

# **BSP Targeting the Freescale MPC8308E RDB Board**

User's Guide

Rev. 1.0  
04/2010



# Content

About This Manual.....	2
Audience .....	2
Organization.....	2
Definitions, Acronyms, and Abbreviations.....	2
1 Introduction .....	4
1.1 LTIB Overview .....	4
1.2 BSP Overview .....	4
2 LTIB Basics.....	6
2.1 Installing the BSP.....	6
2.2 Running LTIB .....	6
3 Target Configuration.....	8
3.1 Target System Memory Map.....	8
3.2 Target Set-up.....	8
3.2.1 Standalone Mode Set-up .....	8
4 Target Deployment.....	10
4.1 Host Set-up.....	10
4.2 Flash U-Boot .....	11
4.3 Configure U-Boot.....	11
4.3.1 Configure U-Boot for Ramdisk Deployment .....	11
4.4 Flash Vitesse Binary.....	13
4.5 Flash Kernel and Ramdisk .....	13
4.6 Deployment .....	13
4.6.1 Deploy Ramdisk from TFTP.....	13
4.6.2 Deploy Ramdisk from Flash .....	14

## About This Manual

This User Manual provides information on the basic features supported by the BSP and provides you with instructions about how to accomplish these tasks:

- Install the BSP on a host development system.
- Run Linux Target Image Builder (LTIB) to build target images.
- Deploy built images to the MPC8308E RDB board.
- Boot Linux on the MPC8308E RDB board.

## Audience

This document is addressed to developers who want to take advantage of the Freescale Linux Target Image Builder (LTIB) for the MPC8308E RDB Board Support Package (BSP).

## Organization

This document is organized into 4 chapters.

- |           |   |
|-----------|---|
| Chapter 1 | Provides an introduction to the MPC8308E RDB BSP. |
| Chapter 2 | Provides basic information on LTIB.               |
| Chapter 3 | Provides important target set-up information.     |
| Chapter 4 | Provides host and target deployment information.  |

## Definitions, Acronyms, and Abbreviations

The following list defines the abbreviations used in this document.

BSP	Board Support Package
DDR	Dual Data Rate
eTSEC	Enhanced Three-Speed Ethernet Controller
JFFS2	Journalling Flash File System version 2
LTIB	Linux Target Image Builder
MTD	Memory Technology Device
PCIE	PCI Express

RTC	Real Time Clock
TFTP	Trivial File Transfer Protocol
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

# 1 Introduction

This User Manual provides information on the basic features supported by the BSP and provides you with instructions about how to accomplish these tasks:

- Install the BSP on a host development system.
- Run Linux Target Image Builder (LTIB) to build target images.
- Deploy built images to the MPC8308E RDB board.
- Boot Linux on the MPC8308E RDB board.

## 1.1 LTIB Overview

The Linux Target Image Builder (LTIB) is a tools framework used to manage, configure, extend and build Linux software elements to easily build a Linux target image and a root filesystem. LTIB runs on an x86 PC running the Linux OS.

This BSP operates with LTIB running on a host development system with the following:

- Ethernet card
- Serial port
- 1 GB of free disk space required
- NFS Server
- TFTP Server
- rsync
- perl

## 1.2 BSP Overview

This MPC8308ERDB BSP is designed for use with LTIB. Once the BSP is installed and running with its basic configuration, you can use LTIB to customize your project.

The BSP components provide the tools, device drivers, and additional features needed for your embedded Linux project.

Linux 2.6.29.6 kernel

- Targeting Freescale MPC8315E RDB board
- LTIB integration
- Linux 2.6.29.6 kernel supporting the e300c3 core
- DUART driver
- PCIE driver supporting PCIe bus in x1 mode
- I2C driver
- eTSEC1& eTSEC2 driver to support 10M/100M/1000M Ethernet function
- USB 2.0 host
- USB 2.0 gadget mode device driver
- RTC driver

- MTD driver supporting NAND flash
- JFFS2 filesystem support NAND flash memory

#### Bootloader version and information

- U-Boot 2009.11-rc1 plus Freescale patches
- Boot from NOR/NAND Flash
- e300c3 core initialization
- DDR2 SDRAM initialization
- NOR Flash Read/Write operation
- NAND Flash Read/Write operation
- DUART port at 115200 baud rate without flow control
- eTSEC operation supporting TFTP
- Load kernel and ramdisk filesystem from NOR Flash
- Support for CPU/board-specific configuration
- PCIE host driver initialization
- eTSEC operation supporting TFTP

#### Toolchain version

- Gcc4.1.2, eglibc-2.5.78, binutils 2.18, CodeSourcery release 78, supporting e300c3/c4 cores with NPTL

## 2 LTIB Basics

### 2.1 Installing the BSP

Please follow the steps below to install LTIB on your host machine.

1. As root, mount the ISO image on your machine:  

```
mount -o loop MPC8308ERDB_201004xx-ltib.iso /mnt/cdrom
```
2. As a non-root user, install the LTIB:  

```
/mnt/cdrom/install
```

You will be prompted to input the desired LTIB install path. Be sure the user has the correct permissions for the install path.

There are no uninstall scripts. To uninstall LTIB you need to remove the `/opt/freescale/pkgs`, `/opt/freescale/ltib` and `<install_path>/ltib-mpc8308erdb-<date>` directories manually.

### 2.2 Running LTIB

To run LTIB, change to the directory into which you installed it and run `./ltib`.

```
cd <install_path>/ltib
./ltib
```

At the first time LTIB runs on your machine a number of host packages are built and installed that support LTIB. This may take a few minutes.

**Important Note:** Please be sure to set the “Target System Configuration” options for your network environment the first time you build.

The LTIB can only run by a non-root user.

To modify the project configuration simply run:

```
./ltib --configure (or -c; type --help to see configuration options)
```

This will re-prompt you for the platform/board configuration. In the board configuration screens, change Sets and select packages as appropriate. When you exit the configuration screen your target image will be adjusted accordingly.

Once you build your project you will get following directory/image files:

- **rootfs.ext2.gz.uboot** – ramdisk image that can be loaded with U-Boot
- **rootfs.jffs2** – JFFS2 filesystem image
- **rootfs/** – directory, the root file system that will be deployed on your board.
- **rootfs/boot/ulmage** – kernel image that can be loaded with U-Boot
- **rootfs/boot/u-boot.bin** – U-Boot binary image that can be programmed into MPC8308ERDB board NOR Flash.
- **rootfs/boot/u-boot-nand.bin** – U-Boot binary image that can be programmed into MPC8308E RDB board NAND Flash.
- **rootfs/boot/mpc8308erdb.dtb** – device tree binary (dtb) files for kernel boot up

If you want to fully re-configure and re-compile all the packages, you can do the following. This is generally not necessary.

1. Clean up all the configure files and objects thoroughly:
 

```
./ltib -m distclean
```
2. You will be prompted to confirm your choice. Type yes to perform a distclean.
3. Run ltib
 

```
./ltib
```

More information on LITB can be found in `<install path>/ltib-mpc8308eRDB-<date>/doc`. Or on the web at <http://savannah.nongnu.org/projects/ltib>.



## 3 Target Configuration

### 3.1 Target System Memory Map

After system startup, the boot loader maps system memory as shown below.

Range Start	Range End	Definition	Size
0x0000_0000	0x07ff_ffff	DDR	128M
0xa000_0000	0xffff_ffff	PCI Express 1 Mem	256M
0xb000_0000	0xb0ff_ffff	PCI Express 1 Config	16M
0xb100_0000	0xb17f_ffff	PCI Express 1 IO	8M
0xe000_0000	0xe00f_ffff	IMMR	1M
0xe060_0000	0xe060_7fff	NAND Flash (CS1)	32K
0xfe00_0000	0xfe7f_ffff	NOR Flash (CS0)	8M

The 8M NOR flash starts at address **0xfe000000**:

Range Start	Range End	Definition
0xfe00_0000	0xfe05_ffff	U-Boot
0xfe0a_0000	0xfe2f_ffff	Kernel
0xfe30_0000	0xfe7e_ffff	Ramdisk file system
0xfe7f_0000	0xfe7f_3fff	DTB
0xfe7f_e000	0xfe7f_ffff	Vitesse

### 3.2 Target Set-up

#### 3.2.1 Standalone Mode Set-up

1. Connect the MPC8308E RDB board to the network via the eTSEC 1 port on the board.
2. Connect the MPC8308E RDB board to the host machine via the serial port with an RS-232 cable.
3. Setup the hyperterminal in the host machine with 115200bps, 8-N-1.

4. Verify all the switches and jumpers are setup correctly to default value. The default frequency is 400/133 MHz (CPU/coherent system bus ):

**Table 1 MPC8308E RDB DIP Default Switches Sets**

Reference	1	2	3	4
S1	OFF	OFF	OFF	OFF
S2	ON	ON	ON	ON

ON – 0    OFF - 1

- S1: Board revision and boot Flash selection switch.  
S2: Reset configuration word source selection switch.

5. Connect the board to power supply.
6. Power on the board.

## 4 Target Deployment

This chapter describes how to deploy the U-Boot, Linux kernel and File system to the Target board.

By default, when the board is powered up, it will display the information and stop at the U-Boot prompt.

### 4.1 Host Set-up

Host setup is critical for your BSP to function. The host must be running tftp in order for deployment to work. The following instructions are generic. Your system may be different and the commands should be adjusted accordingly.

1. Turn off firewall for tftp to work. `iptables -F` or type "setup" at the command line.
2. Install tftp-server.
3. Create the tftpboot directory and its sub-directory for MPC8308ERDB.  

```
mkdir /tftpboot  
mkdir /tftpboot/mpc8308
```
4. Copy over kernel, bootloader, DTB, Vitesse and filesystem images for your deployment to the `/tftpboot/mpc8308` directory:  

```
cp <install_path>/ltib-mpc8308erdb-xx /rootfs/boot/* /tftpboot/mpc8308  
cp <install_path>/ltib-mpc8308erdb-xx /rootfs.ext2.gz.uboot /tftpboot/mpc8308
```

Or  

```
cp /mnt/cdrom/Images/* /tftpboot/mpc8308
```
5. Connect board to the network.
6. Connect the target to the host via a serial connection.
7. Start `minicom` and set it up to talk to the MPC8315E RDB board:
  - Serial Setup: Select correct serial device; Hardware & Software Flow control = No; Bps=115200
  - Modem & dialing: Delete text for the following: Init String, Reset String, Hang-up String, No flow control
8. Power on board and see the console prompt.

## 4.2 Flash U-Boot

The flash range is from 0xFE000000 to 0xFE7FFFFFFF: 8MB.

The U-Boot image should be programmed into Flash, starting from 0xFE000000, by CodeWarrior for PPC version 8.8 + PowerTAP Pro or USB TAP.

An alternative way is to use the U-Boot commands for programming with the precondition that there is already a workable U-Boot programmed in the Flash:

```
=>tftp 2000000 u-boot.bin
=>protect off all
=>erase fe000000 fe0ffffff
=>cp.b 2000000 fe000000 $filesize
```

Then reset the board to boot it up.

## 4.3 Configure U-Boot

### 4.3.1 Configure U-Boot for Ramdisk Deployment

The images generated by LTIB allow you perform ramdisk deployment. Before performing ramdisk deployment, it is needed to configure U-Boot parameters.

At u-boot prompt, set u-boot environment like:

```
=>setenv ipaddr <board_ipaddress>
=>setenv serverip <tftp_serverip>
=>setenv gatewayip <your_gatewayip>

=>setenv tftp_path mpc8308
=>setenv ramdiskaddr 2400000
=>setenv loadaddr 2000000
=>setenv kernaddr 2000000
=>setenv fdtaddr 1f00000
=>setenv ramdisksize 90000
=>setenv dtbname mpc8308erdb.dtb
=>setenv hostname PowerQUICC

=>setenv flashkernaddr fe0a0000
=>setenv flashkernend fe2ffffff
=>setenv flashramdiskaddr fe300000
=>setenv flashramdiskend fe7effff
=>setenv flashfdtaddr fe7f0000
=>setenv flashfdtend fe7f3fff
=>setenv fdtsize 3000
```

```

=>setenv vscfwaddr fe7fe000
=>setenv vscfwend fe7fffff

=>setenv setargs 'setenv bootfile $tftp_path/uImage;setenv
fdtfile $tftp_path/$dtbname;setenv ramdiskfile
$tftp_path/rootfs.ext2.gz.uboot'

=>setenv setramargs 'setenv bootargs root=/dev/ram
ramdisk_size=$ramdisksize rw
ip=$ipaddr:$serverip:$gatewayip:$netmask:$hostname:$netdev:off
console=$consoledev,$baudrate $othbootargs'

=>setenv upvscfw 'tftp $loadaddr $tftp_path/vsc2bin; erase
$vscfwaddr $vscfwend; cp.b $loadaddr $vscfwaddr $filesize'

=>setenv upuboot 'setenv bootargs root=/dev/ram rw
console=$console,$baudrate;tftp $loadaddr mpc8308/u-
boot.bin;protect off all;erase fe000000 fe05ffff;cp.b $loadaddr
fe000000 $filesize'

=>setenv upkernel 'run setargs;run setramargs;tftp $loadaddr
$bootfile;erase $flashkernaddr $flashkernend;cp.b $loadaddr
$flashkernaddr $filesize;tftp $loadaddr $fdtfile;erase
$flashfdtaddr $flashfdtend;cp.b $loadaddr $flashfdtaddr
$filesize'

=>setenv upramdisk 'tftp $ramdiskaddr $ramdiskfile;erase
$flashramdiskaddr $flashramdiskend;cp.b $ramdiskaddr
$flashramdiskaddr $filesize'

=>setenv tftpramboot 'run setargs;run setramargs;tftp
$ramdiskaddr $ramdiskfile;tftp $kernaddr $bootfile;tftp $fdtaddr
$fdtfile;bootm $kernaddr $ramdiskaddr $fdtaddr'

=>setenv flashramboot 'run setargs;run setramargs;cp.b
$flashfdtaddr $fdtaddr $fdtsize;bootm $flashkernaddr
$flashramdiskaddr $fdtaddr'

=>saveenv

```

Now the U-Boot is ready for performing ramdisk deployment.

## 4.4 Flash Vitesse Binary

The Vitesse binary is stored in the flash at 0xFE7FE000 to 0xFE7FFFFFF. Program the binary to flash using the tftp U-Boot command. Set up the U-Boot environment as described above.

```
=>run upvscfw
```

## 4.5 Flash Kernel and Ramdisk

Set up the U-Boot environment as described above.

To flash the kernel and DTB:

```
=>run upkernel
```

To flash the ramdisk:

```
=>run upramdisk
```

## 4.6 Deployment

### 4.6.1 Deploy Ramdisk from TFTP

#### 1. Set U-Boot environment

Please set U-Boot environment as described above.

#### 2. Load images from TFTP

Tftp images to the board, then boot it up.

```
=>tftp 2000000 mpc8308/uImage
```

```
=>tftp 2400000 mpc8308/rootfs.ext2.gz.uboot
```

```
=>tftp 1f00000 mpc8308/mpc8308erdb.dtb
```

#### 3. Boot up board

```
=>bootm 2000000 2400000 1f00000
```

OR using a single command to do all the above:

```
=>run tftpramboot
```

## 4.6.2 Deploy Ramdisk from Flash

The kernel, DTB and ramdisk are programmed into flash as described in section 4.5.

To deploy from flash:

```
=>run flashramboot
```