BlueCat Linux Target Support Guide

DOC-0428-00

For Cirrus Logic EP7312 Boards

Product names mentioned in *BlueCat Linux Target Support Guide for Cirrus Logic EP7312 Boards* are trademarks of their respective manufacturers and are used here for identification purposes only.

Printed in the United States of America.

Contents

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 5
Terminal Emulation Parameters 6
EP7312 Keypad Keys 6
Downloading preboot-clep7312.axf and osloader-clep7312.bnd 7
Definitions 7
Utilities – preboot-clep7312.axf and osloader-clep7312.bnd 7
clep_boot 7
Downloading preboot-clep7312.axf and osloader-clep7312.bnd 8
Using i_osloader to Download a Demo System 11
Definitions 11
i_osloader 11
Downloading i_osloader 11
Downloading a Demo System into Target Flash Memory 14
Booting Options 16
Booting a Demo System from Flash Memory 16
Booting a Demo System over a Network 18

CHAPTER 3 KERNEL CONFIGURATION OPTIONS 21

CHAPTER 4 SUPPORTED DEMO SYSTEMS 33

Demo Systems 33

developer Demo System 34

osloader Demo System 34

showcase Demo System 34

Downloading and Booting showcase with an FFS Root Filesystem 35

Modifying Target IP and Gateway Addresses 38

CHAPTER 5 SUPPORTED DEVICE DRIVERS 41

CHAPTER 6 THUMB SUPPORT SOFTWARE 43

Installing BlueCat Linux Thumb Support Software 44

Thumb Support Directory Structure Overview 45

Thumb Cross-Development Environment 46

Placing Thumb Applications onto the Target 48

Thumb Support in the Kernel 48

Thumb Demo Systems 49

Building a Thumb Demo 49

APPENDIX A CLEP_BOOT COMMAND REFERENCE 51

Examples 52

CHAPTER 1 *Overview*

The *BlueCat Linux Target Support Guide for Cirrus Logic EP7312 Boards* provides information about the BlueCat Linux Target Support Package (TSP) for Cogent CSB238-Cirrus 73xx boards (Cirrus Logic EP7312 boards) supporting Cirrus Logic EP7312, Maverick system on chip with ARM CPU core.

Throughout this Target Support Guide (TSG), the TSP is referred to as the "clep7312;" the board as the "EP7312," or simply as the "target board."

The chapters of this TSG provide the information listed below:

- **Chapter 1** is an overview of this TSG's individual chapters.
- **Chapter 2** describes BlueCat Linux download and boot procedures for the EP7312 board using the Bluecat Linux `showcase` demo system as an example.
- **Chapter 3** provides configuration option information about the clep7312 TSP's default BlueCat Linux kernel.
- **Chapter 4** summarizes BlueCat Linux demo systems supported by the clep7312 TSP.
- **Chapter 5** provides a list of clep7312 TSP-supported device drivers, with important information about each of these devices.
- **Chapter 6** describes Thumb, an extension of ARM architecture that allows users to achieve 32-bit performance and address space without the associated costs of going to a 32-bit system.
- **Appendix A** describes the `clep_boot` utility, used to complete target board-specific bootstrapping operations.

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 into the target flash memory via a network, and then booting the demo system on the target board.

Specifically, this chapter demonstrates loading BlueCat Linux `i_osloader` onto the target board, which then enables downloading the `showcase` demo system. The same procedure can be used for downloading other supported demo systems and custom application systems.

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 and the Target Support Package (TSP) is 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 ARM configuration and the clep7312 TSP have been installed on the cross development host, i.e., the user must:

1. Install the BlueCat Linux ARM configuration 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 clep7312 TSP onto the cross development host as detailed in the “Installing Support for Target Boards” section of Chapter 1, “Installation” in the *BlueCat Linux User's Guide.*
3. Activate support for the clep7312 TSP 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

NOTE: *This section and those that follow provide important information about downloading BlueCat Linux into the target flash memory, and then booting BlueCat Linux system on the same target.*

Read all of this section and those that follow before attempting to boot a target board.

The procedure for downloading and booting the BlueCat Linux `showcase` demo system onto an EP7312 target consists of the following main steps:

- Setting up hardware
- Downloading `preboot-clep7312.axf` and `osloader-clep7312.bnd` to enable using the BlueCat OS loader
- Using the BlueCat OS loader, `i_osloader` to download a Demo System

These steps are performed sequentially:

1. Physically connect the hardware. See “Setting up Hardware” on page 5.
2. Download the utilities, `preboot-clep7312.axf` and `osloader-clep7312.bnd`, into target flash memory from the BlueCat Linux cross development host with the `clep_boot` utility via a serial port connection. See “Definitions” and “clep_boot” for descriptions.

3. Use the copy of `osloader-clep7312.bnd` just loaded into the target flash memory to download `i_osloader` from a BlueCat Linux cross development host into target flash memory via a network connection. For a description of `i_osloader`, see “i_osloader” on page 11.
4. Use the copy of `i_osloader` that was just loaded into the target flash memory to boot a BlueCat Linux demo system, such as `showcase`, on the cross development host via a network connection.

Setting up Hardware

Before downloading `preboot-clep7312.axf` and `osloader-clep7312.bnd` onto the target board, set up the hardware by connecting the serial port (`P4`) on the target board to the host terminal emulation program via a null modem serial cable.

Connect the target board ethernet port (`P10`) to a hub, which in turn is connected to the cross development host.

Connect a headset/speaker to the audio line out (`J3`) of the target board. The headset will be used for an MP3 playback demonstration.

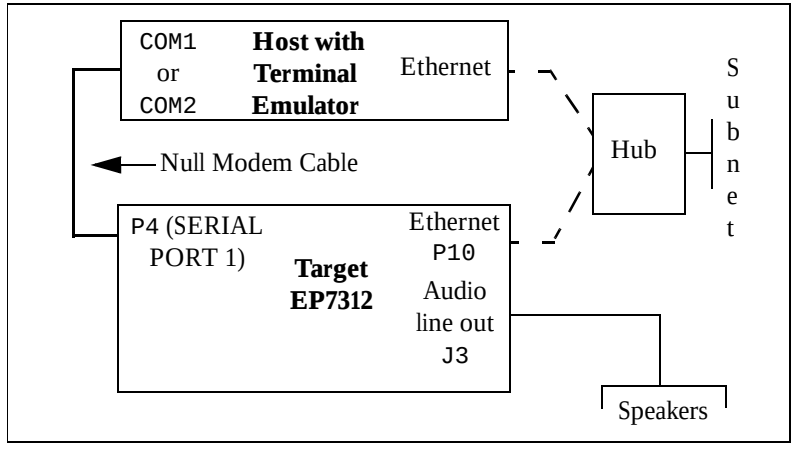

Figure 2-1: Cross Development Host to Target Board Setup

Terminal Emulation Parameters

The following setting of terminal emulation program parameters is recommended:

Bits per second	9600
Data bits	8
Parity	None
Stop bits	1
Flow Control	None

EP7312 Keypad Keys

The keypad keys described in Table 2-1 are useful in performing certain important functions. The user needs to consult the manufacturer documentation to determine the location of these keys on the keyboard.

Table 2-1: EP7312 Keypad Keys

Keypad Key	Description
CPU POR	Resets the EP7312 target board.
CPU BROM	Places the EP7312 target board in Boot ROM Mode if held down while pressing and releasing the **CPU COR** key.

Downloading preboot-clep7312.axf and osloader-clep7312.bnd

Definitions

Utilities – preboot-clep7312.axf and osloader-clep7312.bnd

The utilities, `preboot-clep7312.axf` and `osloader-clep7312.bnd`, are located in the `$BLUECAT_PREFIX/cdt/boot` directory and are downloaded from the cross development host into the target flash memory via a serial port connection.

The following information details `preboot-clep7312.axf` and `osloader-clep7312.bnd`:

`preboot-clep7312.axf`

Resets the EP7312's `COM1` serial port data transfer rate from 9600 bps to 115200 bps.

`osloader-clep7312.bnd`

Contains a *minimum* image needed to initialize an Ethernet interface and provide a BlueCat OS Loader Shell (BLOSH) on the target board.

`preboot-clep7312.axf` and `osloader-clep7312.bnd` are downloaded (in that order) from the cross development host onto the target board using a special BlueCat Linux utility, `clep_boot`.

clep_boot

The `clep_boot` utility is used to download the `preboot-clep7312.axf` and `osloader-clep7312.bnd` utilities into the target flash memory. `clep_boot` is run from a BlueCat Linux cross development host connected via a null modem cable from its `COM1` or `COM2` serial port to the target board's `COM1` serial port (`SERIAL PORT 1`).

When `clep_boot` is invoked on the cross development host, several actions occur. `clep_boot` downloads and transfers control to `preboot-clep7312.axf`. Then `preboot-clep7312.axf` resets the EP7312 serial port's data transfer rate from 9600 bps to 115200 bps; this facilitates a faster transfer of subsequent images from the cross development host to the target board.

`clep_boot` then downloads `osloader-clep7312.bnd` from the cross development host to the target board via the same serial connection that `preboot-clep7312.axf` was transferred over, but now, at the faster 115200 bps rate. Then `osloader-clep7312.bnd` is burned into the target flash memory, where it provides an Ethernet interface and a BLOSH prompt in order to download BlueCat Linux demo systems or customized BlueCat Linux systems.

Downloading preboot-clep7312.axf and osloader-clep7312.bnd

To download `preboot-clep7312.axf` and `osloader-clep7312.bnd` into the target flash memory over a serial port connection, follow these steps:

1. Select and set up hardware as detailed in "Setting up Hardware" on page 5.

NOTE: *BlueCat Linux cross development hosts require* `write` *access to configure serial ports. If the cross development host does not allow* `write` *access for regular users, run BlueCat Linux on the host as* `root`.

---

2. Change to the BlueCat Linux installation directory.
3. Set up the BlueCat Linux environment by entering the following command:

```
BlueCat:# . SETUP.sh clep7312
```

4. Use the `clep_boot` utility to download `preboot-clep7312.axf` and `osloader-clep7312.bnd` into the target flash memory by entering one of the following commands as appropriate:

Linux-based hosts:

```
BlueCat:# clep_boot –c /dev/ttyS0
```

Windows-based hosts:

```
BlueCat:# clep_boot -c COM1
```

Carefully follow the `clep_boot` on-screen instructions.

CAUTION! *Terminal emulation programs cannot be run on the host serial port that is connected to* `COM1` *of the target board* before *and* during *the* `preboot-clep7312.axf` *and* `osloader-clep7312.bnd` *download process. Terminal emulation programs configure the cross development host serial port to a particular baud rate, which may vary from the 115200 bps baud rate used for downloading* `osloader-clep7312.bnd`*. This difference creates a conflict and the installation software becomes dysfunctional on the cross development host as it attempts to download* `osloader-clep7312.bnd` *onto the target board.*

Make sure no other process is running on the host serial port that is connected to `COM1` *of the target board, before and during the download of both* `preboot-clep7312.axf` *and* `osloader-clep7312.bnd`.

`preboot-clep7312.axf` and `osloader-clep7312.bnd` should take less than two minutes to download into the target flash memory.

5. Activate a terminal emulation session on the BlueCat Linux cross development host to observe screen output from the target board's `Serial Port 1`.

6. Reboot the target board by first pressing the **Reset** button (**CPU POR**).

   The screen output below appears on the terminal emulator, validating that the board has rebooted and that a BLOSH prompt (>) is available:

```
Command line: root=101
Uncompressing Linux...done.
Now booting the kernel
Linux version 2.2.12-20 (build@bcm-build1) (gcc
version egcs-2.91.66 19990314/Linux (egcs-1.1.2
release)) #23 Sat Apr 28 16:34:14 MSD 2001
```

```
Calibrating delay loop... 65.33 BogoMIPS
Memory: 14964k/16M available (664k code, 224k
reserved, 344k data, 4k init)
DENTRY hash table entries: 16384 (order: 5, 131072
bytes)
Buffer-cache hash table entries: 32768 (order: 5,
131072 bytes)
Page-cache hash table entries: 4096 (order: 2, 16384
bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society
NET3.039
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
TCP: Hash tables configured (ehash 16384 bhash 16384)
Starting kswapd v 1.5
RAM disk driver initialized:  16 RAM disks of 4096K
size
eth0: cs8900 rev J found at 0x300
cs89x0: EEPROM checksum bad, using default setup
temporarily.
 media RJ-45, IRQ 7, mac =  00 e0 98 06 91 66
RAMDISK: Compressed image found at block 3149364
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 4k init
BlueCat Loader Shell
>
```

Upon successful completion of the `clep_boot` download process, `osloader-clep7312.bnd` automatically boots from the target flash memory every time the board is rebooted by pressing the **Reset** button (**CPU POR**).

NOTE: *LynuxWorks, Inc. does not recommend using* `osloader-clep7312.bnd` *on more than one EP7312 target in the same network segment. Since* `osloader-clep7312.bnd` *uses a compiled-in Ethernet address of* `00:E0:98:06:91:66`*, two installations of it working in the same network segment can cause network problems.*

The `i_osloader` *demo system, however, does not have this limitation, since the Ethernet address of the board can be changed using the* `cs8900a_ether_addr` *kernel option.*

Using i_osloader to Download a Demo System

This section and its subsections provide step-by-step instructions for the process by which any supported demo system (e.g., `showcase`) or custom BlueCat Linux system can be downloaded into target flash memory via a network connection. This is done using the `i_osloader` utility.

Definitions

i_osloader

`i_osloader` is a special version of the BlueCat Linux OS loader Kernel Downloadable Image (KDI). It is used to download the desired BlueCat Linux system (in this case, the `showcase` demo system).

`i_osloader` is located in the `$BLUECAT_PREFIX/demo.clep7312/osloader` directory. It is included in the standard BlueCat Linux distribution for EP7312 boards.

Downloading i_osloader

Note that the BLOSH prompt (>) is displayed on terminal emulation program on cross development host. It is through commands entered at this prompt that `i_osloader` is downloaded to the target board.

To download `i_osloader` on the target board, follow these steps:

NOTE: *These steps provide general information for many versions of Linux. Users need to consult user documentation for their version of Linux to determine any differences.*

---

1. Configure the host as a TFTP server:

   1.1 For example, to configure the TFTP daemon on Red Hat Linux 6.2, uncomment the following line in the `/etc/inetd.conf` configuration file:

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

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

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

where *inetd_pid* is the process identification number of the `inetd` process.

2. Make a `tftpboot` directory under the root directory:

```
BlueCat:# mkdir  /tftpboot
```

3. Copy the files listed below from the `$BLUECAT_PREFIX/demo.clep7312/osloader` directory into the `/tftpboot` directory so that they can be downloaded from there onto the target board:

- `i_osloader.kernel`
- `i_osloader.rfs`

4. Define the IP address for both the cross development host and target board by entering the following commands:

```
> set IP target_IP_address
> set HOST host_IP_address
```

5. Download `i_osloader.kernel` and `i_osloader.rfs` into the target board RAM via the network connection by entering the following commands:

```
> set IF eth0
> set KERNEL tftp \
/tftpboot/i_osloader.kernel
> set RFS tftp /tftpboot/i_osloader.rfs
> boot
```

The following screen display validates that both `i_osloader` files have been successfully downloaded into the target flash memory:

```
eth0: using 10Base-T (RJ-45)
getting /tftpboot/i_osloader.kernel
.....................................................
.................
.............
Received 475434
getting /tftpboot/i_osloader.rfs
.....................................................
.................
.....................................................
.........
Switching to new kernel...
Command line: ramdisk_size=10240
Uncompressing Linux...done.
```

This completes the download of `i_osloader`.

The download process now automatically boots `i_osloader`. The screen output below is displayed:

```
Now booting the kernel
Linux version 2.2.12-1 (build@bcm-build1) (gcc version egcs-
2.91.66 19990314/Linux (egcs-1.1.2 release)) #17 Mon May 7
17:55:47 MSD 2001
NetWinder Floating Point Emulator V0.94.1 (c) 1998 Corel
Computer Corp.
Calibrating delay loop... 65.33 BogoMIPS
Memory: 14124k/16M available (888k code, 224k reserved, 392k
data, 4k init)
DENTRY hash table entries: 1048576 (order: 11, 8388608 bytes)
Buffer-cache hash table entries: 32768 (order: 5, 131072
bytes)
Page-cache hash table entries: 4096 (order: 2, 16384 bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
TCP: Hash tables configured (ehash 16384 bhash 16384)
Starting kswapd v 1.5
RAM disk driver initialized:  16 RAM disks of 10240K size
eth0: cs8900 rev J found at 0x300
cs89x0: EEPROM checksum bad, using default setup temporarily.
media RJ-45, IRQ 7, mac =  00 e0 98 06 91 66
RAMDISK: Compressed image found at block 3149396
Flash mapping for EP7312 board initialized starting at address
0xd0300000
Flash on a EP7312 board: Found 1 x16 CFI device at 0x0 in 16
bit mode
EP7312 Flash MTD driver: Configuration of partitions is :
EP7312 Flash MTD driver: Configured 0 partitions
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 4k init
BlueCat Loader Shell
>
```

After `i_osloader` has been downloaded and booted and the BLOSH (BlueCat Loader Shell) prompt (>) appears, the BlueCat Linux system can be downloaded onto the target board. This is described in “Downloading a Demo System into Target Flash Memory” below.

Downloading a Demo System into Target Flash Memory

To download a BlueCat Linux demo system (`showcase` in this example) into the target flash memory, follow these instructions. Note that these instructions can also be used to load a custom system by replacing the filename, `showcase.kdi`, with the name of the custom system.

1. Define the IP address for both the cross development host and target board by entering the following commands:

```
> set IP 216.100.252.144
> set HOST host_IP_address
> set IF eth0
```

   The target IP address above (`216.100.252.144`) is the default address hardcoded into the `showcase.kdi` file. If the user wishes to change the address, it is necessary to rebuild `showcase.kdi` to change this address. See "Modifying Target IP and Gateway Addresses" on page 38 for details.

NOTE: *On a number of Cirrus Logic EP7312 (Cogent CSB238) boards, the ethernet hardware address is not programmed into EEPROM. Perform the following workaround for the Apache server and the* `showcase` *demo system to work properly:*

```
BlueCat:# cd $BLUECAT_PREFIX
BlueCat:# . SETUP clep7312
BlueCat:# cd $BLUECAT_PREFIX/demo/showcase
BlueCat:# echo "ramdisk_size=8192 \
cs8900a_ether_addr=00:01:02:03:04:05" | mkboot \
-m -k showcase.kernel -f showcase.rfs -c stdin \
showcase.kdi
```

For the same reason as above, perform the following workaround for the FTP client and GDB via a TCP connection in the `developer` *demo system to work properly:*

```
BlueCat:# cd $BLUECAT_PREFIX
BlueCat:# . SETUP clep7312
BlueCat:# cd $BLUECAT_PREFIX/demo/developer
BlueCat:# echo "ramdisk_size=16384 \
cs8900a_ether_addr=00:01:02:03:04:05" | mkboot \
-m -k developer.kernel -f developer.rfs -c \
stdin developer.kdi
```

---

2. Copy the `showcase.kdi` file from the `$BLUECAT_PREFIX/demo.clep7312/showcase` directory into the `/tftpboot` directory so that they can be downloaded from there onto the target board.

3. Enter the following commands to configure the cross development host for downloading `showcase.kdi` into the target flash memory:

```
> set FILE tftp /tftpboot/showcase.kdi

> exec flash_fdisk /dev/mtdchar0 0-23
```

NOTE: *It is important that the BlueCat Linux kernel starts at the beginning of the target's flash memory space (offset* `0x0`*). Make sure that the target flash memory is partitioned accordingly, using the* `flash_fdisk` *tool. Keep in mind that EP7312 on-board flash memory has 128 non-uniform sized sectors that are 128 KB each. As such, the following command would create a single partition in which a BlueCat Linux image of up to 6 MB could be downloaded:*

```
> exec flash_fdisk /dev/mtdchar0 0-47
```

The following screen output validates that the first partition in the range of `0-23` in the target flash memory has been reserved for `showcase.kdi`:

```
EP7312 Flash MTD driver: Configuration of partitions
is 0-23
EP7312 Flash MTD driver: Configured 1 partitions
```

4. Enter the following command to first, erase any data in the target's first flash partition, then burn the `showcase.kdi` image into flash memory:

```
> flash /dev/mtdchar1 erase
```

The screen output below appears, validating that the target's first flash partition has been erased and that `showcase.kdi` has been downloaded to that partition, by listing its file size in number of bytes (in this example `2911232`):

```
Device has 1 regions with sectors of the same size.
Total size of the device is 0x300000 bytes.
Erasing.... done.
eth0: using 10Base-T (RJ-45)
```

```
getting /tftpboot/zimin/kdi
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.................................................
.....................................
Received 2911232
>
```

The commands complete the download of the `showcase` demo system.

The above process can be used to download a custom BlueCat Linux system or application as well as other demo systems supplied with the distribution.

Booting Options

The section details various ways of booting the target board after it has been configured with a BlueCat Linux system.

Booting a Demo System from Flash Memory

At this point, `showcase` has successfully been downloaded into the target flash memory. It is then booted from flash as follows:

Reset the target board by entering the following command:

```
> reset
```

The following screen output appears while the target board is being reset. The boot sequence, when complete, validates that the demo system has booted from the target flash memory by displaying the `bash#` prompt:

```
Command line: ramdisk_size=32000 root=101
Uncompressing Linux...done.
Now booting the kernel
Linux version 2.2.12-1 (bin@bcm-build1.tst) (gcc
version egcs-2.91.66 19990314/Linux (egcs-1.1.2
release)) #4 Sat Apr 28 21:06:12 MSD 2001
NetWinder Floating Point Emulator V0.94.1 (c) 1998
Corel Computer Corp.
Console: colour dummy device 80x25
Calibrating delay loop... 65.33 BogoMIPS
Memory: 12708k/16M available (844k code, 224k
reserved, 412k data, 16k init)
```

```
DENTRY hash table entries: 262144 (order: 9, 2097152
bytes)
Buffer-cache hash table entries: 32768 (order: 5,
131072 bytes)
Page-cache hash table entries: 4096 (order: 2, 16384
bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society
NET3.039
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
TCP: Hash tables configured (ehash 16384 bhash 16384)
Starting kswapd v 1.5
Console: switching to colour frame buffer device 40x30
fb0: frame buffer device, using 112K of video memory
Initialising keyboard driver in a polling mode
ts_7110_init:
ts_7110_init: ts_major=0
audio_7110_init: mcp_major=0
RAM disk driver initialized:  16 RAM disks of 32000K
size
eth0: cs8900 rev J found at 0x300
cs89x0: EEPROM checksum bad, using default setup
temporarily.
 media RJ-45, IRQ 7, mac =  00 e0 98 06 91 66
RAMDISK: Compressed image found at block 3149464
Flash mapping for EP7312 board initialized starting at
address 0xd0300000
Flash on a EP7312 board: Found 1 x16 CFI device at 0x0
in 16 bit mode
EP7312 Flash MTD driver: Configuration of partitions
is :
EP7312 Flash MTD driver: Configured 0 partitions
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 16k init
INIT: version 2.77 booting
INIT: Entering runlevel: 1
Network is configured as follows:

Target  IP address: 216.100.252.144
Gateway IP address:

eth0: using 10Base-T (RJ-45)
Apache server is started.

Microwindows demo is started.

bash#
```

At this point, `showcase` automatically boots from the on-board flash memory every time the board is reset by first pressing the **Reset** button (**CPU POR**).

NOTE: *After being downloaded into target flash memory, the* `showcase` *demo system will start on a board power-up/reset. To boot other embedded systems the user needs to download* `i_osloader` *to target flash memory again.*

Booting a Demo System over a Network

To boot a demo system or a custom embedded system from a cross development host over a network connection using `i_osloader`, use the following procedure:

1. Download `i_osloader` into the target flash memory, as described in the steps detailed in "Downloading preboot-clep7312.axf and osloader-clep7312.bnd" on page 8 through "Downloading i_osloader" on page 11.
2. Boot the BlueCat Linux demo system of choice on the cross development host via a network connection using `i_osloader`.

   Use the following commands to boot the BlueCat Linux `showcase` demo system:

   ```
   > set IP 216.100.252.144
   > set HOST host_IP_address
   > set IF eth0
   > set KERNEL tftp \
   /tftpboot/showcase.kernel
   > set RFS tftp /tftpboot/showcase.rfs
   > set CMD ramdisk_size=8192 \
   cs8900a_ether_addr=00:01:02:03:04:05
   > boot
   ```

NOTE: *On some Cirrus Logic EP7312 (Cogent CSB238) boards, the ethernet hardware address is not programmed into EEPROM. The command* `set CMD ramdisk_size=8192 cs8900a_ether_addr=00:01:02:03:04:05` *is a workaround for the Apache server in the* `showcase` *demo system to work properly.*

Use the following commands to boot the BlueCat Linux `developer` demo system:

```
> set IP 216.100.252.144
> set HOST host_IP_address
> set IF eth0
> set KERNEL tftp \
/tftpboot/developer.kernel
> set RFS tftp /tftpboot/developer.rfs
> set CMD ramdisk_size=16384 \
cs8900a_ether_addr=00:01:02:03:04:05
> boot
```

NOTE: *On some Cirrus Logic EP7312 (Cogent CSB238) boards, the ethernet hardware address is not programmed into the EEPROM. The command* `set CMD ramdisk_size=16384 cs8900a_ether_addr=00:01:02:03:04:05` *is a woraround to enable the FTP client and GDB (via a TCP connection) to work properly in the* `developer` *demo system.*

CHAPTER 3 *Kernel Configuration Options*

The clep7312 TSP comes with a default BlueCat Linux kernel. This kernel has a number of configuration options. These options are detailed in the tables listed in “Table 3-1: Kernel Configuration Options” below:

Table 3-1: Kernel Configuration Options

Configuration Option	Table and Page Numbers
System and Processor Type	Table 3-2, page 22
Code Maturity Level Options	Table 3-3, page 22
Loadable Module Support	Table 3-4, page 23
General Setup	Table 3-5, page 23
Plug And Play Support	Table 3-6, page 24
Block Devices	Table 3-7, page 24
Character Devices	Table 3-8, page 24
Video for Linux	Table 3-9, page 25
Joystick Support	Table 3-10, page 25
Ftape, the Floppy Tape Device Driver	Table 3-11, page 26
Console Drivers	Table 3-12, page 26
Networking Options	Table 3-13, page 26
Amateur Radio Support	Table 3-14, page 27
IrDA Subsystem	Table 3-15, page 27
Network Device Support	Table 3-16, page 28
ARCnet Devices	Table 3-17, page 28
Ethernet (10 or 100 Mbit)	Table 3-18, page 28

Table 3-1: Kernel Configuration Options (Continued)

Configuration Option	Table and Page Numbers
Token Ring Devices	Table 3-19, page 29
WAN Interfaces	Table 3-20, page 29
SCSI Support	Table 3-21, page 30
Filesystems	Table 3-22, page 30
Network Filesystems	Table 3-23, page 31
Partition Types	Table 3-24, page 31
Kernel Hacking	Table 3-25, page 32
LynuxWorks Messenger Support	Table 3-26, page 32
Modular Advanced Power Management	Table 3-27, page 32

Table 3-2: System and Processor Type

Option	Value	Description
CONFIG_ARCH_CLEP7312	Y	ARM system type
CONFIG_CPU_ARM7	Y	Optimize for ARM7 CPU
CONFIG_CPU_32	Y	32-bit processor used

Table 3-3: Code Maturity Level Options

Option	Value	Description
CONFIG_EXPERIMENTAL	N	Prompts for development and/or incomplete code/drivers
CONFIG_ALIGNMENT_TRAP	N	Enables kernel-mode alignment trap handle (Experimental)
CONFIG_TEXT_SECTIONS	N	Splits text into discardable sections

Table 3-4: Loadable Module Support

Option	Value	Description
CONFIG_MODULES	Y	Enables loadable module support
CONFIG_MODVERSIONS	Y	Sets version information on all symbols for modules
CONFIG_KMOD	Y	Kernel module loader support

Table 3-5: General Setup

Option	Value	Description
CONFIG_NET	Y	Networking support
CONFIG_CLEP7312_PREBOOT	N	Generates code suitable for preboot tools
CONFIG_BLUECAT_THUMB	N	BlueCat Linux kernel support for THUMB binaries
CONFIG_BLUECAT_LOADER	N	BlueCat Linux OS loader
CONFIG_BLUECAT_IGNORE_PRINTK	N	BlueCat Linux ignore `printk`
CONFIG_BLUECAT_SMALL_FOOTPRINT	N	BlueCat Linux small memory footprint
CONFIG_BLUECAT_MEMSIZE	N	Memory sizing benchmark
CONFIG_SYSVIPC	N	System V IPC
CONFIG_BSD_PROCESS_ACCT	N	BSD Process Accounting
CONFIG_SYSCTL	N	Systl support
CONFIG_NWFPE	Y	Math emulation
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_ARTHUR	N	RISC OS personality
CONFIG_PARPORT	N	Parallel port support

Table 3-6: Plug and Play Support

Option	Value	Description
CONFIG_PNP	N	Plug and Play support

Table 3-7: Block Devices

Option	Value	Description
CONFIG_BLK_DEV_FD	N	Normal PC floppy disk support
CONFIG_BLK_DEV_IDE	N	Enhanced IDE/MFM/RLL disk/ CD-ROM/tape/floppy support
CONFIG_BLK_DEV_HD_ONLY	N	Old hard disk (MFM/RLL/IDE) driver
CONFIG_BLK_DEV_LOOP	N	Loopback device support
CONFIG_BLK_DEV_NBD	N	Network block device support
CONFIG_BLK_DEV_MD	N	Multiple devices driver support
CONFIG_BLK_DEV_RAM	Y	RAM disk support
CONFIG_BLK_DEV_INITRD	N	Initial RAM disk (`initrd`) support
CONFIG_BLUECAT_RFS	Y	BlueCat Linux RFS support
CONFIG_BLK_DEV_GENERIC_FLASH_DOC	N	M-Systems DiskOnChip
CONFG_BLK_DEV_BLUECAT_SM	N	BlueCat Linux Smart Media support
CONFIG_BLK_DEV_XD	N	XT hard disk support
CONFIG_PARIDE_PARPORT	Y	Parallel port IDE device support
CONFIG_BLK_CPQ_DA	N	Compaq SMART2 support

Table 3-8: Character Devices

Option	Value	Description
CONFIG_VT	Y	Virtual terminal
CONFIG_VT_CONSOLE	Y	Support for console on virtual terminal
CONFIG_SERIAL	N	Standard/generic (dumb) serial support

Table 3-8: Character Devices (Continued)

Option	Value	Description
CONFIG_SERIAL_EXTENDED	N	Extended dumb serial driver options
CONFIG_SERIAL_EP7312	Y	Support for Cirrus Logic EP7312 serial port
CONFIG_SERIAL_EP7312_CONSOLE	Y	Support for console on Cirrus Logic EP7312 serial port
CONFIG_SERIAL_NONSTANDARD	N	Non-standard serial port support
CONFIG_AUDIO_EP7312	Y	Support for Cirrus Logic EP7312 audio device
CONFIG_TOUCH_EP7312	Y	Support for Cirrus Logic EP7312 touch screen
CONFIG_KBD_EP7312	Y	Support for Cirrus Logic EP7312 keyboard
CONFIG_LCD_EP7312	Y	Support for Cirrus Logic EP7312 LCD display
CONFIG_UNIX98_PTYS	N	UNIX98 PTY support
CONFIG_MOUSE	N	Mouse support (not serial mice)
CONFIG_QIC02_TAPE	N	QIC-02 tape support
CONFIG_WATCHDOG	N	Watchdog Timer 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

Table 3-9: Video for Linux

Option	Value	Description
CONFIG_VIDEO_DEV	N	Video for Linux

Table 3-10: Joystick Support

Option	Value	Description
CONFIG_JOYSTICK	N	Joystick support

Table 3-11: Ftape, the Floppy Tape Device Driver

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

Table 3-12: Console Drivers

Option	Value	Description
CONFIG_VGA_CONSOLE	N	VGA text console
CONFIG_FB	Y	Support for frame buffer devices (Experimental)
CONFIG_DUMMY_CONSOLE	Y	Dummy console net driver support
CONFIG_FB_EP7312	Y	EP7312 frame buffer device
CONFIG_FB_VIRTUAL	N	Virtual Frame Buffer support (Only for testing!)
CONFIG_FIXPALETTE_TSBREF	Y	Support for Toshiba TX reference
CONFIG_FBCON_ADVANCED	N	Advanced low-level driver options
CONFIG_FBCON_CFB12	Y	12 bpp packed pixels support
CONFIG_FBCON_FONTWIDTH8_ONLY	N	Supports only 8 pixel wide fonts
CONFIG_FBCON_FONTS	N	Other fonts support
CONFIG_FONT_8x8	Y	VGA 8x8 font support
CONFIG_FONT_8x16	Y	VGA 8x16 font support

Table 3-13: Networking Options

Option	Value	Description
CONFIG_PACKET	Y	Packet socket
CONFIG_NETLINK	Y	Kernel/User netlink socket
CONFIG_RTNETLINK	Y	Routing messages
CONFIG_NETLINK_DEV	Y	Netlink device emulation
CONFIG_FIREWALL	N	Network firewalls
CONFIG_FILTER	Y	Socket filtering

Table 3-13: Networking Options (Continued)

Option	Value	Description
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 auto configuration
CONFIG_IP_ROUTER	N	IP: Optimizes as router not host
CONFIG_NET_IPIP	N	IP: Tunneling
CONFIG_NET_IPGRE	N	IP: GRE tunnels over IP
CONFIG_IP_MROUTE	N	IP: Multicast routing
CONFIG_IP_ALIAS	N	IP: Aliasing support
CONFIG_SYN_COOKIES	N	IP: TCP syncookie support (not enabled per default)
CONFIG_INET_RARP	N	IP: Reverse ARP
CONFIG_SKB_LARGE	N	IP: Allows large windows (not recommended if <16 MB of memory)
CONFIG_IPX	N	IPX protocol
CONFIG_ATALK	N	Appletalk DDP

Table 3-14: Amateur Radio Support

Option	Value	Description
CONFIG_HAMRADIO	N	Amateur radio support

Table 3-15: IrDA Subsystem

Option	Value	Description
CONFIG_IRDA	N	IrDA subsystem support

Table 3-16: Network Device Support

Option	Value	Description
CONFIG_NETDEVICES	Y	Network device support
CONFIG_DUMMY	N	Dummy net driver support
CONFIG_EQUALIZER	N	EQL (serial line load balancing) support
CONFIG_NET_SB1000	N	General Instruments Surfboard 1000
CONFIG_FDDI	N	FDDI driver support
CONFIG_PPP	N	PPP (point-to-point) support
CONGIG_SLIP	N	SLIP (serial line) support
CONFIG_NET_RADIO	N	Wireless LAN (non-ham radio)
CONFIG_NET_FC	N	Fibre Channel driver support
CONFIG_SBNI	N	SBNI 12-xx support

Table 3-17: ARCnet Devices

Option	Value	Description
CONFIG_ARCNET	N	ARCnet support

Table 3-18: Ethernet (10 or 100 Mbit)

Option	Value	Description
CONFIG_NET_ETHERNET	Y	Ethernet (10 or 100 Mbit)
CONFIG_CS8900	Y	Support for Cirrus Logic CS8900 A Ethernet card
CONFIG_ARM_AM79C961A	N	AM79C961A support
CONFIG_NET_VENDOR_3COM	N	3COM cards
CONFIG_LANCE	N	AMD LANCE and PCnet (AT 1500 and NE2100) support
CONFIG_NET_VENDOR_SMC	N	Western Digital/SMC cards
CONFIG_NET_VENDOR_RACAL	N	Racal-Interlan (micom) NI cards

Table 3-18: Ethernet (10 or 100 Mbit) (Continued)

Option	Value	Description
CONFIG_NET_ISA	N	Other ISA cards
CONFIG_NET_EISA	Y	EISA, VLB, PCI, and on-board controllers
CONFIG_PCNET32	N	AMD PCnet32 (VLB and PCI) support
CONFIG_APRICOT	N	Apricot Xen-II on-board Ethernet
CONFIG_CS89x0	N	CS89x0 support
CONFIG_DM9102	N	PCI DM9102 support
CONFIG_DE4x5	N	Generic DECchip & DIGITAL EtherWORKS PCI/EISA
CONFIG_DEC_ELCP	N	DECchip Tulip (dc21x4x) PCI support
CONFIG_DGRS	N	Digi Intl. RightSwitch SE-X support
CONFIG_EEXPRESS_PRO100	Y	EtherExpress PRO/100 support
CONFIG_NE2K_PCI	N	PCI NE2000 support
CONFIG_TLAN	N	TI ThunderLAN support (Experimental)
CONFIG_NET_POCKET	N	Pocket and portable adaptors

Table 3-19: Token Ring Devices

Option	Value	Description
CONFIG_TR	N	Token Ring driver support

Table 3-20: WAN Interfaces

Option	Value	Description
CONFIG_HOSTESS_SV11	N	Comtrol Hostess SV-11 support
CONFIG_COSA	N	COSA/SRP sync serial boards support
CONFIG_SEALEVEL_4021	N	Sealevel Systems 4021 support

Table 3-20: WAN Interfaces (Continued)

Option	Value	Description
CONFIG_DLCI	N	Frame relay DLCI support
CONFIG_SBNI	N	SBNI Leased Line Adapters

Table 3-21: SCSI Support

Option	Value	Description
CONFIG_SCSI	N	SCSI support

Table 3-22: Filesystems

Option	Value	Description
CONFIG_QUOTA	N	Quota support
CONFIG_AUTOFS_FS	N	Kernel automounter support
CONFIG_AFFS_FS	N	Amiga FFS filesystem support
CONFIG_HFS_FS	N	Apple Macintosh filesystem support (Experimental)
CONFIG_FAT_FS	N	DOS FAT filesystem support
CONFIG_MSDOS_FS	N	MS-DOS filesystem support
CONFIG_UMSDOS_FS	N	UMSDOS: UNIX-like filesystem on top of standard MS-DOS filesystem
CONFIG_VFAT_FS	N	VFAT filesystem support
CONFIG_ISO9660_FS	N	ISO 9660 CD-ROM filesystem support
CONFIG_JOLIET	N	Microsoft Joliet CD-ROM extensions
CONFIG_MINIX_FS	Y	Minix filesystem support
CONFIG_NTFS_FS	N	NTFS filesystem support (read only)
CONFIG_HPFS_FS	N	OS/2 HPFS filesystem support (read only)
CONFIG_PROC_FS	Y	/proc filesystem support
CONFIG_ROMFS_FS	N	ROM filesystem support

Table 3-22: Filesystems (Continued)

Option	Value	Description
CONFIG_EXT2_FS	Y	Second extended filesystem support
CONFIG_SYSV_FS	N	System V and Coherent filesystem support
CONFIG_UFS_FS	N	UFS filesystem support
CONFIG_BLUECAT_FFS	N	BlueCat Linux Flash File System support

Table 3-23: Network Filesystems

Option	Value	Description
CONFIG_CODA_FS	N	Coda filesystem support (advanced network filesystem)
CONFIG_NFS_FS	N	NFS filesystem support
CONFIG_LOCKD	N	`lockd` support
CONFIG_SMB_FS	N	SMB filesystem support (to mount WfW shares, etc.)
CONFIG_NCP_FS	N	NCP filesystem support (to mount NetWare volumes)

Table 3-24: Partition Types

Option	Value	Description
CONFIG_BSD_DISKLABEL	N	BSD disklabel (BSD partition tables) support
CONFIG_MAC_PARTITION	N	Macintosh partition map support
CONFIG_SMD_DISKLABEL	N	SMD disklabel (Sun partition tables) support
CONFIG_SOLARIS_X86_PARTITION	N	Solaris (x86) partition table support
CONFIG_NLS	N	Native language support

Table 3-25: Kernel Hacking

Option	Value	Description
CONFIG_FRAME_POINTER	N	Compiles kernel with frame pointer (Useful for debugging)
CONFIG_DEBUG_ERRORS	N	Verbose kernel error messages
CONFIG_DEBUG_USER	Y	Verbose user fault messages
CONFIG_DEBUG_INFO	N	Includes debugging information in kernel binary
CONFIG_MAGIC_SYSRQ	N	Magic SysRq key
CONFIG_BLUECAT_KDBG	N	Includes `kdbg` kernel debugger

Table 3-26: LynuxWorks Messenger Support

Option	Value	Description
CONFIG_BLUECAT_IOPMAN	N	Enables LynuxWorks IOP Manager support
CONFIG_BLUECAT_IOPMAN_SITKA	N	IOP Manager Sitka support
CONFIG_BLUECAT_MSNG	N	Enables Messenger support
CONFIG_BLUECAT_MINET	N	MINET support
CONFIG_BLUECAT_MSNG_API	N	Messenger user API support

Table 3-27: Modular Advanced Power Management

Option	Value	Description
CONFIG_BLUECAT_APM	N	Modular Advanced Power Management support

CHAPTER 4

Supported Demo Systems

This chapter provides information about BlueCat Linux demo systems supported by the clep7312 Target Support Package (TSP).

Demo Systems

Table 4-1 lists the demo systems supported in the clep7312 TSP standard distribution, boot devices supported by each demo system, and their RAM and ROM requirements:

Table 4-1: Demo Systems Supported by clep7312 TSP

Demo	Default Boot Devices Supported	ROM Requirements	RAM Requirements
`developer`	**Flash** **Network** using the OS loader	2760 KB	10240 KB
`osloader`	**Flash** **Network** using the OS loader	1412 KB	12288 KB
`showcase`	**Flash** **Network** using the OS loader	28531 KB	11264 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 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

This section provides information on the `showcase` demo system. The `showcase` demo system showcases certain features specific to the EP7312 target board, and configures an Apache web server. A generic description can be found in Chapter 4 of the *BlueCat Linux User's Guide*. Board-specific features are described below.

DEMO

An Apache web server and a simple MicroWindows MP3 player demonstration

SYNOPSIS

This demo system starts and configures the Apache HTTP daemon turning the target board into a Web server. At the same time, a simple MicroWindows MP3 application is started.

REQUIREMENTS

Storage	Small
RAM	Small
Network	Yes
Disk	None
Special	None
Kernel Option	`ramdisk_size=8000`

DESCRIPTION

The system boots up in single-user mode. `init` starts `bash` without a login prompt. `bash` automatically runs the `/.bashrc` script file, which executes commands to bring up the network interface and set up the kernel routing table. The network configuration is displayed on the terminal connected to the serial line. `.bashrc` starts the `httpd` daemon and a MicroWindows MP3 demo application.

The MicroWindows MP3 demo application draws a window with a picture on the screen starts playing an MP3 audio file. If the user touches the application window, the demo flips from one picture to another, while stopping and restarting the MP3 application. When the demo finishes playing the MP3 file, it pauses for 5 seconds before replaying the file from the beginning. Pressing the **Close** button of the application window causes the demo to quit.

After the demo system is booted the Apache server is accessible from any networked machine using the IP address entered by the user, and serves the `index.html` page located in the `showcase` subtree in the `demo` directory.

The `showcase` demo system can be downloaded into target flash memory either as a BlueCat Linux composite image composed of kernel and root filesystem images, or as a BlueCat Linux image containing a kernel image and an FFS image mounted as root. The former case is detailed under “Downloading showcase into Target Flash Memory” in Chapter 2, and the latter in “Downloading and Booting showcase with an FFS Root Filesystem” below.

Downloading and Booting showcase with an FFS Root Filesystem

This section describes how to configure and download the `showcase` demo system into the target flash memory with the Flash File System image mounted as a root filesystem. This procedure assumes the target board is booted and has the `i_osloader` prompt (>) displayed.

1. Rebuild the `showcase.kdi` BlueCat Linux image so it does not include a compressed root filesystem image. Also, the kernel command line in the BlueCat Linux image must have the **`ep7312_part_conf`** and **`root`** options enabled. The flash memory partition configuration string must be in accordance with the argument supplied to the `flash_fdisk` command in the step below. The following command issued on the cross development host rebuilds the BlueCat Linux image and sets the appropriate kernel command line:

```
BlueCat:# cd \
$BLUECAT_PREFIX/demo/showcase

BlueCat:# echo “ep7312_part_conf=\
\”4-43\” root=1f01” | mkboot -m -k\
showcase.kernel -c stdin showcase.kdi
```

2. Copy the `showcase.kdi` and `showcase.jffs` files from the `$BLUECAT_PREFIX/demo.clep7312/showcase` directory into the `/tftpboot` directory so that they can be downloaded from there onto the target board.

3. Partition the target flash memory by creating two partitions, one for the kernel image and the other for the root filesystem image. Enter the following command at the `i_osloader` prompt on the target board:

```
> exec flash_fdisk /dev/mtdchar0 \
0-3:4-43
```

   This command creates a .5 MB first partition for the kernel at the beginning of target flash memory, and a second partition of 5 MB for the root filesystem image right after the first partition.

4. Set the environment variables to properly configure network access by entering the following command:

```
> set IP target_IP_address

> set HOST host_IP_address
```

```
> set IF eth0
```

5. Set the FILE environment variable so that it points to the showcase root filesystem FFS image by entering the following command:

```
> set FILE tftp /tftpboot/showcase.jffs
```

6. Burn the FFS image into the target flash memory, by entering the following command:

```
> flash /dev/mtdchar2 erase
```

7. Set the FILE environment variable so that it points to the showcase BlueCat Linux image, for example, by entering the following command:

```
> set FILE tftp /tftpboot/showcase.kdi
```

NOTE: *To download and boot other custom systems with FFS mounted as root, the kernel needs to be rebuilt with FFS and MTD (Memory Technology Device) enabled.*

8. Burn the BlueCat Linux image into the target flash memory, by entering the following command:

```
> flash /dev/mtdchar1 erase
```

9. Reset the target board by entering the following command:

```
> reset
```

NOTE: *Because the kernel must scan the entire flash partition, mounting the* showcase *FFS image as a root file system takes about 40 seconds during the kernel bootstrap process.*

Modifying Target IP and Gateway Addresses

The `showcase` demo system includes an Apache web server feature. The Apache web server is a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (web) server. See `http://www.apache.org` for further information.

Users may wish to define their own target IP address, rather than use the default address (`216.100.252.144`) defined in the `showcase` demo system. To change the default IP address for the `showcase` demo system, open the `showcase` demo system's `.bashrc.clep7312` file with any text editor (such as vi), and perform the following steps:

1. Change directory to the location of the `.bashrc.clep7312` file:

   ```
   BlueCat:# cd \
   $BLUECAT_PREFIX/demo.clep7312/\
   showcase/local/etc
   ```

2. Change the `showcase` demo system's default IP address (`216.100.252.144`) to a new user-selected target IP address, for example, `216.100.252.140`, by editing the following line in the `.bashrc.clep7312` file:

   ```
   TARGET_IP=216.100.252.140
   ```

3. Set a gateway IP address by entering a value at this line:

   ```
   GATE_IP=
   ```

4. Rebuild the `showcase` demo system by entering the following commands under the `$BLUECAT_PREFIX/demo.clep7312/showcase` directory:

   ```
   Bluecat:# touch showcase.spec

   Bluecat:# make rootfs

   Bluecat:# make kdi
   ```

5. Copy the updated `showcase.kdi` to the `/tftpboot` directory to replace the old `showcase.kdi`.

Now when the `showcase` demo system is downloaded onto the target board, the Apache web server BlueCat Linux web pages can be accessed at the user's customized IP address, as shown in "Figure 4-1: BlueCat Linux Apache Web Server Web Page Example" on page 39.

Figure 4-1: BlueCat Linux Apache Web Server Web Page Example

CHAPTER 5 Supported Device Drivers

Table 5-1 lists the device drivers supported by the clep7312 TSP and details important information about them:

Table 5-1: Device Drivers Supported by the clep7312 TSP

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
UART Two 16550-compatible devices	`serial_ep7212.c`	`drivers/char`	`CONFIG_SERIAL_EP7312` `CONFIG_SERIAL_EP7312_CONSOLE`	
Touch Screen Maxim MAX148	`ts_ep7212.c`	`drivers/char`	`CONFIG_TOUCH_EP7312`	
LCD Controller	`con_ep7212.c`	`drivers/char`	`CONFIG_LCD_EP7312`	
LCD Frame Buffer	`fb_ep7212.c`	`drivers/video`	`CONFIG_FB_EP7312`	
Digital Audio Interface CS 434x CS 533x	`audio_ep7212.c` `mcp.S`	`drivers/char` `drivers/char`	`CONFIG_AUDIO_EP7312`	The driver works only if DAC is enabled in hardware by setting the JP36 jumper to the position 1-2.
Ethernet CS8900A	`cs8900a.c` `cs8900a.h`	`drivers/net`	`CONFIG_CS8900A`	

Table 5-1: Device Drivers Supported by the clep7312 TSP (Continued)

Hardware Device	Device Drivers	Location in Source Tree	Kernel Configuration Options	Notes
Keyboard Scan matrix	kbd_ep7212.c keymap_ep7212.map	drivers/char drivers/char	CONFIG_KBD_EP7312	
Real Time Clock	rtc_ep7212.c rtc_ep7212.h	drivers/char include/linux	CONFIG_RTC	
NOR Flash	ep7312.c	drivers/mtd	CONFIG_MTD_EP7312 CONFIG_MTD_EP7312_PART	
SmartMedia	sm.c smcommon.c sm7312.c smart1,2.h sm.h	drivers/block	CONFIG_BLK_DEV_BLUECAT_SM	
IDE	ide.c ide-disc.c	drivers/block	CONFIG_BLK_DEV_IDE CONFIG_BLK_DEV_IDEDISK	

CHAPTER 6 *Thumb Support Software*

Thumb is an extension of ARM (Advanced RISC Machine) architecture that allows users to achieve 32-bit performance and address space without incurring the costs associated with going to a 32-bit system. It has 36 instruction formats drawn from the standard 32-bit ARM instruction set. These have been recoded into 16-bit wide op-codes. This results in very high code density, as Thumb instructions are half the width of ARM instructions. On execution, these new 16-bit Thumb op-codes are decompressed by the processor to their ARM instruction set equivalents. The decompressed 16-bit Thumb op-codes are then run on an ARM core.

Thumb is not just another mixed instruction-set concept. Thumb-aware cores have two separate instruction sets. This provides a unique advantage over single instruction-set cores, as Thumb provides designers with all the power of ARM's 32-bit instruction set while simultaneously maintaining the code-size advantages of the Thumb double instruction-set design. The fact that the two instruction sets that make up the Thumb double instruction set are distinct, results in extremely simple decoding logic. This simplification enables small silicon size and maintains ARM's industry-leading low-power capability and MIPS/Watt performance.

Thumb-aware cores—such as the ARM7TDMI—have a full 32-bit architecture, similar to ARM. Because of this, designers retain 32-bit RISC performance. The combination of the two instruction-sets running on a 32-bit Thumb-aware core provides an effective solution for code size and performance issues of 16-bit systems.

NOTE: *Not all ARM cores support* Thumb. *For example, the StrongARM SA-110, supported by BlueCat Linux, does not contain a* Thumb *extension.*

Installing BlueCat Linux Thumb Support Software

BlueCat Linux Thumb support software is distributed on a separate CD-ROM from the rest of the BlueCat Linux distribution CD-ROMs. The Thumb software CD-ROM is used to install Thumb support on top of the BlueCat Linux development environment installed on the cross development host. (See the *BlueCat Linux User's Guide* for a detailed description of the BlueCat Linux installation procedure.)

To install the Thumb support software follow these steps:

1. Insert the Thumb distribution CD-ROM in the cross development host CD-ROM drive.
2. If the cross development host does not auto-mount the CD-ROM, mount the CD-ROM on the host.
3. Go to the top of the BlueCat Linux installation for the ARM target board, and enable the BlueCat Linux execution environment:

```
$ cd $HOME/bluecat

$ . SETUP.sh clep7312

BlueCat:$
```

4. Run the **`install`** installation script from the top of the Thumb distribution CD-ROM to install Thumb support cross-development and target support packages in the BlueCat Linux environment:

```
BlueCat:$ /mnt/cdrom/install
```

Upon successful completion of the steps above, Thumb support is installed in the BlueCat Linux development environment on the cross development host.

Thumb Support Directory Structure Overview

The installation procedure described in “Installing BlueCat Linux Thumb Support Software” on page 44 results in the creation of the Thumb-specific files and directories shown in gray in Figure 6-1:

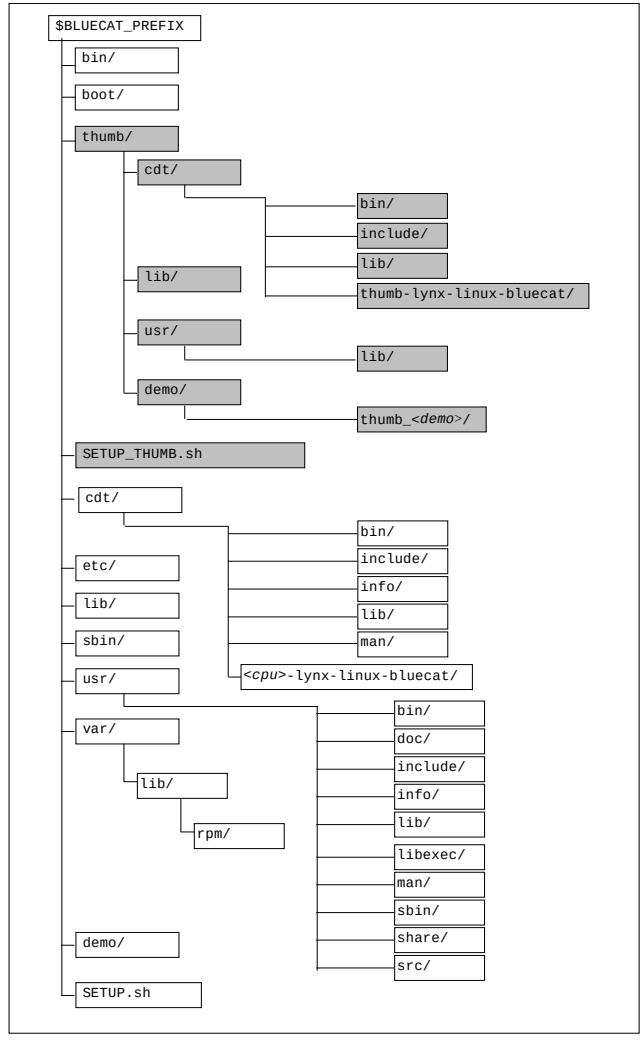

Figure 6-1: Thumb-Specific Directories and Files

The `thumb/cdt` subtree contains the Thumb cross-development tools.
The `thumb/lib` subtree contains the Thumb target libraries.
The `thumb/demo` subtree contains the Thumb-specific demo systems.

Table 6-1 briefly describes the key components of the Thumb-specific directories in the BlueCat Linux tree:

Table 6-1: Key Components of Thumb-Specific Directories

Node	Description
`$BLUECAT_PREFIX`	Installation directory
`- thumb/`	Thumb-specific subtree
`-- cdt/`	Thumb cross-development tools
`--- bin/`	Thumb cross development binaries
`--- include/`	Thumb cross development `include` files
`--- lib/`	Thumb cross development libraries
`--- thumb-lynx-linux-bluecat/`	Thumb cross development binaries
`-- lib/`	Thumb libraries
`-- usr/`	Thumb top-level `usr` directory
`--- lib/`	More Thumb libraries
`-- demo/`	BlueCat Linux configurations
`--- thumb_`*demo*`/`	Thumb-specific demo system configurations
`SETUP_THUMB.sh`	Shell script for setting up Thumb environment

Thumb Cross-Development Environment

The Thumb cross development environment does not replace the cross development environment for the ARM mode. Instead, the cross development tools for both execution modes are available in parallel.

Thumb cross development tools are available in the directory `$BLUECAT_PREFIX/thumb/cdt` in the BlueCat Linux development tree.

The GNU toolkit does not support switching between compiling for the ARM instruction set and the Thumb instruction set on anything other than a

per-file basis. There are two completely different compilers. One produces the ARM assembler instructions, and the other produces the Thumb assembler instructions.

The standard development tools for the ARM mode are the default configuration, while the Thumb development tools are optional. This means that the `gcc` (`cc`, `egcs`) commands start the standard ARM compiler. To run the Thumb compiler, users can choose one of the following options:

- Enter a command specifying the full path to the compiler:

  ```
  $BLUECAT_PREFIX/thumb/cdt/\
  thumb-lynx-linux-bluecat/bin/gcc
  ```

- Use one of the Thumb-specific commands, **`tcc`**, **`tgcc`** or **`tegcs`** to start the Thumb compiler. These Thumb-specific tools reside in the BlueCat Linux tree along with the standard cross development tools in the `$BLUECAT_PREFIX/cdt/\ thumb-lynx-linux-bluecat/bin` directory. The commands are symbolic links to the actual Thumb tools, as shown here:

  ```
  tcc -> ../../../thumb/cdt/thumb-lynx-linux \
  -bluecat/bin/cc
  tegcs -> ../../../thumb/cdt/thumb-lynx-linux \
  -bluecat/bin/egcs
  tgcc -> ../../../thumb/cdt/thumb-lynx-linux \
  -bluecat/bin/gcc
  ```

  The Thumb-specific commands shown above are available to the user immediately after the BlueCat Linux Thumb-specific `SETUP_THUMB.sh` script is executed.

  Run the `SETUP_THUMB.sh` script from the BlueCat Linux root directory by entering the following command:

  ```
  $ cd $BLUECAT_PREFIX

  $ . SETUP_THUMB.sh

  BlueCat:$
  ```

  Once the script above is executed, the standard commands, `cc`, `gcc` or `egcs` run the Thumb tools.

A new option, `--thumb-bc`, has been added to the standard linker `ld`. This option is used to mark the executable as a Thumb application. In order to produce a Thumb executable from the object files, the user must specify the **`--thumb-bc`** option in the command line. Use the **`-static`** option

to build a statically linked Thumb application, for example, by entering the following command:

```
ld ... -o thumbexec ... file1.o file2.o --thumb-bc \
-static
```

If the Thumb executable is linked by calling the Thumb-specific `tgcc`, the `--thumb-bc` option is generated and passed to the `ld` command automatically.

A new option, `-mthumb-bc`, has been added to the standard assembler to specify the Thumb assembler instructions in the input file.

Placing Thumb Applications onto the Target

Users can combine ARM and Thumb applications on the target board and execute them in parallel. Use the **`mkrootfs`** cross development tools to put Thumb applications in a filesystem downloaded onto the target board.

Thumb Support in the Kernel

Users must enable Thumb support in the BlueCat Linux kernel to run Thumb applications on the target board. Thumb support is configured in the kernel using the **`CONFIG_BLUECAT_THUMB`** kernel configuration option.

Use the standard kernel configuration tools (for instance, `make xconfig`) to enable the Thumb support in the kernel. From the main menu, go to **General Setup**, and enable the BlueCat Linux kernel support for the Thumb binaries option.

Thumb Demo Systems

The following demo systems reside in the `$BLUECAT_PREFIX/thumb/demo` directory and describe the build and execution of Thumb applications on the 7T ARM target board:

- The `hello_thumb` demo system shows the execution of an `init`-like Thumb application on the target board.
- The `shell_thumb` demo system shows the execution of a Thumb application in parallel with ARM applications. Its functionality is identical to that of a normal `shell` demo system. This demo system starts the Thumb mode shell, and then runs all the other included utilities in the ARM mode to demonstrate how they can be executed in parallel.
- The `hello_dynamic_thumb` demo system shows the dynamically-linked `hello_world` application.
- The `gdb_thumb` demo system shows debugging of a simple Thumb application with the Thumb GDB debugger.

Building a Thumb Demo

The demo systems included in the Thumb distribution were originally configured for a specific TSP; the `.config` filenames reflect the relevant board's name. If users wish to run these demo systems on other ARM target boards such as, in this case, the EP7312, they must reconfigure the kernel by running `make xconfig`. For example, to build the `shell_thumb` demo system for EP7312 target boards, users would perform the following steps:

1. To use the `shell_thumb` demo system on the EP7312 board, copy the original `.config` file (`arm_willow.config`) to a new file with a `clep7312` suffix by entering the following command:

   ```
   BlueCat:$ cp \
   shell_thumb_arm_willow.config \
   shell_thumb_clep7312.config
   ```

2. Set up the kernel to run on the EP7312 board by entering the following command:

   ```
   BlueCat:$ make xconfig
   ```

3. Enter the **System and processor type** submenu, and change the **CMA120/Willow** system type to **CL-EP7312-based**.
4. Return to the main menu.
5. Enter the **Character devices** submenu.
6. Deselect the `Standard/generic (dumb) serial support` option.
7. Select the `Cirrus Logic EP7312 serial port support` and `Support for console on serial port` options.
8. Return to the main menu.
9. Select the **Save** and **Exit** button to save the new `.config` file.
10. Build the demo system by entering the following command:

```
BlueCat:$ make
```

This series of instructions and commands successfully builds a `shell_thumb` demo system. These steps can be repeated for all other Thumb demo systems on the distribution.

For more information on downloading and booting this demo on a target board, refer to Chapter 2 of this document.

APPENDIX A *clep_boot Command Reference*

This appendix describes the `clep_boot` utility.

UTILITY

`clep_boot` – Downloads `preboot-clep7312.axf` and `osloader-clep7312.bnd` into target flash memory.

SYNOPSIS

```
clep_boot [-h] [-v] [-c port]
```

DESCRIPTION

The `clep_boot` utility downloads `preboot-clep7312.axf` and `osloader-clep7312.bnd` into target flash memory over a serial connection. To download `preboot-clep7312.axf` and `osloader-clep7312.bnd`, set up the BlueCat Linux cross development environment on the cross development host (**`. SETUP.sh`**), run **`clep_boot`**, and follow the on-screen instructions.

On the cross development host, the `clep_boot` utility requires write access to the serial port special files. If the cross development host does not allow such access for regular users, this program must be run as `root`.

When `clep_boot` is invoked on the cross development host, `clep_boot` downloads and transfers control to `preboot-clep7312.axf`. `preboot-clep7312.axf` resets the EP7312 serial port baud rate from 9600 bps to 115200 bps; this facilitates a faster transfer of `osloader-clep7312.bnd` to the target board.

`clep_boot` then downloads `osloader-clep7312.bnd` from the cross development host to the target board via the

same serial connection that `preboot-clep7312.axf` was transferred over, but at the faster 115200 bps rate. `osloader-clep7312.bnd` burns itself into target flash memory, where it provides an Ethernet interface and a BLOSH prompt in order to download BlueCat Linux demo systems or customized BlueCat Linux kernel images.

NOTE: `osloader-clep7312.bnd` *uses a compiled-in Ethernet address* `00:E0:98:06:91:66`. *Using multiple copies of such versions of the OS loader in the same network segment will cause network errors. LynuxWorks, Inc. recommends downloading the full version of the OS loader into target flash memory from this limited version immediately after running* `clep_boot`.

OPTIONS

`-h`	Prints a short instruction on the program usage.
`-v`	Verbose output—if this option is specified, all messages coming from the board are printed out.
`-c` *name*	This option tells the program to use port *name* instead of the default port (where *name* is either `/dev/ttyS1` or `dev/ttyS0` for Linux, or `COM1`, `COM2` for Windows).

Examples

To download `osloader-clep7312.bnd` into target flash memory over a serial cable connected to the cross development host serial `COM2` port (the default host serial port), enter the following command:

```
BlueCat:# clep_boot
```

Follow the instructions printed on the screen.

If the cable is connected to the cross development host's serial `COM1`, enter the following commands:

- on Linux hosts:

```
BlueCat:# clep_boot -c /dev/ttyS0
```

- on Windows hosts:

```
BlueCat:$ clep_boot -c COM1
```

Follow the instructions displayed on the screen.