MPC8349EA-MDS BSP

User's Guide

Rev. 1.1

12/2006

freescale semiconductor

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
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

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

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

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

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

Contents

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 MPC8349EA-MDS board.
- Boot Linux on the MPC8349EA-MDS board.

Audience

This document is addressed to developers who want to take advantage of the Freescale Linux Target Image Builder (LTIB) for the MPC8349EA-MDS Board Support Package (BSP).

Organization

This document is organized into 4 chapters.

Chapter 1	Provides an introduction to the MPC8349EA-MDS 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.

ADS	Application Development System
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
MDS	MPC83xx family application Development System
NFS	Network File System
OCP	On Chip Peripherals
OTG	On-The-Go
PCI	Peripheral Components Interconnect
PCS	Platform Creation Suite 3.0.1 version
PB	83xx Processor Board
PIB	Platform I/O board- expands the MDS functionality
PMC-PCI	PMC PCI extension card to provide PCI slot
QUICCstart	also means MPC83xx MDS board
RTC	Real Time Clock
RCWL	Reset Configuration Word Low
RCWH	Reset Configuration Word High
RMII	Reduced Media Independent Interface
ROOTFS	Root filesystem
SDRAM	Synchronous Dynamic Random Access Memory
SEC	Security Engine
SYS	Previous name of MPC83xx PB board
TBD	To Be Defined
TFTP	Trivial File Transfer Protocol

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 MPC8349EA-MDS 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 2.6.11 kernel

- DUART support for console control
- Local bus support
- I2C bus driver and device probe support
- IPIC support
- TSEC1 and TSEC2 working as ethernet port
- PCI host support with 33/66MHz
- JFFS2/NFS/Ramdisk file system support

U-Boot 1.1.3

- Enable MPC8349EA
- Local bus support
- I2C bus
- DDR2 memory initialize
- DUART
- TSEC1 working as TFTP port
- Flash enable
- Reset command
- Bootup from both BCSR and Flash

Peripheral Device Driver

- SPI driver and testing code working with on board Serial Flash Memory (M25P40)
- MPC8349EA software watchdog driver
- PCI Host driver working 66MHz/64Bit, 66MHz/32Bit or 33MHz/32Bit
- PCI agent driver working at 66MHz/32bit
- MPC8349EA security engine SEC2.x version 1.5 driver and test working in kernel space
- MPC8349EA PB USB host support USB2.0 host mode
- MPC8349EA PB USB gadget support USB2.0 slave mode
- MPC8349EA PB USB OTG driver
- MPC8349EA USB-PMC Card driver
- CodeTEST SWIC and HWIC softweare ananlysis demo support
- CodeWarrior debug for the Linux kernel.

gcc-3.4.3-e300-glibc-2.3.3 for 834x, binutils-2.15

Codewarrior 8.7 version

Documentation. See START_HERE.html on this CD.

Chapter 2
LTIB Basic

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
```

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.

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 settings 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 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*>/vmlinux.gz.uboot – 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 litb

   ```
   ./ltib
   ```

More information on LITB 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 MPC8349EA-MDS board. This BSP is known to work on the following board(s) revision(s):

PIB: Pilot
PB: MPC8349EA PB Production Pilot
PMC-PCI: PMC to PCI extension board in PCI slot on PIB

3.2 Target System Memory Map

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

Start	End	Definition
0x00000000	0x07ffffff	DDR2 SDRAM
0x80000000	0xBfffffff	PCI1&2 memory
0xe0000000	0xe00fffff	IMMR
0xf0000000	0xf3000000	local bus SDRAM
0xf8000000	0xf8007fff	BCSR
0xfe000000	0xfeffffff	Flash

The flash starts at address **0xfe000000**.

Start	End	Definition
0xfe000000	0xfe01ffff	HRCW
0xfe020000	0xfe0fffff	reserved for BIT test image
0xfe100000	0xfe8fffff	JFFS2 file system
0xfe900000	0xfecfffff	Ramdisk file system
0xfed00000	0xffcfffff	Unused
0xffd00000	0xffefffff	Linux kernel
0xfff00000	0xfff3ffff	U-Boot
0xfff40000	0xffffffff	U-Boot environment

3.3 Target Set-up

1. Plug the MPC8349EA-MDS-PB and PMC-PCI board into PIB baseboard.

2. Connect your board to the network via the TSEC1 port(J1).

3. Connect your board to your host machine via the serial port(P2).

4. Setup the terminal, baudrate 115200bps, 8-N-1, no flow control.
5. Verify the dip switches and jumpers are set correctly. The dip switches and jumpers on MPC8349EA-MDS-PB board need to be setup properly in order to program and run u-boot from flash. In the tables below are the jumper and switch settings for the BSP. Please see the Hardware Getting Started Guide on the BSP ISO image for meanings of the jumpers and switches. Table 3-1, 0-stand for ON, 1-stand for OFF.

Table 3-1 MPC8349EA-MDS-PB Board DIP Switches Settings for DDR at 266M

Switch	1	2	3	4	5	6	7	8
SW3	0	0	0	0	0	1	0	0
SW4	1	0	1	0	1	0	0	0
SW6	0	0	1	1	0	0	0	0
SW7	0	0	0	0	1	0	0	0

Table 3-2 MPC8349EA-MDS-PB Board DIP Switches Settings for DDR at 333M

Switch	1	2	3	4	5	6	7	8
SW3	0	0	0	0	0	1	0	1
SW4	1	0	1	0	1	0	0	0
SW6	0	0	1	1	0	0	0	0
SW7	0	1	0	0	0	1	1	0

Table 3-3 MPC8349EA-MDS-PB Board Jumper Settings

Jumper	Setting
JP1	1-2
JP2	open
JP3	mounted

6. Connect your board to power supply, power on your board.

7. The U-boot terminal should show clock and memory setting, make sure it is correct.

Table 3-4 MPC8349EA-MDS-PB clock and memory settings

Clock	Value(MHz)	Value(MHz)
CSB	266	333
Core	528	500
LBC	266	333
LB	66	88
DDR	266	333
SEC	88	110
I2C1&2	266	333
TSEC1&2	266	333
USB MPH&DR	88	110
Memory	Value(MB)	Value(MB)
DDR	256	256
FLASH	32	32

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 /tftpboot`
5. Link `rootfs` to an exportable directory once you have built your project.
   `ln –s <install_path>/ltib/rootfs /tftpboot/ltib`
6. Copy over kernel, bootloader, and flash filesystem images for your deployment to the `/tftpboot` directory

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

7. Edit `/etc/exports` and add the following line:
   `/tftpboot/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 = /tftpboot
    }
   ```

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 MPC8349EA-MDS-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 8349e project with wizard.
6. Click *Tools->Flash Programmer*
7. Click load settings to select *MPC8349EA_MDS_init.cfg* attached in BSP iso package.
8. Make sure target processor and connection is correct.
9. Erase flash from *0xfe000000* to *0xffffffff*.
10. Select *u-boot.bin* file to flash, and apply address offset is *0xfff00000*, click program to flash, wait for one minute. Flash will be completed.

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 RAM on the board via tftp. See the instructions as below.

```
=> tftp 400000 /tftpboot/u-boot.bin

=> protect off all

=> erase fff00000 fff3ffff

=> cp.b 400000 fff00000 <u-boot size in hex>

=> protect on all
```

4.3 Configuring U-Boot

4.3.1 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:MPC8349EA:eth0:of
f console=ttyS0,115200

=>saveenv
```

4.3.2 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

=>saveenv
```

4.3.3 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
```

4.4 NFS Depoyment

1. Copy the kernel image from <*install_path*>/ltib/vmlinux.gz.uboot to the /tftpboot directory created during host setup.

2. At the u-boot prompt, use printenv to ensure target IP address, tftp server IP, and MAC address are set properly.

   =>printenv -- Lists u-boot enviroment 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 the RAM.

   ```
   =>tftp 400000 /tftpboot/vmlinux.gz.uboot
   ```

6. To boot linux, issue the following uboot command.

   ```
   =>bootm 400000
   ```

4.5 Ramdisk Deployment

1. Copy the kernel image from <*install_path*>/ltib/vmlinux.gz.uboot 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 enviroment 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 RAM.

   ```
   =>tftp 200000 /tftpboot/vmlinux.gz.uboot
   ```

7. Burn the Linux kernel to flash.

   ```
   =>protect off all

   =>erase ffd00000 ffefffff

   =>cp.b 200000 ffd00000 <kernel size in hex>

   =>protect on all
   ```

8. Download the Ramdisk file system image to RAM.

   ```
   =>tftp 400000 /tftpboot/rootfs.ext2.gz.uboot
   ```

9. Burn the ramdisk image to flash.

   ```
   =>protect off all

   =>erase fe900000 fecfffff

   =>cp.b 400000 fe900000 <ramdisk size in hex>

   =>protect on all
   ```

10. To boot linux, issue the following uboot command

    `=>bootm 200000 fe900000` or

    `=>bootm ffd00000 fe900000` ---production

4.6 JFFS2 Deployment

1. Copy the kernel image from `<install_path>/ltib/vmlinux.gz.uboot` 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 enviroment settings

4. If the settings are not correct, use `setenv` to set them, or type help at the u`boot` prompt for other options.

5. Make sure u-boot configuration is okay, see section 4.3.3.

6. Download the Linux kernel binary to RAM using the following:

```
=>tftp 200000 /tftpboot/vmlinux.gz.uboot
```

7. Burn the Linux kernel to flash.

```
=>protect off all

=>erase ffd00000 ffefffff

=>cp.b 200000 ffd00000 <kernel size in hex>

=>protect on all
```

8. Download the JFFS2 file system image to RAM using the following:

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

9. Burn the JFFS2 image to flash.

```
=>protect off all

=>erase fe100000 fe8fffff

=>cp.b 400000 fe100000 <JFFS2 size in hex>

=>protect on all
```

10. To boot linux, issue the following uboot command

```
=>bootm 200000 or

=>bootm ffd00000 ---production
```