

---

# MPC832XE-RDB BSP

User's Guide

Rev. 1.5

03/2008



## ***How to Reach Us:***

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **E-mail:**

[support@freescale.com](mailto:support@freescale.com)

### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor  
Technical Information Center, CH370  
1300 N. Alma School Road  
Chandler, Arizona 85224  
+1-800-521-6274 or +1-480-768-2130  
[support@freescale.com](mailto:support@freescale.com)

### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH  
Technical Information Center  
Schatzbogen 7  
81829 Muenchen, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[support@freescale.com](mailto:support@freescale.com)

### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku,  
Tokyo 153-0064, Japan  
0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor Hong Kong Ltd.  
Technical Information Center  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T., Hong Kong  
+800 2666 8080  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

### **For Literature Requests Only:**

Freescale Semiconductor Literature Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
1-800-441-2447 or 303-675-2140  
Fax: 303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document. Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

**Learn More:** For more information about Freescale products, please visit [www.freescale.com](http://www.freescale.com).

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2008. All rights reserved.

---

# Contents

<b>About This Manual .....</b>	<b>vi</b>
Audience .....	vi
Organization.....	vi
Conventions .....	vi
Definitions, Acronyms, and Abbreviations .....	vi
<b>Chapter 1 Introduction.....</b>	<b>1-1</b>
1.1    LTIB Overview.....	1-1
1.2    BSP Overview .....	1-1
<b>Chapter 2 LTIB Basics .....</b>	<b>2-2</b>
2.1    Installing the BSP .....	2-2
2.2    Running LTIB .....	2-2
<b>Chapter 3 Target Configuration .....</b>	<b>3-1</b>
3.1    Supported Target Revisions .....	3-1
3.2    Target System Memory Map.....	3-1
3.3    Target Set-up .....	3-1
<b>Chapter 4 Target Deployment .....</b>	<b>4-1</b>
4.1    Host Set-up .....	4-1
4.2    Flashing U-Boot .....	4-2
4.2.1    Programming U-Boot using CodeWarrior 8.7 and PowerTAP Pro .....	4-2
4.2.2    Programming U-Boot using U-Boot .....	4-2
4.3    Uploading Firmware Using U-Boot .....	4-3
4.4    Configuring U-Boot.....	4-3
4.4.1    NFS file system.....	4-3
4.4.2    Ramdisk file system .....	4-3

---

4.4.3	JFFS2 file system .....	4-4
4.5	NFS Depoyment .....	4-4
4.6	Ramdisk Deployment .....	4-5
4.7	JFFS2 Deployment .....	4-6



---

## 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 MPC832XE-RDB board.
- Boot Linux on the MPC832XE-RDB board.

## Audience

This document is addressed to developers who want to take advantage of the Freescale Linux Target Image Builder (LTIB) for the MPC832XE-RDB Board Support Package (BSP).

## Organization

This document is organized into xxx chapters.

Chapter 1	Provides an introduction to the MPC832XE-RDB BSP.
Chapter 2	Provides basic information on LTIB
Chapter 3	Provides important target set-up information
Chapter 4	Provides host and target-specific build and deployment information

## Conventions

This document uses the following notational conventions:

- Courier monospaced type indicates commands, command parameters, code examples, expressions, data types, and directives.
- Italic type indicates replaceable command parameters.
- All source code examples are in C.

## Definitions, Acronyms, and Abbreviations

The following list defines the abbreviations used in this document.

ATM	Asynchronous Transfer Mode
AAL	ATM Adaption Layer
APC	ATM Pace Control
BCSR	Board Control and Status Register
BSP	Board Support Package
COP	Common On-chip Processor

---

CSB	Coherency System Bus
CW	CodeWarrior IDE for PowerPC
DDR	Double Data Rate RAM
DIP	Dual-In-Line Package
DMA	Direct Memory Access
FLASH	Non-volatile reprogrammable memory
GPCM	General Purpose Chip-select Machine
GPL	General Purpose Line
HRCW	Hardware Reset Configuration Word
HS	High Speed
LBC	Local Bus Controller
LB	Local Bus
LSB	Least Significant Bit
LS	Low Speed
LTIB	Linux Target Image Builder
MII	Media Independent Interface
NFS	Network File System
OCP	On Chip Peripherals
OTG	On-The-Go
PCI	Peripheral Components Interconnect
PCS	Platform Creation Suite
PB	83xx Processor Board
PIB	Platform I/O board- expands the MDS functionality
PMC-PCI	PMCPPI extension card to provide PCI slot
QE	83xx Quicc Engine
RTC	Real Time Clock
RCWL	Reset Configuration Word Low
RCWH	Reset Configuration Word High
RMII	Reduced Media Independent Interface
SDRAM	Synchronous Dynamic Random Access Memory
SEC	Security Engine
TBD	To Be Defined
TFTP	Trivial File Transfer Protocol
UCC	Universal Communication Controllers in QE
UEC	UCC Ethernet Controller
UTOPIA	Universal Test and Operation Physical Interface for ATM





---

# Chapter 1

## Introduction

### 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

**NOTE:** Be aware that some host side packages may not function properly on every Linux distribution. The following are platforms where LTIB was tested.

- Redhat: 7.3, 8.0, 9.0
- Fedora Core: 1, 2, 3
- Debian: 3.1r0 (stable), unstable
- SuSE: 8.2, 9.2, 10.0

### 1.2 BSP Overview

This MPC832XE-RDB 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 2.6.24 kernel

- DUART support for console control
- Local bus support
- I2C bus driver and device probe support
- IPIC support
- Quicc Engine framework support
- UCC2 and UCC3 working as ethernet port, 10M/100M bps support with MII mode
- PCI host support with 33/66MHz

- 
- JFFS2/NFS/Ramdisk file system support

#### U-Boot 1.3.2

- Enable MPC8323E rev1.1
- Download QE uCode patch
- Local bus support
- I2C bus
- DDR2 memory fixed initialize
- DUART
- UCC2 and UCC3 ethernet support with both MII and RMII mode
- Flash enable
- Reset command
- Bootup from both BCSR and Flash

gcc-4.2.82-eglibc-2.5.82 for 832x,

binutils-2.17

Codewarrior 8.8 version

Documentation. See [START\\_HERE.html](#) on this CD.



---

## Chapter 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 <target-bsp.iso> <mount point>
```
2. As a non-root user, install the LTIB:  

```
<mount point>/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` 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 -preconfig config/platform/mpc832x_rdb/swang.preconfig
```

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. Once basic packages are installed, configuration window with preset packages will pop-up. Save and exit the configuration window for ltib to continue building packages.

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

- `<install path>/rootfs` – directory, the root file system that will be deployed on your board.
- `<install path>/rootfs.ext2.gz.uboot` – ramdisk file system that can be flashed to your board.
- `<install path>/rootfs.jffs2` – JFFS2 file system that can be flashed to your board.
- `<install path>/rootfs/uImage` – kernel image that can be loaded with uboot.

---

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 --preconfig config/platform/mpc832x_rdb/swang.preconfig
```

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



---

## Chapter 3

# Target Configuration

### 3.1 Supported Target Revisions

The target system is the MPC832XE-RDB board. This BSP is known to work on the following board(s) revision(s):

PB: MPC832XE-RDB Prototype Rev1.0

### 3.2 Target System Memory Map

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

Start	End	Definition
0x00000000	0x04000000	DDR2 SDRAM
0x80000000	0x9fffffff	PCI memory
0xe0000000	0xe01fffff	IMMR
0xe2000000	0xe3000000	PCI I/O
0xf8000000	0xf8007fff	BCSR
0xf8008000	0xf800ffff	to PIB cs2
0xf8010000	0xf8017fff	to PIB cs3
0xfe000000	0xfeffffff	Flash

The flash starts at address 0xfe000000.

Start	End	Definition
0xfe000000	0xfe01ffff	HRCW
0xfe100000	0xfe8fffff	JFFS2 file system
0xfe900000	0xfecfffff	Ramdisk file system
0xfed00000	0xfeefffff	Linux kernel
0xfef00000	0xfef3ffff	U-Boot
0xfef40000	0xfefffffff	U-Boot environment

### 3.3 Target Set-up

1. Connect your board to the network via the ENET1 port.
2. Connect your board to your host machine via the serial port.  
Setup the terminal, baudrate 115200bps, 8-N-1, no flow control.
3. Connect your board to power supply, power on your board.
4. The U-boot terminal should show clock and memory setting, make sure it is correct.

Table 3-3 MPC832XE-RDB-PB clock and memory settings

Clock	Value(MHz)
CSB	133
Core	333
QE	200
LBC	133
LB	66
DDR	266
SEC	133
I2C	133
Memory	Value(MB)
DDR2	64
FLASH	16



---

# Chapter 4

## Target Deployment

### 4.1 Host Set-up

Host setup is critical for your BSP to function. The host must be running tftp and nfs 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. Install nfs-server

4. Create the tftboot directory.

```
mkdir /tftboot
```

5. Link rootfs to an exportable directory once you have built your project.

```
ln -s <install_path>/ltib/rootfs /tftboot/ltib
```

6. Copy over kernel, bootloader, and flash filesystem images for your deployment to the /tftboot directory

```
cp <install_path>/ltib/rootfs/boot/* /tftboot
cp <install_path>/ltib/<flashfs> /tftboot
cp <cd mount point>/bootloaders/* /tftboot
```

7. Edit /etc/exports and add the following line:

```
/tftboot/ltib/ <target board IP>(rw,no_root_squash, async)
```

8. Edit /etc/xinetd.d/tftp to enable tftp like this:

```
{
disable      = no
socket_type  = dgram
protocol     = udp
wait        = yes
user         = root
server       = /usr/sbin/in.tftpd
server_args  = /tftboot
}
```

9. Restart the nfs and tftp servers on your host

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

---

```
/etc/init.d/nfsserver restart
```

10. Connect board to the network .
11. Connect the target to the host via a serial connection.
12. Start `minicom` and set it up to talk to the MPC832XE-RDB-PB board
  - Serial Setup: Select correct serial device; Hardware & Software Flow control = No; bps = 115,200
  - Modem & dialing: Delete text for the following: Init String, Reset String, Hang-up String, No flow control
13. Power on board and see the console prompt.

## 4.2 Flashing U-Boot

We recommend you flash the u-boot located on your CD in `/images/u-boot.bin` or `<install_path>/ltib/rootfs/boot/u-boot.bin`

### 4.2.1 Programming U-Boot using CodeWarrior 8.7 and PowerTAP Pro

1. Plug PowerTAP Pro into JTAG port of the board.
2. Connect PowerTAP Pro to network via cable.
3. Make sure the switches and jumpers on board is correct.
4. Power on the board and PowerTAP Pro.
5. Open Codewarrior and new a 8325 project with wizard.
6. Click Tools->Flash Programmer
7. Make sure target processor and connection is correct.
8. Erase flash from 0xffff00000 to 0xffffffff.
9. Select u-boot.bin file to flash, and apply address offset is 0xffff00000, click program to flash, wait for one minute. Flash will be finished.

### 4.2.2 Programming U-Boot using U-Boot

If you want u-boot to upgrade your u-boot, it can work. You need download the u-boot image to ddr2 ram on the board via tftp. See the instructions as below.

```
=> tftp 200000 /tftpboot/u-boot.bin  
  
=> protect off all
```

```
=> erase fef00000 fef3ffff
=> cp.b 200000 fef00000 <u-boot size in hex>
=> protect on all
```

## 4.3 Uploading Firmware Using U-Boot

From Production Release RC-2 onwards, Quicc Engine firmware upload procedure has changed. Before Uploading Quicc Engine Firmware, ensure that you are running u-boot-1.3.2 or above. Also firmware uploaded via u-boot is supported by linux-2.6.25 or above although a patch is applied on Linux-2.6.24 to make it use firmware uploaded via u-boot.

There is a new package added in Ltib i.e. “DPA Firmware” under dpa\_firmware-x.y.z.tar.bz2. where x,y,z can be seen from release notes provided with this BSP. Firmware installation details are also provided in same readme.

## 4.4 Configuring U-Boot

### 4.4.1 NFS file system

```
=>setenv ipaddr 10.193.20.177
=>setenv ethaddr 00:00:01:12:23:01
=>setenv serverip 10.193.20.88
=>setenv gatewayip 10.193.20.254

=>setenv bootargs root=/dev/nfs rw
nfsroot=10.193.20.88:/tftpboot/10.193.20.177
ip=10.193.20.177:10.193.20.88:10.193.20.254:255.255.255.0:mpc832xe:eth0:off
console=ttyS0,115200

=>saveenv
```

### 4.4.2 Ramdisk file system

```
=>setenv ipaddr 10.193.20.177
=>setenv ethaddr 00:00:01:12:23:01
=>setenv serverip 10.193.20.88
=>setenv gatewayip 10.193.20.254

=>setenv bootargs root=/dev/ram console=ttyS0,115200
ramdisk_size=4000000
```

---

```
=>saveenv
```

### 4.4.3 JFFS2 file system

```
=>setenv ipaddr 10.193.20.177

=>setenv ethaddr 00:00:01:12:23:01

=>setenv serverip 10.193.20.88

=>setenv gatewayip 10.193.20.254

=>setenv bootargs root=/dev/mtdblock1 rootfstype=jffs2 rw
console=ttyS0,115200

=>saveenv
```

## 4.5 NFS Deployment

1. Copy the kernel image from `<install_path>/ltib/rootfs/boot/uImage` to the `/tftpboot` directory created during host setup.
2. At the `uboot` prompt, use `printenv` to ensure target IP address, tftp server IP, and MAC address are set properly.

```
=>printenv -- Lists u-boot environment settings
```

3. If the settings are not correct, use `setenv` to set them, or type `help` at the `uboot` prompt for other options.
4. Make sure u-boot configuration is okay, see section 4.3.1.
5. Download the Linux kernel binary to ddr2 ram.

```
=>tftp 200000 uImage
```

6. Download the dts file

```
=>tftp 400000 mpc832x_rdb.dtb
```

7. To boot linux, issue the following `uboot` command.

```
=>bootm 200000 - 4000000
```

## 4.6 Ramdisk Deployment

1. Copy the kernel image from `<install_path>/ltib/rootfs/boot/uImage` to the `/tftpboot` directory created during host setup.
2. Copy the ramdisk image from `<install_path>/ltib/rootfs.ext2.gz.uboot` to the `/tftpboot` directory created during host setup.
3. At the `uboot` prompt, use `printenv` to ensure target IP address, tftp server IP, and MAC address are set properly.

```
=>printenv -- Lists u-boot environment settings
```

4. If the settings are not correct, use `setenv` to set them, or type `help` at the `uboot` prompt for other options.
5. Make sure u-boot configuration is okay, see section 4.3.2.

6. Download the Linux kernel binary to ddr2 ram.

```
=>tftp 100000 uImage
```

7. Download the Ramdisk file system image to ddr2 ram.

```
=>tftp 300000 rootfs.ext2.gz.uboot
```

8. Burn the ramdisk image to flash.

```
=>protect off all
```

```
=>erase fe900000 fecffffff
```

```
=>cp.b 300000 fe900000 <ramdisk size in hex>
```

```
=>protect on all
```

9. Burn the Linux kernel to flash.

```
=>protect off all
```

```
=>erase fed00000 feffffff
```

```
=>cp.b 100000 fed00000 <kernel size in hex>
```

```
=>protect on all
```

10. Download the Device Tree file to ddr2 ram.

```
=>tftp 100000 mpc832x_rdb.dtb
```

11. To boot linux, issue the following `uboot` command

```
=>bootm fed00000 fe900000 100000
```

## 4.7 JFFS2 Deployment

1. Copy the kernel image from `<install_path>/ltib/rootfs/boot/uImage` to the `/tftpboot` directory created during host setup.
2. Copy the JFFS2 image from `<install_path>/ltib/rootfs.jffs2` to the `/tftpboot` directory created during host setup.
3. At the `uboot` prompt, use `printenv` to ensure target IP address, tftp server IP, and MAC address are set properly.

```
=>printenv -- Lists u-boot environment settings
```

4. If the settings are not correct, use `setenv` to set them, or type `help` at the `uboot` prompt for other options.
5. Make sure u-boot configuration is okay, see section 4.3.3.
6. Download the Linux kernel binary to DDR2 RAM using the following:

```
=>tftp 200000 uImage
```

7. Download the JFFS2 file system image to ddr2 ram using the following:

```
=>tftp 400000 rootfs.jffs2
```

8. Burn the JFFS2 image to flash.

```
=>protect off all
```

```
=>erase fe100000 fe8fffff
```

```
=>cp.b 400000 fe020000 <JFFS2 size in hex>
```

```
=>protect on all
```

9. Burn the Linux kernel to flash.

```
=>protect off all
```

```
=>erase fed00000 feffffff
```

```
=>cp.b 200000 fed00000 <kernel size in hex>
```

```
=>protect on all
```

---

12. Download the Device Tree file to ddr2 ram.

```
=>tftp 100000 mpc832x_rdb.dtb
```

10. To boot linux, issue the following uboot command

```
=>bootm fed00000 fe100000 100000
```