Linux BSP for the Freescale i.MX31ADS

User's Guide

Rev. 4.6

05/2007

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

Learn More: For more information about Freescale products, please visit www.freescale.com.

Contents

About This Book

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 i.MX31ADS board.
- Boot Linux on the i.MX31ADS board.

Audience

This document is addressed to developers who want to take advantage of the Freescale Linux Target Image Builder (LTIB) for the i.MX31ADS Board Support Package (BSP).

Organization

This document is organized into four chapters.

Chapter 1	Provides an introduction to the i.MX31ADS 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.

ATA	Advanced Technology Attachment
BSP	Board Support Package
DPM	Dynamic Power Management
DPTC	Dynamic Process and Temperature Compensation
DVFS	Dynamic Voltage Frequency Scaling
LTIB	Linux Target Image Builder
NFS	Network File System
OSS	Open Sound System
RTC	Real Time Clock
TFTP	Trivial File Transfer Protocol
USB	Universal Serial Bus

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 gigabyte of free disk space
- 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 i.MX31ADS 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 i.MX31ADS system with additional features.

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

Kernel Features:

- Linux-2.6.19.2 kernel
  - UART --On chip
  - Framebuffer
  - Synchronous Display
  - Asynchronous Display
  - NEC VGA Panel
  - Keypad
  - TV-OUT VGA, NTSC, PAL

- Ethernet
- SPI
- Touchscreen
- iMagic IM8012 camera
- Mangachip MC521DA camera
- Video4Linux: Pre-Processing support, SDC, ADC, Camera
- I2C
- Video Post-Processing
- Video Pre-Processing
- ATA PIO 0-4 Mode, UDMA 0-3 Mode
- ALSA Audio with OSS emulation
- USB Host: Mass Storage
- USB Gadget: Mass Storage, Ethernet, HID devices
- USB OTG pin detect
- Power Management: Doze mode, backlight control, DPM, DPTC, DVFS
- Battery Gauge
- NAND Boot, MTD
- NOR Boot, MTD
- CodeTEST
- RTC
- Watchdog
- PCMCIA (untested)
- SD/SDHC/SDIO

Redboot version FSL 200712
NAND Boot

GNU ARM gcc 4.1.1, glibc 2.4, binutils-2.16, and
linux-libc-headers-2.6.16

Various open source user space packages including

- alsa-lib (audio library)
- alsa-utils (audio utilities)
- base-libs
- bash
- busybox
- coreutils
- device files
- devmem2 (utility to read/write memory)
- dhcp
- diffutils
- dropbear (lightweight ssh utility)
- e2fsprogs
- fbset (framebuffer utilities)
- findutils
- gdb (debugger)
- gstreamer (multimedia framework)
- glib2 (gnome library)
- hdparm
- hotplug
- irattach

- less
- libid3tag (mp3 id tag library)
- libjpeg (jpeg library)
- libmad (mp3 library)
- liboil (common library functions)
- libtermcap
- libusb (usb library)
- libxml2 (xml library)
- madplay (mp3 player)
- mxc-misc (sample/test code for mx processors)
- module-init-tools
- mtd-utils (flash utilities)
- ncurses
- netperf
- net-tools
- ntp-client
- openssl
- portmap
- procinfo
- procps
- psmisc
- Qtopia (GUI demo)
- strace (system trace utility)
- sysconfig (system config package)
- timezone (timezone package)
- tslib (touchscreen library)
- tvout-test-mx31
- udev (userspace device manager daemon)
- usbutils
- wireless-tools
- zlib (compression library)

Proprietary Packages:

GX200-BU: This package provides proprietary binary kernel modules, libraries, and test code built from the MBX OpenGL ES (GX200) DDK.

hantro-binary: This package provides libraries, header files and API documentation to assist with MPEG4 encoding and decoding.

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`

The BSP EULA will be presented for acceptance. To continue installing accept the licesnse. Then, 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
```

This will 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 directory/image files:

- **rootfs** – directory, the root file system that will be deployed on your board.
- **rootfs.jffs2** – JFFS2 filesystem that can be flashed to your board.
- **rootfs/boot/zImage** – kernel image that can be loaded with Redboot

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 i.MX31ADS board. This BSP is known to work on the following board revisions:

Baseboard: Rev 3.3
CPU Card: Rev 3.5
MC13783 Card: Rev B

This BSP only supports a configuration with the MC13783 card plugged-in.

3.2 Target Set-up

1. Plug the i.MX31 CPU card and the MC13783 card into the ADS Baseboard

2. Connect your board to the network via the Ethernet port.

3. Connect your board to your host machine via the serial port. For Linux, use the top serial port marked "UART C" next to the Ethernet jack.

4. Verify the dip switches and jumpers are set correctly. See the the applicable software docs on this CD in /help/software for jumper and switch settings for specific peripherals. See *HW Getting Started Guide* in the kit for more information on jumpers and switches.

   NOTE: Except when running TV-Out, set switch 4 of SW3 on your baseboard to ON.

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 and may require root permissions. 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 MX31ADS 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 Bootloader

We recommend you flash the Redboot located on your CD in `/bootloaders`.

To reprogram a Redboot image using an existing running Redboot image, load the Redboot image into SDRAM and then program it into flash. Be sure to setup your host as outlined in section 4.1. The Redboot commands are:

```
load -r -b 0x1000000 /tftpboot/mx31ads_redboot.bin
run (Stop autoboot, if set)
romupdate
```

Reset your board to use the new redboot.

The system configuration also needs to be set. This is done via the `fconfig` command. I used the following configuration for the i.MX31 ADS:

```
RedBoot> fconfig -l
Run script at boot: false
Use BOOTP for network configuration: true
Default server IP address: 10.81.4.170
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
```

NOTE 1: The new configuration is not used until you issue a reset

4.3 Development Deployment

1. Copy the kernel image from `<install_path>/ltib/rootfs/boot/zImage` to the `/tftpboot` directory created during host setup.

2. At the `redboot` prompt, use `fconfig` to ensure target IP address, tftp server IP, and MAC address are set properly.

   `fconfig -l` -- Lists redboot settings

3. If the settings are not correct use `fconfig` to set them, or type help at the `redboot` prompt for other options.

4. Download the Linux kernel binary to SDRAM using the following:

   ```
   load -r -b 0x100000 /tftpboot/zImage
   ```

5. To boot linux, issue the following Redboot command. `-l` indicates the size of the image; `-c` indicates the kernel command line. Be sure to modify the Ethernet settings for your network.

```
exec -b 0x100000 -l 0x200000 -c "noinitrd console=ttymxc0,115200
root=/dev/nfs nfsroot=172.27.180.19:/tftpboot/ltib/ init=/linuxrc
ip=172.27.210.2:172.27.180.19"
```

4.4 NOR Flash Deployment

1. Power off board and set the i.MX31ADS CPU Card Boot Mode Dip Switches to boot from NOR:

Switch	1	2	3	4	5
SW2	ON	ON	OFF	ON	OFF

2. Copy the kernel image from `<install_path>/ltib/rootfs/boot/zImage` to the `/tftpboot` directory created during host setup.

3. Copy the jffs2 filesystem image from `<install_path>/ltib` to the `/tftpboot` directory

4. At the `redboot` prompt, use `fconfig` to ensure target IP address, tftp server IP, and MAC address are set properly.

   `fconfig -l` -- Lists redboot settings

5. If the settings are not correct, use `fconfig` to set them, or type `help` at the `redboot` prompt for other options.

6. `fis init` is required to initialize the FLASH image system and only needs to be done the first time.

   ```
   fis init
   ```

7. Download the Linux kernel binary to SDRAM using the following:

   ```
   load -r -b 0x100000 /tftpboot/zImage
   ```

8. Flash the kernel

   ```
   fis create -f 0xa0100000 kernel
   ```

   Note: It is not necessary to program the kernel image into flash unless you want to boot the same image multiple times. But `fis create` still needs to be issued at least once in order to create the Redboot partition for the kernel. Otherwise the kernel may not boot.

9. Download the JFFS2 file system image to SDRAM.

   ```
   load -r -b 0x100000 /tftpboot/rootfs.jffs2
   ```

10. Flash the file system. This example assumes a kernel that is less than 2MB. Adjust your `-f` value based on your kernel size. Also, if your file system image is not padded, you may want to use the `-l` option to extend the length. To avoid stepping on existing Redboot partitions, use `fis list`.

    ```
    fis create -f 0xa0300000 root
    ```

    Note: The jffs2 can be mastered on the host build machine, pre-extended to fill the rootfs flash partition, and pre-warmed to eliminate the delay on first boot with a procedure similar to this one:

    ```
    rmmod mtdram
    modprobe mtdram total_size=29568 erase_size=128
    modprobe mtdblock
    modprobe mtdchar
    (cat /proc/mtd should show this new mtd device as mtd0)
    dd if=rootfs.jffs2 of=/dev/mtd0
    mkdir -p /mnt/tmp
    mount -t jffs2 /dev/mtdblock0 /mnt/tmp
    umount /mnt/tmp
    dd if=/dev/mtd0 of=rootfs-expanded-warm.jffs2
    ```

    Then deploy `rootfs-expanded-warm.jffs2` to the target as usual.

11. To boot Linux issue the following Redboot commands. `-l` indicates the size of the image; `-c` indicates the kernel command line. Be sure to modify the kernel command line to reflect your deploy.

    ```
    fis load kernel

    exec -b 0x100000 -l 0x200000 -c "noinitrd console=ttymxc0,115200
    root=/dev/mtdblock2 rootfstype=jffs2 init=linuxrc ip=none”
    ```

4.5 NAND Flash Deployment

12. Reconfigure and build the kernel with the correct kernel command line. For example,

    ```
    noinitrd console=ttymxc0,115200 root=/dev/mtdblock7 rootfstype=jffs2
    init=linuxrc ip=none
    ```

13. Copy the kernel image from `<install_path>/ltib/rootfs/boot/zImage` to `nand_boot/image/Image` to build the correct bootloader files. If you are using a JFFS2 file system, make sure it is configured for 16k erase block size.

14. Build the NAND bootloader files and copy them to the root filesystem. For build instructions see README.txt in /bootloaders/nandboot-mx-20070125.tar.gz on the CD image.

15. Boot the i.MX31 with an nfs mounted filesystem from NOR as outlined above.

16. Erase the NAND flash partitions. Use `flash_erasall --help` to see available options.

```
flash_eraseall /dev/mtd/5
flash_eraseall /dev/mtd/6
flash_eraseall /dev/mtd/7
```

To see the partitions use `cat /proc/mtd`. You should see something similar to this:

```
dev:    size   erasesize  name
mtd0: 00040000 00020000 "RedBoot"
mtd1: 00140000 00020000 "kernel"
mtd2: 018a0000 00020000 "root"
mtd3: 0001f000 00008000 "FIS directory"
mtd4: 00001000 00008000 "RedBoot config"
mtd5: 00004000 00004000 "IPL-SPL"
mtd6: 00400000 00004000 "Kernel"
mtd7: 01400000 00004000 "ROOTFS.CRAMFS"
mtd8: 067fc000 00004000 "JFFS2"
```

17. Copy the NAND bootloader, kernel, and filesystem onto the NAND flash card as follows. Use `nandwrite --help` to see available options.

```
nandwrite /dev/mtd/5 IMX31ADS_nb_iplspl.bin -p
nandwrite /dev/mtd/6 Image_nb_crc -p
nandwrite /dev/mtd/7 <filesystem> -p
```

18. Power off board and set the i.MX31ADS CPU Card Boot Mode Dip Switches to boot from NAND:

Switch	1	2	3	4	5
SW2	OFF	ON	ON	ON	OFF

19. Power on the board. You may have to press the reset button multiple times before the NAND boot menu appears.