BlueCat Linux Board Support Guide

Release 4.0

DOC-0560-00

for the Intel DBPXA250 Board

Product names mentioned in the *BlueCat Linux Board Support Guide for the Intel DBPXA250 Board* are trademarks of their respective manufacturers and are used here only for identification purposes.

Printed in the United States of America.

Contents

Typographical Conventions v
Special Notes vi
Technical Support vi

CHAPTER 1 OVERVIEW 1

CHAPTER 2 DOWNLOADING AND BOOTING BLUECAT LINUX ON THE TARGET 3
Prerequisites 3
Downloading and Booting Overview 4
Setting up Hardware 4
Connecting Target Board Serial Ports to Host 4
Connecting Target Board Ethernet Card to Host 5
Installing Boot Loader into Flash on the DBPXA250 5
Setting up the LynuxWorks Boot Loader Firmware 7
Downloading a BlueCat Linux System to Flash 8
Downloading a BlueCat Linux System to Flash Using Boot Loader 9
Booting a Demo System from Flash 10
Booting a Demo System from a Network 11

CHAPTER 3 KERNEL CONFIGURATION OPTIONS 13

CHAPTER 4 SUPPORTED DEMO SYSTEMS 37
Demo Systems 37
developer Demo System 38
osloader Demo System 38
showcase Demo System 38
Using Selected RPM Packages 38

Using BusyBox RPM Package 39
Using TinyLogin RPM Package 43
Using Zebra RPM Package 47

CHAPTER 5 SUPPORTED DEVICE DRIVERS 55

CHAPTER 6 KNOWN LIMITATIONS 57
DBPXA250 Target Board Problems and Limitations 57

Preface

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; *italicized* for emphasis, new terms, and book titles	Refer to the *LynxOS User's Guide.*

Environment variables, file names, functions, methods, options, parameter names, path names, commands, and computer data
Commands that need to be highlighted within body text, or commands that must be typed as is by the user are **`bolded`**.
	`ls`
	`-l`
	`myprog.c`
	`/dev/null`
	`login: myname`
	# **`cd /usr/home`**

Text that represents a variable, such as a file name or a value that must be entered by the user	`cat <filename>` `mv <file1> <file2>`

Blocks of text that appear on the display screen after entering instructions or commands	`Loading file /tftpboot/shell.kdi into 0x4000`
	`.....`
	`File loaded. Size is 1314816`
	`Copyright 2002 LynuxWorks, Inc. All rights reserved.`
	
	`LynxOS (ppc) created Mon Jan 17 17:50:22 GMT 2002`
	`user name:`

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.lynuxworks.com`

customer support `http://www.lynuxworks.com/support/custhelp.php3`

CHAPTER 1 *Overview*

The *BlueCat Linux Board Support Guide for the Intel DBPXA250 Board* provides information about the BlueCat Linux Board Support Package (BSP) for the Intel DBPXA250 board. The Intel DBPXA250 board is a development platform that features the PXA250 processor on the plug-in processor card installed onto the BBPXA2xx I/O baseboard.

Throughout this Board Support Guide (BSG), the BSP is referred to as the "dbpxa250" and the board as the "DBPXA250", or simply as the "target board".

This BSG is organized as follows:

- *Chapter 1* is an overview of the individual chapters.
- *Chapter 2* describes BlueCat Linux downloading and booting procedures for the DBPXA250 board using the BlueCat Linux `developer` demo system as an example.
- *Chapter 3* provides configuration option information about the default kernel for the `dbpxa250` BSP.
- *Chapter 4* summarizes BlueCat Linux demo systems supported by the `dbpxa250` BSP.
- *Chapter 5* lists the device drivers supported by the `dbpxa250` BSP.
- *Chapter 6* describes the defect fixes included in this release, as well as known limitations and workarounds.

CHAPTER 2 *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. As such, a basic familiarity with the target board hardware and operation is required before using this guide. The user must also have an understanding of system administration for the particular cross-development host on which BlueCat Linux Xscale binary architecture and the 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 XScale configuration and the `dbpxa250` BSP have been installed on the cross-development host. This means that the user must:

1. Install the BlueCat Linux XScale 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 `dbpxa250` BSP onto the cross-development host as detailed in the section "Installing Target Board Support" in Chapter 1, "Installation" in the *BlueCat Linux User's Guide*.
3. Activate support for the `dbpxa250` BSP as detailed in the "Activating Support for a Target Board" section of Chapter 1, "Installation" in the *BlueCat Linux User's Guide*.

Downloading and Booting Overview

The procedure for downloading and booting a BlueCat Linux system on the DBPXA250 target consists of the following main steps:

- Setting up hardware
- Downloading and booting a BlueCat Linux system from target Flash memory or a network

Downloading and booting a BlueCat Linux system can be performed using the LynuxWorks Boot Loader. The LynuxWorks Boot Loader is firmware intended to load an image of BlueCat Linux on the target board. Boot Loader is able to load BlueCat Linux embedded systems from Flash or over a network and to program them to Flash.

Setting up Hardware

Connecting Target Board Serial Ports to Host

The target board has two serial ports. The first port is used both by the LynuxWorks Boot Loader firmware and the BlueCat Linux system console.

Before using the board, at least the first serial port needs to be connected to the development host. It is recommended to connect the target serial connector to COM1 on the host.

The serial port settings on the host must be as follows:

- The serial port connected to the target first serial port has a baud rate of 115200
- The serial port connected to the second target port has a baud rate of 115200

Throughout this chapter, the terminal window connected to the first serial connector is referred to as the *Boot Loader console* or *BlueCat Linux console*, depending on the context.

Connecting Target Board Ethernet Card to Host

The Ethernet port on the target board is used to provide a standard network connection for the board, and, in particular, to load BlueCat Linux embedded systems onto the board over a network.

There is a single Ethernet port on the target board. The user must use this port to connect the target board to the LAN.

It also required to set up networking on the host system. In particular, the user must choose a unique IP address for the development host as well as for the target board.

These addresses are referred to as *`development_host_IP`* and *`target_board_IP,`* respectively. For more information on how to set up networking on the host, please refer to the host operating system documentation.

TFTP must be enabled on the host. Refer to “Setting Up a TFTP Server” in Chapter 3 of the *BlueCat Linux User’s Guide* for more information.

Installing Boot Loader into Flash on the DBPXA250

To install Boot Loader on the DBPXA250 board, the on-board Flash chip must be programmed with the Boot Loader image (`$BLUECAT_PREFIX/boot/dbpxa250.bin`).

This can be done via JTAG using the JFlash utility. Perform the following steps to install Boot Loader using JFlash.

1. Download the JFlash Kit for DBPXA2xx platforms from the following URL:

   `http://www.intel.com/design/pca/applicationsprocessors/\`

   `swsup/downloads/JFlash_DBPXA2xx_V2_09_001_Kit.htm`

2. Run `JFlash_DBPXA2xx_V2_09_001_Kit.exe` to install the JFlash Kit onto a Windows 98, 2000, or NT host.
3. Turn off the power to the host and the DBPXA250 board.
4. Connect the DBPXA250 board to the host using the JTAG cable which comes with the Intel DBPXA250 development platform kit following the instructions from the `RelNote_JFlash_DBPXA2xx.htm` file located in the JFlash Kit installation directory.

5. Select which Flash memory bank will be programmed with the Boot Loader image.
   - To program Boot Loader into the boot Flash memory (recommended), set the S15 switch to the no-dot position and the S14 switch to the dot position.
   - To program Boot Loader into the application Flash memory, set the S15 switch to the dot position.
6. Turn on the power to the host and the DBPXA250 board.
7. For Windows 2000 or Windows NT, install the `giveio.sys` driver. To install the driver, do the following:
   - Log in as Administrator.
   - Open a DOS window and go to the JFlash Kit installation directory.
   - Copy the `giveio.sys` driver to `\winnt\system32\drivers`.

```
C:\Program Files\Intel XScale Software\JFlash>copy \
giveio.sys c:\winnt\system32\drivers\
        1 file(s) copied.
```

   - Install the driver using the `instdrv.exe` utility.

```
C:\Program Files\Intel XScale Software\JFlash>instdrv.exe \
giveio c:\winnt\system32\drivers\giveio.sys
    CreateService SUCCESS
    StartService SUCCESS
    CreateFile SUCCESS
```

8. Run the JFlash utility and pass to it the Boot Loader image file name as a single argument.

```
C:\Program Files\Intel XScale Software\JFlash>
JFlash_DBPXA250_V2_09_001.exe c:\Temp\dbpxa250.bin
JFLASH Version 2.09.001 - DBPXA250
COPYRIGHT (C) 2000, 2001 Intel Corporation
JTAG Test Passed
PXA250 revision B1

There are two 16-bit Flash devices in parallel

Characteristics for one device:
 Number of blocks in device = 128
 Block size = 65536 0x10000 word(16-bit)
 Device size = 8388608 0x800000 word(16-bit)

Sample block to address list:

 Block 0 = hex address: 00000000
 Block 40 = hex address: 00A00000
 Block 80 = hex address: 01400000
 Block 120 = hex address: 01E00000
The last 42 percent of image file is all zeros
Would you like to save time by not programming that area? [y/n]:
```

```
  Confirm programming the whole image by answering 'n'.

Would you like to save time by not programming that area? [y/n]: n
Starting erase
Erasing done
Starting programming
Programming done hex address    7f480, 99.44% done
Starting Verify
Verification successful!

C:\Program Files\Intel XScale Software\JFlash>
```

9. Set the S17/18 switches to the no-dot position.
10. Reset the DBPXA250 board.

    The Boot Loader prompt will now appear at the first DBPXA250 serial console.

Setting up the LynuxWorks Boot Loader Firmware

Use the following procedure to set up the LynuxWorks Boot Loader firmware options for BlueCat Linux operations:

1. Reset the target board.
2. At the Boot Loader console, enter the following commands:

   ```
   dbpxa250> set autoboot 0

   dbpxa250> set boot_tftp_host_ip \
   <development_host_IP>

   dbpxa250> set boot_tftp_client_ip <target_board_IP>

   dbpxa250> set flash_tftp_host_ip \
   <development_host_IP>

   dbpxa250> set flash_tftp_client_ip
   <target_board_IP>

   dbpxa250> save
   ```

   where *<target_board_IP>* is the IP address of the target and *<development_host_IP>* is the IP address of the development host.
3. Reset the target board.

Downloading a BlueCat Linux System to Flash

This section provides instructions on how a BlueCat embedded system can be downloaded into the target Flash memory using the Boot Loader firmware and the BlueCat Linux OS loader. Refer also to the *BlueCat Linux User's Guide* for additional details about the BlueCat Linux OS loader.

Specifically, these instructions are applicable to any of the demo systems. This chapter uses the `developer` demo system as an example.

The DBPXA250 board has two Flash memory banks:

- ROM bank (or boot Flash)
- FLASH bank (or application Flash)

Boot Loader and BlueCat Linux support both these banks. By default, the code located at the beginning of the boot Flash is started after system reset. It is possible, however, to swap Flash banks on DBPXA250. To do that, set the S15 switch to the dot position.

NOTE: Boot Loader occupies the first 1Mb of the boot Flash if the S15 switch is set to the no-dot position or the first 1Mb of the application Flash, otherwise. The Flash memory occupied by Boot Loader cannot be used for programming BlueCat Linux demo systems. Also, note that the boot Flash memory is write-protected if the S14 switch is in the no-dot position. To remove boot Flash write-protection, set the S14 switch to the dot position.

Depending on the S15 switch setting, Boot Loader and BlueCat Linux set one of the following device mappings.

Table 2-1: S15 is in the No-Dot Position (Recommended)

Flash Bank	Boot Loader Device Name	BlueCat Linux Device Name	Available Flash Memory
Boot Flash	`flash0`	`/dev/mtdchar0`	0x0100000 - 0x1ffffff
Application Flash	`flash1`	`/dev/mtdchar1`	0x0000000 - 0x1ffffff

Table 2-2: S15 is in the Dot Position

Flash Bank	Boot Loader Device Name	BlueCat Linux Device Name	Available Flash Memory
Application Flash	`flash0`	`/dev/mtdchar0`	0x0100000 - 0x1ffffff
Boot Flash	`flash1`	`/dev/mtdchar1`	0x0000000 - 0x1ffffff

Throughout this chapter, the `flash0` and `/dev/mtdchar0` devices will be referred to as the primary Flash, while the `flash1` and `/dev/mtdchar1` devices will be referred to as the secondary Flash.

For demonstration purposes, the examples below use the first Flash bank only. If it is required to program BlueCat images into the secondary Flash, `flash0` must be replaced with `flash1` and `/dev/mtdchar0` with `/dev/mtdchar1`, respectively.

Downloading a BlueCat Linux System to Flash Using Boot Loader

Use the following procedure to download `developer` into the primary Flash of the target board using the LynuxWorks Boot Loader:

1. Copy the `developer.kdi` file from the `$BLUECAT_PREFIX/demo/developer` directory to the `/tftpboot` directory on the development host.
2. Reset the target board.
3. At the Boot Loader console, enter the following commands:

```
dbpxa250> set flash_device tftp
dbpxa250> set flash_tftp_file developer.kdi
dbpxa250> set flash_target flash0
dbpxa250> set flash_offset 0x100000
dbpxa250> flash
```

After these commands have been performed, the `developer` demo system is programmed into the primary Flash and can be booted as described in "Booting a Demo System from Flash" below.

Booting a Demo System from Flash

Use the following procedure to boot a demo installed into the primary Flash memory. For detailed information on how to install the demo system to Flash, refer to “Downloading a BlueCat Linux System to Flash” on page 8.

1. Reset the target board.
2. At the Boot Loader console, type the following:

```
dbpxa250> set boot_device flash0
dbpxa250> set boot_flash_offset 0x100000
dbpxa250> set boot_os BlueCat
dbpxa250> boot
```

These commands will start the demo system programmed into the primary Flash at offset 0x100000.

The DBPXA250 board can be configured to start a demo system programmed into Flash automatically at the board power-up. Use the following commands to prepare the DBPXA250 board to boot BlueCat Linux from primary Flash automatically:

```
dbpxa250> set boot_device flash0
dbpxa250> set boot_flash_offset 0x100000
dbpxa250> set boot_os BlueCat
dbpxa250> set autoboot 1
dbpxa250> save
```

The demo system programmed into primary Flash will be started by the Boot Loader monitor automatically on board power-up.

Booting a Demo System from a Network

Use the following procedure to boot the `developer` demo system over a network using the Boot Loader firmware.

1. Copy the `developer.kdi` file from the `$BLUECAT_PREFIX/demo/developer` directory to the `/tftpboot` directory on the cross-development host.
2. Reset the target board.
3. At the Boot Loader console, enter the following commands:

```
dbpxa250> set boot_device tftp
dbpxa250> set boot_tftp_file developer.kdi
dbpxa250> boot
```

These commands load the `developer` demo system from a network onto the target board and then automatically start it.

CHAPTER 3 *Kernel Configuration Options*

The dbpxa250 BSP comes with a default BlueCat Linux kernel. This kernel has a number of configuration options. This chapter details these options in the tables listed in Table 3-1: "BlueCat Linux Default Configuration for the dbpxa250 BSP Distribution" below.

Table 3-1: BlueCat Linux Default Configuration for the dbpxa250 BSP Distribution

Table Number and Configuration Parameter
Table 3-2: "Code Maturity Level Options"
Table 3-3: "Loadable Module Support"
Table 3-4: "System Type"
Table 3-5: "SA11x0 Implementations"
Table 3-6: "General Setup"
Table 3-7: "Parallel Port Support"
Table 3-8: "Memory Technology Devices (MTD)"
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: "NAND Flash Device Driver"
Table 3-13: "Plug and Play Configuration"
Table 3-14: "Block Devices"
Table 3-15: "Multi-device Support (RAID and LVM)"
Table 3-16: "Networking Options"
Table 3-17: "QoS and/or Fair Queueing"

Table 3-1: BlueCat Linux Default Configuration for the dbpxa250 BSP Distribution (Continued)

Table Number and Configuration Parameter
Table 3-18: "Network Device Support"
Table 3-19: "Arcnet Devices"
Table 3-20: "Ethernet (10 or 100Mbit)"
Table 3-21: "Ethernet (1000 Mbit)"
Table 3-22: "Wireless LAN (Non-hamradio)"
Table 3-23: "Token Ring Devices"
Table 3-24: "WAN Interfaces"
Table 3-25: "Amateur Radio Support"
Table 3-26: "IrDA (Infrared) Support"
Table 3-27: "ATA/IDE/MFM/RLL Support"
Table 3-28: "SCSI Support"
Table 3-29: "I2O Device Support"
Table 3-30: "ISDN Subsystem"
Table 3-31: "Input Core Support"
Table 3-32: "Character Devices"
Table 3-33: "Serial Drivers"
Table 3-34: "I2C Support"
Table 3-35: "L3 Serial Bus Support"
Table 3-36: "Mice"
Table 3-37: "Joysticks"
Table 3-38: "Watchdog Cards"
Table 3-39: "Ftape, the Floppy Tape Device Driver"
Table 3-40: "Multimedia Devices"
Table 3-41: "File System"
Table 3-42: "Network File Systems"
Table 3-43: "Partition Types"

Table 3-1: BlueCat Linux Default Configuration for the dbpxa250 BSP Distribution (Continued)

Table Number and Configuration Parameter
Table 3-44: "Console Drivers"
Table 3-45: "Frame-Buffer Support"
Table 3-46: "Multimedia Capabilities Port Drivers"
Table 3-47: "USB Support"
Table 3-48: "USB Serial Converter Support"
Table 3-49: "Bluetooth Support"
Table 3-50: "Kernel hacking"
Table 3-51: "Modular Advanced Power Management"
Table 3-52: "Messenger Support"

Table 3-2: Code Maturity Level Options

Option	Value	Description
CONFIG_EXPERIMENTAL	Y	Prompt for Development and/or Incomplete Code/Drivers
CONFIG_OBSOLETE	N	Prompt for Obsolete 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: System Type

Option	Value	Description
CONFIG_ARCH_DBPXA250	Y	Arm System Type

Table 3-5: SA11x0 Implementations

Option	Value	Description
CONFIG_ASSABET_NEPONSET	N	Include Support for Neponset

Table 3-6: General Setup

Option	Value	Description
CONFIG_ANGELBOOT	N	Load Kernel Using Angel Debug Monitor
CONFIG_BLUECAT_IGNORE_PRINTK	N	BlueCat Ignore printk
CONFIG_BLUECAT_THUMB	N	Bluecat Kernel Support For THUMB Binaries
CONFIG_BLUECAT_LOADER	N	BlueCat OS Loader
CONFIG_BLUECAT_SMALL_FOOTPRINT	N	BlueCat Small Memory Footprint
CONFIG_HOTPLUG	N	Support Hot-pluggable Devices
CONFIG_NET	Y	Networking Support
CONFIG_BLUECAT_MEMSIZE	N	Memory Sizing Benchmarks
CONFIG_SYSVIPC	N	System V IPC
CONFIG_BSD_PROCESS_ACCT	N	BSD Process Accounting
CONFIG_SYSCTL	N	Sysctl Support
CONFIG_FPE_NWFPE	Y	NWFPE Math Emulation
CONFIG_FPE_FASTFPE	N	FastFPE Math Emulation (Experimental)

Table 3-6: General Setup (Continued)

Option	Value	Description
CONFIG_KCORE_ELF	Y	Kernel Core (/proc/kcore) Format
CONFIG_BINFMT_AOUT	N	Kernel Support for a.out Binaries
CONFIG_BINFMT_ELF	Y	Kernel Support for ELF Binaries
CONFIG_BINFMT_MISC	N	Kernel Support for MISC Binaries
CONFIG_PM	N	Power Management Support (Experimental)
CONFIG_ARTHUR	N	RISC OS Personality
CONFIG_ALIGNMENT_TRAP	Y	Kernel-Mode Alignment Trap Handler

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
CONFIG_MTD_REDBOOT_PARTS	N	RedBoot Partition Table Parsing
CONFIG_MTD_BOOTLDR_PARTS	N	Compaq bootldr Partition Table Parsing
CONFIG_MTD_AFS_PARTS	N	ARM Firmware Suite Partition Parsing
CONFIG_MTD_CHAR	Y	Direct Char Device Access to MTD Devices

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

Option	Value	Description
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_JEDECPROBE	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_NORA	N	CFI Flash Device Mapped on Nora

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

Option	Value	Description
CONFIG_MTD_ARM_INTEGRATOR	N	CFI Flash Device Mapped on ARM Integrator/P720T
CONFIG_MTD_CDB89712	N	Cirrus CDB89712 Evaluation Board Mappings
CONFIG_MTD_DBPXA250_1	Y	Primary Flash Chip Mapping on the DBPXA250 Board
CONFIG_MTD_DBPXA250_1_PART	:	Partitions layout
CONFIG_MTD_DBPXA250_2	Y	Secondary Flash Chip Mapping on the DBPXA250 Board
CONFIG_MTD_DBPXA250_2_PART	:	Partitions Layout

Table 3-11: Self-contained MTD Device Drivers

Option	Value	Description
CONFIG_MTD_SLRAM	N	Uncached System RAM
CONFIG_MTD_MTDRAM	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 Driver

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_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	28472	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)
CONFIG_MD_LINEAR	N	Linear (append) Mode
CONFIG_MD_RAID0	N	RAID-0 (striping) Mode
CONFIG_MD_RAID1	N	RAID-1 (mirroring) Mode
CONFIG_MD_RAID5	N	RAID-4/RAID-5 Mode
CONFIG_MD_MULTIPATH	N	Multipath I/O Support

Table 3-16: Networking Options

Option	Value	Description
CONFIG_PACKET	N	Packet Socket
CONFIG_NETLINK	N	Kernel/User Netlink Socket
CONFIG_NETFILTER	N	Network Packet Filtering (replaces ipchains)
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	N	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 (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	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)

Table 3-16: Networking Options (Continued)

Option	Value	Description
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: QoS and/or Fair Queueing

Option	Value	Description
CONFIG_NET_SCHED	N	QoS and/or Fair Queueing

Table 3-18: Network Device Support

Option	Value	Description
CONFIG_NETDEVICES	Y	Network Device Support?

Table 3-19: Arcnet Devices

Option	Value	Description
CONFIG_ARCNET	N	ARCnet 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

Table 3-20: Ethernet (10 or 100Mbit)

Option	Value	Description
CONFIG_NET_ETHERNET	Y	Ethernet (10 or 100Mbit)
CONFIG_NET_VENDOR_3COM	N	3COM Cards
CONFIG_NET_VENDOR_SMC	Y	Western Digital/SMC Cards
CONFIG_SMC9194	Y	SMC 9194 Support
CONFIG_NET_VENDOR_RACAL	N	Racal-Interlan (Micom) NI Cards
CONFIG_NET_POCKET	N	Pocket and Portable Adapters

Table 3-21: Ethernet (1000 Mbit)

Option	Value	Description
CONFIG_ACENIC_OMIT_TIGON_I	N	Omit Support for Old Tigon I Based AceNICs
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

Table 3-22: Wireless LAN (Non-hamradio)

Option	Value	Description
CONFIG_NET_RADIO	N	Wireless LAN (Non-Ham Radio)

Table 3-23: Token Ring Devices

Option	Value	Description
CONFIG_NET_FC	N	Fibre Channel Driver Support
CONFIG_SHAPER	N	Traffic Shaper (Experimental)

Table 3-24: WAN Interfaces

Option	Value	Description
CONFIG_WAN	N	WAN Interfaces Support

Table 3-25: Amateur Radio Support

Option	Value	Description
CONFIG_HAMRADIO	N	Amateur Radio Support

Table 3-26: IrDA (Infrared) Support

Option	Value	Description
CONFIG_IRDA	N	IrDA Subsystem Support

Table 3-27: ATA/IDE/MFM/RLL Support

Option	Value	Description
CONFIG_IDE	N	ATA/IDE/MFM/RLL Support

Table 3-28: SCSI Support

Option	Value	Description
CONFIG_SCSI	N	SCSI Support

Table 3-29: I2O Device Support

Option	Value	Description
CONFIG_I2O	N	I2O Support

Table 3-30: ISDN Subsystem

Option	Value	Description
CONFIG_ISDN	N	ISDN Support

Table 3-31: Input Core Support

Option	Value	Description
CONFIG_INPUT	N	Input Core Support
CONFIG_INPUT_MOUSEDEV_SCREEN_X	1024	Horizontal Screen Resolution
CONFIG_INPUT_MOUSEDEV_SCREEN_Y	768	Vertical Screen Resolution

Table 3-32: Character Devices

Option	Value	Description
CONFIG_VT	Y	Virtual Terminal
CONFIG_VT_CONSOLE	Y	Support for Console on Virtual Terminal

Table 3-32: Character Devices (Continued)

Option	Value	Description
CONFIG_SERIAL	Y	Standard/Generic (8250/16550 and compatible UARTs) Serial Support
CONFIG_SERIAL_CONSOLE	Y	Support for Console On Serial Port
CONFIG_SERIAL_EXTENDED	N	Extended Dumb Serial Driver Options
CONFIG_SERIAL_NONSTANDARD	N	Nonstandard Serial Port Support

Table 3-33: Serial Drivers

Option	Value	Description
CONFIG_SERIAL_ANAKIN_CONSOLE	N	Console on Anakin Serial Port
CONFIG_SERIAL_AMBA_CONSOLE	N	Support for Console on AMBA Serial Port
CONFIG_SERIAL_CLPS711X_CONSOLE	N	Support for Console on CLPS711X Serial Port
CONFIG_SERIAL_21285_CONSOLE	N	Console on DC21285 Serial Port
CONFIG_SERIAL_UART00_CONSOLE	N	Support for console Excalibur Serial Port
CONFIG_SERIAL_SA1100_CONSOLE	N	Console on Serial Port
CONFIG_SERIAL_8250	N	8250/16550 and Compatible Serial Support (Experimental)
CONFIG_SERIAL_8250_MANY_PORTS	N	Support More Than 4 Serial Ports
CONFIG_SERIAL_8250_SHARE_IRQ	N	Support for Sharing Serial Interrupts
CONFIG_SERIAL_8250_DETECT_IRQ	N	Autodetect IRQ on Standard Ports (Unsafe)
CONFIG_SERIAL_8250_MULTIPORT	N	Support Special Multiport Boards
CONFIG_SERIAL_8250_HUB6	N	Support Bell Technologies HUB6 Card
CONFIG_UNIX98_PTYS	Y	Unix98 PTY Support
CONFIG_UNIX98_PTY_COUNT	32	Maximum Number of Unix98 PTYs In Use (0-2048)

Table 3-34: I2C Support

Option	Value	Description
CONFIG_I2C	N	I2C Support

Table 3-35: L3 Serial Bus Support

Option	Value	Description
CONFIG_L3	N	L3 Support

Table 3-36: Mice

Option	Value	Description
CONFIG_BUSMOUSE	N	Bus Mouse Support
CONFIG_MOUSE	N	Mouse Support (not Serial and Bus Mice)

Table 3-37: Joysticks

Option	Value	Description
CONFIG_QIC02_TAPE	N	QIC-02 Tape Support

Table 3-38: Watchdog Cards

Option	Value	Description
CONFIG_WATCHDOG	N	Watchdog Timer Support
CONFIG_NVRAM	N	/dev/nvram Support
CONFIG_RTC	N	Enhanced Real Time Clock Support

Table 3-38: Watchdog Cards (Continued)

Option	Value	Description
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

Table 3-39: Ftape, the Floppy Tape Device Driver

Option	Value	Description
CONFIG_FTAPE	N	Ftape (QIC-80/Travan) Support
CONFIG_AGP	N	/dev/agpgart (AGP Support)
CONFIG_DRM	N	Direct Rendering Manager (XFree86 DRI Support)
CONFIG_MWAVE	N	ACP Modem (Mwave) Support

Table 3-40: Multimedia Devices

Option	Value	Description
CONFIG_VIDEO_DEV	N	Video for Linux

Table 3-41: File System

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 Version 3)
CONFIG_REISERFS_FS	N	Reiserfs Support

Table 3-41: File System (Continued)

Option	Value	Description
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 Journalling File System Support (Experimental)
CONFIG_FAT_FS	N	DOS FAT File System Support
CONFIG_UMSDOS_FS	N	UNIX-like File System on Top of Standard MSDOS File System
CONFIG_EFS_FS	N	EFS File System Support (read only) (Experimental)
CONFIG_JFFS_FS	Y	Journalling Flash File System (JFFS) Support
CONFIG_JFFS_FS_VERBOSE	0	JFFS Debugging Verbosity (0 = quiet, 3 = noisy)
CONFIG_JFFS_PROC_FS	N	JFFS Stats Available in `/proc` filesystem
CONFIG_JFFS2_FS	N	Journalling Flash File System v2 (JFFS2) Support
CONFIG_CRAMFS	N	Compressed ROM File System Support
CONFIG_TMPFS	N	Virtual Memory File System Support (former shm file system)
CONFIG_RAMFS	N	Simple RAM-based File System Support
CONFIG_ISO9660_FS	N	ISO 9660 CDROM File System Support
CONFIG_MINIX_FS	N	Minix File System Support
CONFIG_FREEVXFS_FS	N	FreeVxFS File System Support (VERITAS VxFS(TM) compatible)
CONFIG_NTFS_FS	N	NTFS File System Support (read only)

Table 3-41: File System (Continued)

Option	Value	Description
CONFIG_HPFS_FS	N	OS/2 HPFS File System Support
CONFIG_PROC_FS	Y	/proc File System Support
CONFIG_DEVFS_FS	N	/dev File System Support (Experimental)
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 File System 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 3-42: 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 fs)
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 Windows shares etc.)
CONFIG_NCP_FS	N	NCP File System Support (to mount NetWare volumes)

Table 3-43: Partition Types

Option	Value	Description
CONFIG_PARTITION_ADVANCED	N	Advanced Partition Selection

Table 3-44: Console Drivers

Option	Value	Description
CONFIG_VGA_CONSOLE	N	VGA Text Console

Table 3-45: Frame-Buffer Support

Option	Value	Description
CONFIG_FB	Y	Support for Frame Buffer Devices (Experimental)
CONFIG_FB_PXA250	Y	PXA250 LCD Support
CONFIG_FB_VIRTUAL	N	Virtual Frame Buffer Support (ONLY FOR TESTING!)
CONFIG_FBCON_ADVANCED	N	Advanced Low Level Driver Options
CONFIG_FBCON_FONTWIDTH8_ONLY	N	Support Only 8 Pixels Wide Fonts
CONFIG_FBCON_FONTS	N	Select Compiled-in Fonts

Table 3-46: Multimedia Capabilities Port Drivers

Option	Value	Description
CONFIG_MCP	Y	Multimedia Drivers
CONFIG_AC97_PXA250	Y	Intel PXA250/210 AC97 Support
CONFIG_MCP_PXA250	Y	Support PXA250 MCP Interface

Table 3-46: Multimedia Capabilities Port Drivers (Continued)

Option	Value	Description
CONFIG_MCP_UCB1200	Y	Support for UCB1200 / UCB1300 / UCB1400
CONFIG_MCP_UCB1200_TS	Y	Touchscreen Interface Support

Table 3-47: USB Support

Option	Value	Description
CONFIG_USB_STORAGE_DEBUG	N	USB Mass Storage Verbose Debug
CONFIG_USB_STORAGE_DATAFAB	N	Datafab MDCFE-B Compact Flash Reader
CONFIG_USB_STORAGE_FREECOM	N	Freecom USB/ATAPI Bridge Support
CONFIG_USB_STORAGE_ISD200	N	ISD-200 USB/ATA Bridge Support
CONFIG_USB_STORAGE_JUMPSHOT	N	Lexar Jumpshot Compact Flash Reader
CONFIG_USB_STORAGE_DPCM	N	Microtech CompactFlash/SmartMedia reader
CONFIG_USB_STORAGE_HP8200e	N	HP CD-Writer 82xx Support
CONFIG_USB_STORAGE_SDDR09	N	SanDisk SDDR-09 (and other SmartMedia) Support

Table 3-48: USB Serial Converter Support

Option	Value	Description
CONFIG_USB_SERIAL_GENERIC	N	USB Generic Serial Driver
CONFIG_USB_SERIAL_BELKIN	N	USB Belkin and Peracom Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_WHITEHEAT	N	USB ConnectTech WhiteHEAT Serial Driver (Experimental)
CONFIG_USB_SERIAL_DIGI_ACCELEPORT	N	USB Digi International AccelePort USB Serial Driver

Table 3-48: USB Serial Converter Support (Continued)

Option	Value	Description
CONFIG_USB_SERIAL_EMPEG	N	USB Empeg empeg-car Mark I/II Driver (Experimental)
CONFIG_USB_SERIAL_FTDI_SIO	N	USB FTDI Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_VISOR	N	USB Handspring Visor /Palm m50x/Sony Clie Driver
CONFIG_USB_SERIAL_IR	N	USB IR Dongle Serial Driver (Experimental)
CONFIG_USB_SERIAL_EDGEPORT	N	USB Inside Out Edgeport Serial Driver (Experimental)
CONFIG_USB_SERIAL_KEYSPAN_PDA	N	USB Keyspan PDA Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_KEYSPAN	N	USB Keyspan USA-xxx Serial Driver (Experimental)
CONFIG_USB_SERIAL_KEYSPAN_USA28	N	USB Keyspan USA-28 Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA28X	N	USB Keyspan USA-28X Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA28XA	N	USB Keyspan USA-28XA Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA28XB	N	USB Keyspan USA-28XB Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA19	N	USB Keyspan USA-19Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA18X	N	USB Keyspan USA-18X Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA19W	N	USB Keyspan USA-19W Firmware
CONFIG_USB_SERIAL_KEYSPAN_USA49W	N	USB Keyspan USA-49W Firmware
CONFIG_USB_SERIAL_MCT_U232	N	USB MCT Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_PL2303	N	USB Prolific 2303 Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_CYBERJACK	N	USB REINER SCT cyberJack pinpad/e-com chipcard reader (Experimental)

Table 3-48: USB Serial Converter Support (Continued)

Option	Value	Description
CONFIG_USB_SERIAL_XIRCOM	N	USB Xircom/ Entregra Single Port Serial Driver (Experimental)
CONFIG_USB_SERIAL_OMNINET	N	USB ZyXEL omni.net LCD Plus Drive (Experimental)

Table 3-49: Bluetooth Support

Option	Value	Description
CONFIG_BLUEZ	N	Bluetooth Subsystem Support

Table 3-50: Kernel hacking

Option	Value	Description
CONFIG_NO_FRAME_POINTER	Y	Compile Kernel Without Frame Pointer
CONFIG_DEBUG_ERRORS	N	Verbose Kernel Error Messages
CONFIG_DEBUG_USER	N	Verbose User Fault Messages
CONFIG_DEBUG_INFO	N	Include Debugging Information In Kernel Binary
CONFIG_DEBUG_SLAB	N	Debug Memory Allocations
CONFIG_MAGIC_SYSRQ	N	Magic SysRq Key
CONFIG_BLUECAT_KDBG	N	Include kdbg Kernel Debugger
CONFIG_DEBUG_SPINLOCK	N	Spinlock Debugging
CONFIG_DEBUG_LL	N	Kernel Low-level Debugging Functions

Table 3-51: Modular Advanced Power Management

Option	Value	Description
CONFIG_BLUECAT_APM	N	MAPM Support

Table 3-52: Messenger Support

Option	Value	Description
CONFIG_BLUECAT_IOPMAN	N	Enable IOP Manager Support
CONFIG_BLUECAT_MSNG	N	Enable Messenger Support

CHAPTER 4 *Supported Demo Systems*

This chapter provides information about BlueCat Linux demo systems supported by the dbpxa250 BSP.

Demo Systems

Table 4-1 lists the demo systems supported in the dbpxa250 BSP distribution, the boot devices supported by each demo system, and their respective RAM and ROM requirements.

Table 4-1: Demo Systems Supported by dbpxa250 BSP

Demo	Boot Devices Supported by Default	ROM Requirements	RAM Requirements
`developer`	Flash (using the LynuxWorks Boot Loader or OS loader) Network (using the LynuxWorks Boot Loader or OS loader)	3550 KB	16384 KB
`osloader`	Flash (using the LynuxWorks Boot Loader or OS loader) Network (using the LynuxWorks Boot Loader or OS loader)	756 KB	5120 KB
`showcase`	Flash (using the LynuxWorks Boot Loader or OS loader) Network (using the LynuxWorks Boot Loader or OS loader)	9505 KB	46080 KB

developer Demo System

The `developer` demo system is a package consisting of the functions 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 component demo systems.

osloader Demo System

`osloader` is the BlueCat OS Loader system 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.

The dbpxa250 BSP makes an enhancement to the standard `showcase` demo system. Namely, a simple Qt/Embedded application automatically started on the LCD and Touch screen panels demonstrates interactive capabilities that are possible on the DBPXA250 board using the dbpxa250 BSP.

The `showcase` demo for the PXA250 utilizes Qtopia, the robust graphics application software from Trolltech. Qtopia is a comprehensive application environment developed for embedded Linux. It runs on mobile phones, personal digital assistants (PDAs), communicators, hybrid devices, and any other device that runs Linux. Qtopia includes personal information management (PIM) applications, business productivity and Internet applications, games, and utilities. Qtopia is included in BlueCat under a noncommercial license (evaluation use only). For licensing terms for Qtopia software, please contact sales@trolltech.com.

Using Selected RPM Packages

This section provides a description on how to use selected RPM packages that are frequently deployed in the embedded systems environment.

Using BusyBox RPM Package

The BusyBox RPM package combines tiny versions of many common UNIX utilities into a single small executable. It provides minimalist replacements for most UNIX utilities, such as `fileutils`, `shellutils`, `findutils`, `textutils`, `grep`, `gzip`, and `tar`. 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 cousins. The options that are included, however, provide the expected functionality and behave very much like their GNU counterparts.

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

Creating a BlueCat Linux System for BusyBox

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

1. Create a new directory by typing:

   ```
   BlueCat:$ mkdir -p \
   $BLUECAT_PREFIX/demo/busybox/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/busybox` directory. For instance, type the following commands:

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

   BlueCat:$ make xconfig
   ```

   Select **`Save and Exit`** to update the `.config` file.

   ```
   BlueCat:$ cp .config \
   $BLUECAT_PREFIX/demo/busybox/busybox.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 (`busybox.kernel`):

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

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

4. Create a spec file (`busybox.spec`) that contains 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.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 to produce the BusyBox kernel and RFS images. The user must change the line `KDI_NAME = developer` to `KDI_NAME = busybox` and then run the **`make all`** command.

Booting the BusyBox Images from a Network

Use the following procedure to boot the BlueCat Linux with the BusyBox utility from a network using the BlueCat Linux OS Loader. Refer to Chapter 2, "Downloading and Booting BlueCat Linux on the Target" for additional details about the BlueCat Linux OS Loader.

1. At the OS Loader prompt, type the following commands:

```
> set IF eth0
> set IP <target_board_IP>
> set HOST <development_host_IP>
> set KERNEL tftp busybox.kernel
> set RFS tftp busybox.rfs
> set CMD console=ttyS0,115200 ramdisk_size=28472
> boot
```

   where *`<target_board_IP>`* is an IP address of the target and *`<development_host_IP>`* is an IP address of the development host.

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

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       8068 Jun 26  2002 /sbin/reboot
/ # chmod 755 /sbin/reboot
/ # ls -la /sbin/reboot
-rwxr-xr-x   1 0       0       8068 Jun 26  2002 /sbin/reboot
```

- echo

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

- date

```
/ # date
Sun Jun  2 18:37:32 UTC 2002
```

- uname

```
/ # uname -a
Linux (none) 2.4.10-1 #7 Wed Jun 26 13:39:16 MSD 2002 ppc unknown
```

- mount

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

- ifconfig

```
/ # ifconfig eth0 192.168.4.13
/ # ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:90:27:0D:C3:3B
      inet addr:192.168.4.13  Bcast:192.168.4.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:1 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:306 (306.0 b)  TX bytes:0 (0.0 b)
      Interrupt:19 Base address:0xf000
```

- ping

```
/ # ping 192.168.4.121
PING 192.168.4.121 (192.168.4.121): 56 data bytes
64 bytes from 192.168.4.121: icmp_seq=0 ttl=255 time=1.1 ms
64 bytes from 192.168.4.121: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 192.168.4.121: icmp_seq=2 ttl=255 time=0.1 ms
```

```
64 bytes from 192.168.4.121: icmp_seq=3 ttl=255 time=0.1 ms
64 bytes from 192.168.4.121: icmp_seq=4 ttl=255 time=0.1 ms

--- 192.168.4.121 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.3/1.1 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 use of the TinyLogin 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 by typing:

   ```
   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. For instance, type the following commands:

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

   ```
   BlueCat:$ make xconfig
   ```

   Select **`Save and Exit`** to update the `.config` file.

   ```
   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 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` symlinks point to `/usr/bin/tinylogin`. This allows the user to change the user's password simply by 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

1:12345: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::/:/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 to produce the TinyLogin kernel and RFS images. The user must change the line `KDI_NAME = developer` to `KDI_NAME = tinylogin` and then run the `make all` command.

Booting the TinyLogin Images from a Network

Use the following procedure to boot the BlueCat Linux with the TinyLogin utility from a network using the BlueCat Linux OS Loader. Refer to Chapter 2,

"Downloading and Booting BlueCat Linux on the Target" for additional details about the BlueCat Linux OS Loader.

1. At the OS Loader prompt, type the following commands:

```
> set IF eth0
> set IP <target_board_IP>
> set HOST <development_host_IP>
> set KERNEL tftp tinylogin.kernel
> set RFS tftp tinylogin.rfs
> boot
```

   where *<target_board_IP>* is an IP address of the target and *<development_host_IP>* is an IP address of the development host.

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

Using TinyLogin Utility

This section provides 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[13]: password for `guest' changed by user `guest'
Password changed.
bash-2.04$ exit
myhostname login: guest
Password: <new_guest_password>
bash-2.04$ exit
```

- Changing the root password:

```
myhostname login: root
login[15]: 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[16]: password for `root' changed by user `root'
Password changed.
bash-2.04# exit
myhostname login: root
Password: <new_root_password>
login[18]: root login  on `console'

bash-2.04# exit
```

- Getting the root permissions:

```
myhostname login: guest
Password: <guest_password>
bash-2.04$ tinylogin su
Password:
login[17]: 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 multiserver and multithread 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

Use 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
```

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.

Select **Save and Exit** to update the .config file.

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

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 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 shutdown halt 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/vtysh               /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

l0:0:wait:/sbin/halt
l6:6: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 172.17.3.11/16

!
! Static default route.
!
ip route 213.24.0.0 255.255.0.0 172.17.0.1

log stdout
```

NOTE: This configuration file sets the Zebra password to `zebra`. The user has to enter this password any time 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 to produce the Zebra kernel and RFS images. The user must change the line `KDI_NAME = developer` to `KDI_NAME = zebra` and then run the `make all` command.

Booting the Zebra Images from a Network

Use the following procedure to boot the BlueCat Linux with the Zebra utility from a network using the BlueCat Linux OS Loader. Refer to Chapter 2, “Downloading and Booting BlueCat Linux on the Target” for additional details about BlueCat Linux OS Loader.

1. At the OS Loader prompt, type the following commands:

```
> set IF eth0
> set IP <target_board_IP>
> set HOST <development_host_IP>
> set CMD ramdisk_size=8192
> set KERNEL tftp zebra.kernel
> set RFS tftp zebra.rfs
> boot
```

   where *<target_board_IP>* is an IP address of the target and *<development_host_IP>* is an IP address of the development host.

The Zebra utility is loaded onto the target board and then 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 00:80:4D:46:22:B8
          inet addr:172.17.3.11  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:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3904  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 172.17.3.11 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
Warning: time of day goes back, taking countermeasures.
64 bytes from 172.17.0.1: icmp_seq=0 ttl=255 time=0 usec
64 bytes from 172.17.0.1: icmp_seq=1 ttl=255 time=650 usec

--- 172.17.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.000/0.325/0.650/0.325 ms

bash-2.04# ping -c 3 213.24.253.87
PING 213.24.253.87 (213.24.253.87) from 172.17.3.11 : 56(84) bytes of
data.
64 bytes from 213.24.253.87: icmp_seq=0 ttl=254 time=902 usec
Warning: time of day goes back, taking countermeasures.
64 bytes from 213.24.253.87: icmp_seq=1 ttl=254 time=1.821 msec
64 bytes from 213.24.253.87: icmp_seq=2 ttl=254 time=898 usec

--- 213.24.253.87 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.898/1.207/1.821/0.434 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

C>* 127.0.0.0/8 is directly connected, lo
C>* 172.17.0.0/16 is directly connected, eth0
S>* 213.24.0.0/16 [1/0] via 172.17.0.1, eth0
Router#
```

CHAPTER 5 *Supported Device Drivers*

Table 5-1 lists the device drivers supported by the dbpxa250 BSP.

Table 5-1: Device Drivers Supported by the dbpxa250 BSP

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
UART, Full Function and Blue Tooth UART controllers of the PXA250	`serial.c`	`drivers/char`	`CONFIG_SERIAL` `CONFIG_SERIAL_CONSOLE`	Blue Tooth controller runs in standard RS-232.
Ethernet Controller, Standard Microsystems LAN91C96 Ethernet controller	`smc9194.c`	`drivers/net`	`CONFIG_SMC9194`	
LCD, Sharp LM8V31 LCD panel	`fb_pxa250.c`	`drivers/video`	`CONFIG_FB_PXA250`	

Flash, Intel StrataFlash 32M+32M	`arm_flash0.c`
	`arm_flash1.c`
		`drivers/mtd/maps`	`CONFIG_MTD_DBPXA250_1`
			`CONFIG_MTD_DBPXA250_1_PART`
			`CONFIG_MTD_DBPXA250_2`
			`CONFIG_MTD_DBPXA250_2_PART`
				Supported via JFFS and an MTD driver

Table 5-1: Device Drivers Supported by the dbpxa250 BSP (Continued)

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
Keyboard, PS/2 keyboard interface on the SA-1111 companion chip	`sa1111_keyb.c` `keyboard.c`	`drivers/char`	`CONFIG_VT`	Not tested explicitly

Mouse, PS/2 mouse interface on the SA-1111 companion chip	`sa1111_keyb.c`	`drivers/char`	`CONFIG_MOUSE`
			`CONFIG_PSMOUSE`
			`CONFIG_VT`
				Not tested explicitly

Touch Screen, UCB1400	`ucblx00-ts.c`	`drivers/misc`	`CONFIG_MCP_UCB1200_TS`
			`CONFIG_MCP_UCB1200`
			`CONFIG_MCP`	

CHAPTER 6 *Known Limitations*

DBPXA250 Target Board Problems and Limitations

- If `mkrootfs` is terminated (either by an error or by a signal), it tries to clean all its temporary files before exiting. Due to certain features of the Cygwin execution environment, however, 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.
- Debugging of multithreaded applications via GDB is not supported.
- The `tclx` RPM package is not included in the Windows-hosted distribution.
- 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.
- The USB host interface of the DBPXA250 is not supported at this time. This is due to a hardware defect of the hardware USB implementation. Specifically, there is a hardware bug in the CPLD U4 on the DCPXA250 processor card. The bug can be described as follows.

  The USB controller uses DMA to perform transfers to the processor card SDRAM. The processor card data bus buffers (U19B and U9) are controlled by the DATA_DIR and DATA_OE signals controlled by the CPLD U4. In the case of a write DMA access from SA-1111 to SDRAM, the direction of the data buffers is incorrect. The problem is due to a bug in the CPLD code:

```
data_dir        <=              ( ...
                                  ...
                                or      (mbgnt_1111 and not(npwe_st));
```

Refer to the following hardware documentation at Intel's Web site for further details:

```
http://www.intel.com/design/pca/\
applicationsprocessors/schems/index.htm
```

- Intel- BBPXA2xx Development Baseboard Schematic Diagram
- Intel- DBPXA250 Development Platform CPLD Source Code
- Intel- DBPXA250 Development Platform Parts Lists
- Intel- DCPXA250 Processor Card Schematic Diagram.