO	N	ESS Maestro, Maestro2, Maestro2E driver
CONFIG_SOUND_MAESTRO3	N	ESS Maestro3/Allegro driver (Experimental)
CONFIG_SOUND_ICH	N	Intel ICH (i8xx) audio support
CONFIG_SOUND_RME96XX	N	RME Hammerfall (RME96XX) support
CONFIG_SOUND_SONICVIBES	N	S3 SonicVibes
CONFIG_SOUND_TRIDENT	N	Trident 4DWave DX/NX, SiS 7018 or ALi 5451 PCI Audio Core
CONFIG_SOUND_MSNDCLAS	N	Support for Turtle Beach MultiSound Classic, Tahiti, Monterey

Table 3-50: Sound(Continued)

Option	Value	Description
CONFIG_SOUND_MSNDPIN	N	Support for Turtle Beach MultiSound Pinnacle, Fiji
CONFIG_SOUND_VIA82CXXX	N	VIA 82C686 Audio Codec
CONFIG_SOUND_OSS	N	OSS sound modules

Table 3-51: USB Support

Option	Value	Description
CONFIG_USB	N	Support for USB
CONFIG_USB_STORAGE_SDDR09	N	SanDisk SDDR-09 (and other SmartMedia) support

Table 3-52: Bluetooth Support

Option	Value	Description
CONFIG_BLUEZ	N	Bluetooth subsystem support

Table 3-53: Kernel Hacking

Option	Value	Description
CONFIG_MAGIC_SYSRQ	N	Magic SysRq key
CONFIG_KGDB	N	Include kgdb kernel debugger
CONFIG_XMON	N	Include xmon kernel debugger
CONFIG_BLUECAT_KDBG	N	Include kdbg kernel debugger
CONFIG_SERIAL_TEXT_DEBUG	Y	Support for early boot texts over serial port

Table 3-54: Modular Advanced Power Management

Option	Value	Description
CONFIG_BLUECAT_APM	N	MAPM support

This chapter provides information about BlueCat Linux demo systems supported by the ppmc260 Board Support Package (BSP).

Demo Systems

The table below lists the demo systems supported in the ppmc260 BSP distribution, the boot devices supported by each demo system, and their respective RAM and ROM requirements.

Table 4-1: Demo Systems Supported by ppmc260 BSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
developer	Network using firmware Network using the OS loader ROM/Flash memory	3713 KB	16384 KB
osloader	Network using firmware Network using the OS loader ROM/Flash memory	859 KB	5120 KB
showcase	Network using firmware Network using the OS loader ROM/Flash memory	2701 KB	13824 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.

osloader Demo System

`osloader` is the BlueCat Linux OS loader used to boot a BlueCat Linux system on the target board. Refer to Chapter 4 of the *BlueCat Linux User's Guide* for details.

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 the full-featured GNU utilities; 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 set of 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.sysinit
chmod 755 /bin /sbin
cp BLUECAT_PREFIX/lib/libnss_files-*.so /lib
# 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 RFS images.

Booting BusyBox Images from a Network

Use the following procedure to boot the BlueCat Linux with BusyBox utility from a network using the PowerBoot 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 PowerBoot prompt, type the following commands:

```
PowerBoot> netload busybox.kdi 1000000 /
<target_ip> <host_ip>

PowerBoot> go 1007000
```

where `<target_ip>` is an IP address of the target, `<host_ip>` is an IP address of the development host.

2. The following sample output appears:

```

loaded at:      01007000 0115D600
relocated to:  00110000 00266600
zimage at:     00115400 0017D54A
initrd at:     00182000 00266600
avail ram:     00267000 00367000

Linux/PPC load: ramdisk_size=28472 hda=bswap hdb=bswap hdc=bswap
hdd=bswap root=101
Uncompressing Linux...done.
Now booting the kernel
r4 = 0x182000 r5 = 0x266600
Total memory = 512MB; using 2048kB for hash table (at c0400000)
Linux version 2.4.10-1 (root@build1.tst) (gcc version 2.95.3
20010315 (release)) #10 Mon May 20 11:12:15 MSD 2002
pciauto_bus_scan: current_bus = 0
pciauto_bus_scan: first_busno = 0
PCI Autoconfig: Found device 10 function 0
PCI Autoconfig: Found Bus 0, Device 10, Function 0
PCI Autoconfig: BAR 0x10, Mem size=0x1000, address=0x8ffff000
PCI Autoconfig: BAR 0x14, I/O, size=0x40, address=0xffffc0
PCI Autoconfig: BAR 0x18, Mem size=0x20000, address=0x8ffc0000
PCI Autoconfig: Found device 14 function 0
PCI Autoconfig: Found P2P bridge, device 14
PCI Autoconfig: Found Bus 0, Device 14, Function 0
pciauto_bus_scan: current_bus = 1
pciauto_bus_scan: first_busno = 0
Marvell/Galileo EV-64260-BP Evaluation Board
EV-64260-BP port (C) 2001 MontaVista Software, Inc.
(source@mvista.com)
On node 0 totalpages: 131072
zone(0): 131072 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: ramdisk_size=28472 hda=bswap hdb=bswap
hdc=bswap hdd=bswap root=101
time_init: decremter frequency = 33.250000 MHz
Warning: real time clock seems stuck!
Calibrating delay loop... 1196.03 BogomIPS
Memory: 510872k available (780k kernel code, 372k data, 56k init,
0k highmem)
Dentry-cache hash table entries: 65536 (order: 7, 524288 bytes)
Inode-cache hash table entries: 32768 (order: 6, 262144 bytes)
Mount-cache hash table entries: 8192 (order: 4, 65536 bytes)
Buffer-cache hash table entries: 32768 (order: 5, 131072 bytes)
Page-cache hash table entries: 131072 (order: 7, 524288 bytes)
POSIX conformance testing by UNIFIX
PCI: Probing PCI hardware
PCI:00:00.0 Resource 6 [ff000000-ffffff] is unassigned
ev64260_map_irq: idsel = 0 pin = 1
IRQ = -1
ev64260_map_irq: idsel = 10 pin = 1
IRQ = 74
ev64260_map_irq: idsel = 14 pin = 1
IRQ = 74
PCI:00:00.0: Resource 4: 14000000-1400ffff (f=200)
PCI: Cannot allocate resource region 4 of device 00:00.0
PCI:00:00.0: Resource 5: 14000000-14000003 (f=101)
PCI: Cannot allocate resource region 5 of device 00:00.0
PCI:00:0a.0: Resource 0: 8ffff000-8fffffff (f=200)
PCI:00:0a.0: Resource 1: 00ffffc0-00fffffff (f=101)

```

```

PCI:00:0a.0: Resource 2: 8ffc0000-8ffdf000 (f=200)
PCI: Failed to allocate resource 0(a0000000-8fffffff) for 00:00.0
PCI: Error while updating region 00:00.0/5 (00001001 != 00000001)
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Starting kswapd v1.8
pty: 256 Unix98 ptys configured
GT64260 MPSC driver version 1.00
block: queued sectors max/low 339469kB/208397kB, 1024 slots per
queue
RAMDISK driver initialized: 16 RAM disks of 28472K size 1024
blocksize
eepro100.c:v1.09j-t 9/29/99 Donald Becker
http://cesdis.gsfc.nasa.gov/linux/drivers/eepro100.html
eepro100.c: $Revision: 1.36 $ 2000/11/17 Modified by Andrey V.
Savochkin <saw@saw.sw.com.sg> and others
eth0: PCI device 8086:1209, 00:20:CE:A0:02:E0, IRQ 74.
Receiver lock-up bug exists -- enabling work-around.
Board assembly d12dc2-009, Physical connectors present: RJ45 BNC
Primary interface chip unknown-14 PHY #3.
General self-test: passed.
Serial sub-system self-test: passed.
Internal registers self-test: passed.
ROM checksum self-test: passed (0x1d68d8db).
Receiver lock-up workaround activated.
In gt64260_eth_module_init()
gt64260_eth_init()
gt64260_eth_init()
gt64260_eth_init()
cards = 3
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 4096 buckets, 32Kbytes
TCP: Hash tables configured (established 131072 bind 65536)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 1544
Freeing BlueCat RFS memory: 913k freed
EXT2-fs warning: checktime reached, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 56k init
init started: BusyBox v0.52pre (2002.05.16-17:20+0000) multi-call
binary

hush -- the humble shell v0.01 (testing)

/ #

```

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
Sat May  6 11:54:46 UTC 2002

```
- `uname`

```

/ # uname -a
Linux (none) 2.4.10-1 #10 Mon May 20 11:12:15 MSD 2002 ppc unknown

```
- `mount`

```

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

```
- `ifconfig`

```

/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:20:CE:A0:02:E0
          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:74
/ # 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=1.2 ms
64 bytes from 172.17.0.1: icmp_seq=1 ttl=255 time=0.1 ms
--- 172.17.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.3/1.2 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 spec file (tinylogin.spec) that contains the following minimal set of 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

mknod /dev/ptyp0 c 2 0
mknod /dev/ptyp1 c 2 1
mknod /dev/ptyp2 c 2 2
mknod /dev/ptyp3 c 2 3
mknod /dev/ptyp4 c 2 4
mknod /dev/ptyp5 c 2 5
mknod /dev/ptyp6 c 2 6
mknod /dev/ptyp7 c 2 7
mknod /dev/ptyp8 c 2 8
mknod /dev/ptyp9 c 2 9

chmod 0666 /dev/ptyp0
chmod 0666 /dev/ptyp1
chmod 0666 /dev/ptyp2
chmod 0666 /dev/ptyp3
chmod 0666 /dev/ptyp4
chmod 0666 /dev/ptyp5
chmod 0666 /dev/ptyp6
chmod 0666 /dev/ptyp7
chmod 0666 /dev/ptyp8
chmod 0666 /dev/ptyp9

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::0:0:/root:::/bin/bash
guest::500:10:::/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 RFS 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 PowerBoot 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 PowerBoot prompt, type the following commands:

```
PowerBoot> netload tinylogin.kdi 1000000 \  
<target_ip> <host_ip>  
  
PowerBoot> go 1007000
```

where *<target_ip>* is an IP address of the target, *<host_ip>* is an IP address of the development 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 current 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. Setup 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 file `zebra.config` 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 spec file (`zebra.spec`) that contains the following minimal set of directives:

```
strip on

mkdir /dev
mknod /dev/console c 5 1
ln -s /dev/console /dev/tty
ln -s /dev/console /dev/ttyl
# 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
```

```

mknod /dev/ptyp0 c 2 0
mknod /dev/ptyp1 c 2 1
mknod /dev/ptyp2 c 2 2
mknod /dev/ptyp3 c 2 3
mknod /dev/ptyp4 c 2 4
mknod /dev/ptyp5 c 2 5
mknod /dev/ptyp6 c 2 6
mknod /dev/ptyp7 c 2 7
mknod /dev/ptyp8 c 2 8
mknod /dev/ptyp9 c 2 9

chmod 0666 /dev/ptyp0
chmod 0666 /dev/ptyp1
chmod 0666 /dev/ptyp2
chmod 0666 /dev/ptyp3
chmod 0666 /dev/ptyp4
chmod 0666 /dev/ptyp5
chmod 0666 /dev/ptyp6
chmod 0666 /dev/ptyp7
chmod 0666 /dev/ptyp8
chmod 0666 /dev/ptyp9

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 of the following contents:

```

!
! zebra configuration file
!
hostname Router
password zebra
enable password zebra

```

```
!
! Interface's description.
!
interface lo
ip address 127.0.0.1/8

interface eth0
ip address 172.17.1.3/16

!
! Static default route.
!
ip route 80.240.0.0 255.255.0.0 172.17.0.1

log stdout
```

NOTE: This configuration file sets the password to zebra. The user has to enter this password when connecting to Zebra or changing the Zebra configuration mode by entering the **enable** command at the command prompt.

8. Copy the `fstab`, `passwd`, `mtab`, `other`, `hosts`, `protocols`, `resolv.conf`, `shadow`, `pam.d/*`, `xinetd.d/*`, and empty files from the `$BLUECAT_PREFIX/demo/developer/local` directory to the `$BLUECAT_PREFIX/demo/zebra/local` directory:
9. Create a root file system image (`zebra.rfs`) by entering the following command:

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

NOTE: Makefile for the `developer` demo system can be used as a starting point to produce the Zebra kernel and RFS images.

Booting the Zebra Images from a Network

The user must step through the following procedure to boot the BlueCat Linux with the Zebra utility from a network using the PowerBoot 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 PowerBoot prompt, type the following commands:

```
PowerBoot> netload zebra.kdi 1000000 <target_ip> \  
<host_ip>  
PowerBoot> go 1007000
```

where `<target_ip>` is an IP address of the target, `<host_ip>` is an IP address of the development host.

The Zebra utility is loaded onto the target board and automatically starts.

Using Zebra Utility

This section provides an examples of using the Zebra utility:

```

loaded at:      01007000 01312600
relocated to:  00120000 0042B600
zimage at:     00125400 0018F548
initrd at:     00194000 0042B600
avail ram:     0042C000 0052C000

Linux/PPC load: ramdisk_size=28472 hda=bswap hdb=bswap hdc=bswap hdd=bswap
root=101
Uncompressing Linux...done.
Now booting the kernel
r4 = 0x194000 r5 = 0x42b600
Total memory = 512MB; using 2048kB for hash table (at c0600000)
Linux version 2.4.10-1 (root@buildd.tst) (gcc version 2.95.3 20010315
(release)) #12 Mon May 20 11:13:31 MSD 2002
pciauto_bus_scan: current_bus = 0
pciauto_bus_scan: first_busno = 0
PCI Autoconfig: Found device 10 function 0
PCI Autoconfig: Found Bus 0, Device 10, Function 0
PCI Autoconfig: BAR 0x10, Mem size=0x1000, address=0x8ffff000
PCI Autoconfig: BAR 0x14, I/O, size=0x40, address=0xffffc0
PCI Autoconfig: BAR 0x18, Mem size=0x20000, address=0x8ffc0000
PCI Autoconfig: Found device 14 function 0
PCI Autoconfig: Found P2P bridge, device 14
PCI Autoconfig: Found Bus 0, Device 14, Function 0
pciauto_bus_scan: current_bus = 1
pciauto_bus_scan: first_busno = 0
Marvell/Galileo EV-64260-BP Evaluation Board
EV-64260-BP port (C) 2001 MontaVista Software, Inc. (source@mvista.com)
On node 0 totalpages: 131072
zone(0): 131072 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: ramdisk_size=28472 hda=bswap hdb=bswap hdc=bswap
hdd=bswap
root=101
time_init: decremter frequency = 33.250000 MHz
Warning: real time clock seems stuck!
Calibrating delay loop... 1196.03 BogoMIPS
Memory: 509108k available (800k kernel code, 376k data, 56k init, 0k
highmem)
Dentry-cache hash table entries: 65536 (order: 7, 524288 bytes)
Inode-cache hash table entries: 32768 (order: 6, 262144 bytes)
Mount-cache hash table entries: 8192 (order: 4, 65536 bytes)
Buffer-cache hash table entries: 32768 (order: 5, 131072 bytes)
Page-cache hash table entries: 131072 (order: 7, 524288 bytes)
POSIX conformance testing by UNIFIX
PCI: Probing PCI hardware
PCI:00:00.0 Resource 6 [ff000000-ffffff] is unassigned
ev64260_map_irq: idsel = 0 pin = 1
IRQ = -1
ev64260_map_irq: idsel = 10 pin = 1
IRQ = 74
ev64260_map_irq: idsel = 14 pin = 1
IRQ = 74

```

```

PCI:00:00.0: Resource 4: 14000000-1400ffff (f=200)
PCI: Cannot allocate resource region 4 of device 00:00.0
PCI:00:00.0: Resource 5: 14000000-14000003 (f=101)
PCI: Cannot allocate resource region 5 of device 00:00.0
PCI:00:0a.0: Resource 0: 8ffff000-8fffffff (f=200)
PCI:00:0a.0: Resource 1: 00ffffc0-00fffffff (f=101)
PCI:00:0a.0: Resource 2: 8ffc0000-8ffdffff (f=200)
PCI: Failed to allocate resource 0(a0000000-8fffffff) for 00:00.0
PCI: Error while updating region 00:00.0/5 (00001001 != 00000001)
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd v1.8
pty: 256 Unix98 ptys configured
GT64260 MPSC driver version 1.00
block: queued sectors max/low 338290kB/207218kB, 1024 slots per queue
RAMDISK driver initialized: 16 RAM disks of 28472K size 1024 blocksize
eepro100.c:v1.09j-t 9/29/99 Donald Becker
http://cesdis.gsfc.nasa.gov/linux/drivers/eepro100.html
eepro100.c: $Revision: 1.36 $ 2000/11/17 Modified by Andrey V. Savochkin
<saw@saw.sw.com.sg> and others
eth0: PCI device 8086:1209, 00:20:CE:A0:02:E0, IRQ 74.
Receiver lock-up bug exists -- enabling work-around.
Board assembly d12dc2-009, Physical connectors present: RJ45 BNC
Primary interface chip unknown-14 PHY #3.
General self-test: passed.
Serial sub-system self-test: passed.
Internal registers self-test: passed.
ROM checksum self-test: passed (0xld68d8db).
Receiver lock-up workaround activated.
In gt64260_eth_module_init()
gt64260_eth_init()
gt64260_eth_init()
gt64260_eth_init()
cards = 3
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 4096 buckets, 32Kbytes
TCP: Hash tables configured (established 131072 bind 65536)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 1616
Freeing BlueCat RFS memory: 2653k freed
EXT2-fs warning: checktime reached, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 56k init
INIT: version 2.78 booting
INIT: Entering runlevel: 1

myhostname login: root
bash-2.04# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:20:CE:A0:02:E0
          inet addr:172.17.1.3  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:74

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 192.168.4.122 : 56(84) bytes of data.
64 bytes from 172.17.0.1: icmp_seq=0 ttl=255 time=1.126 msec
64 bytes from 172.17.0.1: icmp_seq=1 ttl=255 time=160 usec

--- 172.17.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.160/0.643/1.126/0.483 ms
bash-2.04# ping -c 3 80.240.102.3
PING 80.240.102.3 (80.240.102.3) from 172.17.1.3 : 56(84) bytes of data.
64 bytes from 80.240.102.3: icmp_seq=0 ttl=62 time=3.070 msec
64 bytes from 80.240.102.3: icmp_seq=1 ttl=63 time=664 usec
64 bytes from 80.240.102.3: icmp_seq=2 ttl=63 time=682 usec

--- 80.240.102.3 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.664/1.472/3.070/1.129 ms
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>* 80.240.0.0/16 [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 ppmc260 BSP:

Table 5-1: Device Drivers Supported by the ppmc260 BSP

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
UART Dual GT64260	gt64260_mpsc.c gt64260_mpsc.h	drivers/char	CONFIG_GT64260_MPSC CONFIG_GT64260_MPSC_0 CONFIG_GT64260_MPSC_1 CONFIG_GT64260_CONSOLE	
Ethernet Controller Three GT64260	gt64260_eth.c gt64260_eth.h	drivers/net	CONFIG_GT64260_ETH CONFIG_GT64260_ETH_0 CONFIG_GT64260_ETH_0_MACADDR CONFIG_GT64260_ETH_1 CONFIG_GT64260_ETH_1_MACADDR CONFIG_GT64260_ETH_2 CONFIG_GT64260_ETH_2_MACADDR	
NOR Flash	ppmc260.c	drivers/mtd/maps	CONFIG_MTD_PPMC260 CONFIG_MTD_PPMC260_PART	
Ethernet Controller Intel i82559 compatible	eepr0100.c	drivers/net	CONFIG_EEPR0100	
Ethernet Controller Digital DEC21143	*.c *.h	drivers/net/tulip	CONFIG_TULIP CONFIG_TULIP_MWI CONFIG_TULIP_MMIO	

Defect Fixes and Known Limitations

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

Table 6-1: Defect Fixes in BlueCat Linux

Platform	Subcomponent	ID	Summary
All	BlueCat Misc	16057	Ctrl-C, Ctrl-Z etc. do not work from shell
Windows (cross)	BlueCat Installation procedure	16130	Windows install should support non-C drives
All	BlueCat OS Loader	16358	The BLOSH <code>ntar</code> command hangs the system
All	BlueCat Misc	17308	Certain sequence of file updates sometimes causes FFS to crash after reboot
All	BlueCat Debuggers	18404	Update BlueCat <code>gdbserver</code> to process signals to <code>gdb</code> on the host correctly
All	BlueCat Linux Misc	18696	Provide support for NFS in BlueCat Linux
All	BlueCat mkimage	19605	<code>mkrootfs</code> doesn't support paths > 128 characters in <code>lcd</code> command

Force PPMC260 Target Board Problems and Limitations

- Due to absence of a definition of a hard reset procedure for the board, rebooting of the Force PPMC260 board from the BlueCat Linux kernel is not supported in this release.
- Due to absence of a battery on the available test board, RTC support is not tested in this release.
- Support for the DEC21143 PMC Ethernet controller is not tested.

- BlueCat Linux for Force PPMC260 does not autodetect the amount of RAM on the board. The BlueCat Linux kernel uses a default 512MB of RAM.

If the PPMC260 board uses a different amount of memory, the BlueCat Linux kernel must be recompiled with the appropriate RAM amount. The following kernel command line option must be used:

```
mem=<memory>M
```

where *<memory>* is the amount of memory on the board.

This command can be edited in the kernel Makefile or set manually from the BLOSH prompt. For example, from BLOSH:

```
> s CMD mem=256M
```

- Installing of a BlueCat Linux embedded system image to flash from the PowerBoot firmware is not tested.
- 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.

- Rebuilding the BlueCat kernel with `make xconfig` to add character virtual terminal support fails. The default BlueCat Linux 2.4.10 kernel configuration for the PowerPC does not include the option to support graphical terminals. This option requires a PC-style graphical console device, which is not included on PPMC260 boards.

However, to rebuild the kernel with support for graphical terminals, use the following workaround:

Edit the file:

```
BLUECAT_PREFIX/usr/src/linux/include/asm/keyboard.h
```

add the following line before the `#include <asm/io.h>` line:

```
#include <linux/pm.h>
```

