i.MX53 START Linux
User's Guide

Document Number: 924-76373
Rev. 11.09.01
0 9/2011

freescale™
semiconductor

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

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 Book

This document explains how to build and install the Freescale Linux BSP on the i.MX53 START board. All steps needed to get the i.MX53 START board running are detailed, including board dip switch settings, steps to download an OS image through the manufacturing (MFG) tool, and instructions on configuring and using the u-Boot bootloader.

Audience

This document is intended for software, hardware, and system engineers who are planning to use the product and for anyone who wants to understand more about the product.

References

1. i.MX Family Linux Software Development Kit Reference Manual

1. Introduction

The i.MX53 START Linux BSP is a collection of binary, source code, and support files that can be used to create a Linux kernel image and a root file system for i.MX53 START board. This document is only for the general Linux platform. For the steps on how to run and configure a new Ubuntu rootfs please read the Ubuntu 10.04 Startup Guide for the i.MX5X board. For the steps on how deploy a new copy of the demo image provided with the boards please read the i.MX53_START_Linux_DemoImage_Readme.

Please note that the i.MX53 START board was formerly named i.MX53 LOCO board. This name change has not been completely propagated through the build environment, so instances of the name "LOCO" will still be seen throughout this document. This is an older name for the START board.

1.1 Boot Loader

The i.MX53 START Linux delivery package contains the following U-Boot bootloader binary:

```
<Version>_images_MX5X/u-boot-mx53-loco.bin
```

This bootloader supports Micro SD boot for MX53 START board.

1.2 Linux Kernel image

This Freescale i.MX BSP contains a pre-built kernel image based on the 2.6.35.3 version of the Linux kernel. The i.MX53 START kernel image is found at the following location:

```
<Version>_images_MX5X/uImage
```

1.3 Ubuntu demo rootfs

An Ubuntu demo rootfs (lucid version) with demo applications is provided for demo purpose.

2. Building the Linux Platform

This chapter explains how to set up the build environment, install and build LTIB, set the rootfs for NFS, and set up the host environment.

2.1 Setting Up the Linux Host

See "`ltib_build_host_setup.pdf`" to setup the Linux host server.

2.2 Installing and Building LTIB

To install and build LTIB, follow the steps below:

NOTE

In some Linux systems, the following procedure must be done with **root** permissions. However, these instructions are for performing the procedure "not as root".

To run LTIB, some host packages are needed. If any error related to a host package is raised, install the host package.

1. Remove all previously-installed packages in `/opt/freescale/pkgs/`.
2. Install the LTIB package not as root:

```
tar zxf <ltib_release>.tar.gz
./<ltib_release>/install
```

3. Install the patches located in the patches.tar.gz file.

This command installs LTIB to your directory.

4. Build LTIB:

```
cd <LTIB directory>
./ltib -m config
```

5. Select platform to **Freescale iMX reference boards** and exit, saving the changes. At the next menu, select platform type as **imx5x** and package profile. Exit and save changes. Please note that only the profiles of **Min profile**, **FSL gnome release packages** and **mfg firmware profile** pass build tests.

6. To build U-Boot for MX53 START board, select “Choose your board for u-boot” as “mx53_loco”. Please note this option is only for U-Boot. For kernel image, current default kernel configuration can build the same images for all i.MX5x parts boards.

```
|--- Choose your board                                                    |
                  |
  | |    board (mx53_loco)  --->
```

7. Run the following command:

```
./ltib
```

When this procedure is completed, the kernel image and the U-boot images are located at:

```
rootfs/boot/
```

8. Input the following command to get LTIB command help:

```
./ltib -help
/* Get the source code of one package */
./ltib -m prep -p <package name>
/* Build one package */
./ltib -m scbuild -p <package name>
/* Install one package to rootfs */
./ltib -m scdeploy -p <package name>
```

NOTE

If your system is not using a touchscreen as a primary device you should remove touchcreens from the kernel configuration.

- Run ./ltib -m config
- Select Configure the kernel
- Run ./ltib again
- Select Device drivers
- Select Input device support
- Deselect touchscreens

2.3 Setting rootfs for NFS

There are two ways to set up the rootfs for NFS on this package.

- Using the ext2 format rootfs package provided in the distribution
- Using the rootfs that is created after making the build of the kernel

Use the following commands to set the rootfs directory for NFS using the rootfs.ext2.gz package already included in the distribution (you must be the root user for this operation):

```
mkdir /mnt/rootfs
cp imx5x/rootfs.ext2.gz  /tools
cd /tools
gunzip rootfs.ext2.gz
mount -o loop -t ext2 rootfs.ext2 /mnt/rootfs
cp -r /mnt/rootfs .
export ROOTFS_DIR=/tools/rootfs
```

NOTE

In some Linux distributions (such as Fedora), the user needs to make sure that the contents inside `/tools/rootfs` have the proper permission for user access. Since the mount command is made as root, the content shows as restricted access after the command `cp -r /mnt/rootfs`, which may prevent the NFS mount from working correctly.

To use the root file system created in the LTIB directory after the kernel build, use the command:

```
%export ROOTFS_DIR=/<LTIB directory>/rootfs
```

2.4 Copying images to TFTP server

To use tftp server to download the image, copy the kernel image in the release package or LTIB to the tftp directory. For example:

```
cp imx5x/uImage /tftpboot
```

or

```
cp /<LTIB directory>/rootfs/boot/uImage /tftpboot
```

2.5 How to generate no-padding U-Boot

To generate no-padding U-Boot, run:

```
sudo dd if=u-boot.bin of=u-boot-mx53-loco-no-padding.bin bs=512 skip=2
```

2.6 How to Generate uImage from a zImage

To generate a uImage with ltib, in the kernel source code, change the build target from “zImage” to “uImage”.

If you want to generate a uImage from a zImage you built, you can generate a “uImage," based on the above zImage as below:

- Build u-boot package to get “mkimage” tool under `rpm/BUILD/u-boot-<version>/tools/mkimage`.
- Copy `mkimage to /usr/bin/`
- Run the below command:

  ```
  mkimage -A arm -O linux -T kernel -C none -a 0x70008000 -e 0x70008000 -n "Linux-<kernel_version>" -d zImage uImage
  ```

  Note: Replace kernel_version with the appropriate kernel version for your image. For example, 2.6.35-151-xxxx.

2.7 How to Build U-Boot and Kernel in Standalone Environment

To build U-Boot in a standalone environment, do the following in the root folder of U-Boot sources:

```
make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-
4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-
gnueabi/bin/arm-none-linux-gnueabi- distclean

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-
4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-
gnueabi/bin/arm-none-linux-gnueabi- mx53_loco_config

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-
4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-
gnueabi/bin/arm-none-linux-gnueabi-
```

To build the kernel in the standalone environment, do as the following:

```
make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-
multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi- imx5_defconfig

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-
multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi- uImage
```

2.8 Build Manufacturing Firmware

Please setup LTIB environment and then configure Firmware build profile.

```
./ltib --selectype
```

Choose correct item as below:

```
--- Choose the platform type

        Selection (imx5x)  --->

--- Choose the packages profile

        Selection (mfg firmware profile)--->
```

In “Freescale iMX5x Based Boards” section, choose the board information as the following:

```
--- Choose your board for u-boot

      board (mx53_loco)  --->
```

After `ltib` has completed the build, **initramfs.cpio.gz.uboot** is generated under the ltib root folder. The **u-boot.bin** and **uImage** for MFG tool are generated under `rootfs/boot/`.

3. How to Boot the START Board

i.MX53 START board does not provide boot dip switches for different boot modes. To support different boot modes such as SATA or SDHC3 boot, refer to the HW schematic for HW rework.

3.1 How to Enter Serial Download Mode for MFG Tool

i.MX53 START does not provide dip switches to select serial download mode directly. The user can force the ROM to enter serial download mode according to ROM logic by implementing the following steps:

1. Do not insert a Micro SD card into slot 1.
2. Plug-in the power supply. Connect a USB cable between PC and USB OTG port on the i.MX53 START board.
3. Press the “POWER” key and ensure that the LED turns blue, indicating that the board has been powered.
4. Open “MfgTool.exe” and ensure “Freescale i.MX53 USB BulkIO Device” is found.
5. After the USB device is found, insert MicroSD card. Start MFG tool operations.

3.2 How to Boot From Micro SD Card

1. Insert a Micro SD card in slot 1.
2. Plug-in the power supply. Press “POWER” key and ensure the LED turns blue.
3. U-Boot log messages should appear in the serial console.

4. Flash Memory Map

This chapter describes the software layout in MMC/SD cards. This information is useful for understanding later sections about image download.

4.1 MMC/SD/SATA Memory Map

The MMC/SD/SATA scheme is different from the NAND and NOR flash which are deployed in the BSP software. The MMC/SD/SATA must keep the first sector (512 bytes) as the MBR (Master Boot Record) in order to use MMC/SD as the rootfs.

Upon boot up, the MBR is executed to look up the partition table to determine which partition to use for booting. The bootloader should be at the end of MBR. The kernel image and `rootfs` may be stored at any address after bootloader.

The MBR can be generated through the `fdisk` command when creating partitions in MMC/SD cards on a Linux Host server.

5. Downloading Images Using MFG Tool

5.1 Installing the MFG Tools

Unzip `Mfgtools-Rel-<version>_MX53_UPDATER.tar.gz`

5.2 Usage

Read the MFG tool documentation in the "Document" folder, before using the MFG tool.

Follow these instructions to use the MX53 START MFG tool:

- Connect a USB cable from a PC to the USB OTG port (J3) on the board.
- Connect UART to PC for console output. Open a Terminal emulator program.
- With no Micro SD card inserted into the board, press the "POWER" key and ensure that the LED color turns blue.
- The manufacturing tool requires your file system to be packed and compressed using `bzip2` algorithm. To create this file, you can run the following commands as a root user:

```
>cd your_rootfs_dir

>tar –cjf rootfs.tar.bz2 *
```

- You can specify your images in two ways: The first is by editing "`Profiles\MX53 Linux Update\OS Firmware\ucl.xml`" to modify the file path or flash operations according to your usage. Note that "MX53LOCO-xxx" lists are the example codes for MX53 START. You can modify them for MX53 START programming. After the modification is completed, save the changes and exit. Another way is by copying your files in "`Profiles\MX53 Linux Update\OS Firmware\files`" directory. You can replace the files inside this folder. Note that you will find `u-boot-mx53-loco.bin` and `uImage` binaries in "`Profiles\MX53 Linux Update\OS Firmware`" folder. These files should not be replaced. They are different from your image files and serve another purpose.
- Execute "`MfgTool.exe`". Select the "Options -> Configuration" menu. If this is the first time connecting an i.MX53 board with the MFG tool, install a USB driver under "`Drivers\iMX_BulkIO_Driver`".
- Select the appropriate USB port in the sheet "USB Ports".

- Select the appropriate profile in the sheet “Profiles.” Type the item in “Operations.” A Pop-Up list appears. Select “MX53LOCO-SD” to program images to SD. If you want to program images to SATA, you can refer to MX53SMD-SATA to write the codes. Note that the default profiles are used as examples. They should be modified for programming operations.
- Insert Micro-SD card.
- Start the downloading process by pressing the green, **Start,** button. You will see the progress bar as well as the current task in the notification bar as shown in Figure 5-5. When you see “Update Complete” in the notification bar, press the red, **Stop,** button to finish.
  - The manufacturing tool may sometimes report an error message while downloading the file system in an SD card. This issue can be caused by insufficient space in the SD card due to a small partition size. To prevent the error message from being reported, edit the file “`Profiles\MX53 Linux Update\OS Firmware\fdisk-u.input`” and increase the size of the partition according to your file system requirements. Note that the contents of this file are *`fdisk`* inputs, so to increase the partition size, increase the number before “w” letter. To edit the fdisk-u.input file please use a binary editor. The file contains characters which are not recognized by a general text editor

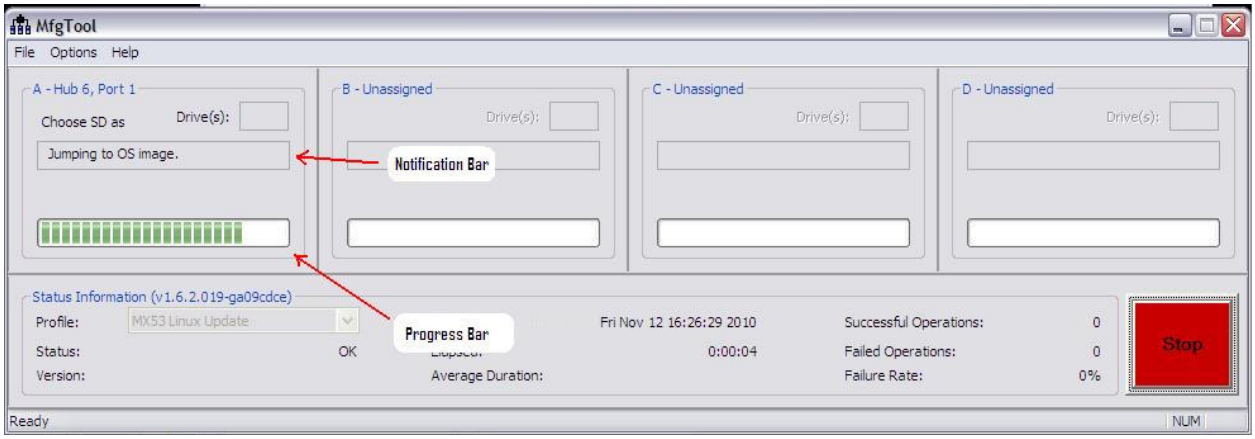

Figure 5-1 Programming SD With Manufacturing Tool.

6. Download Images by Bootloader or NFS

6.1 Setup Terminal

The i.MX53 START board can communicate with a host server (Windows or Linux) using the serial cable. Common serial communication programs such as HyperTerminal, Tera Term or PuTTY can be used. The example below describes the serial terminal setup using HyperTerminal on a Windows host:

1. Connect the target and the Windows PC using a serial cable.
2. Open HyperTerminal on the Windows PC, and select the settings as shown in Figure 6-1.

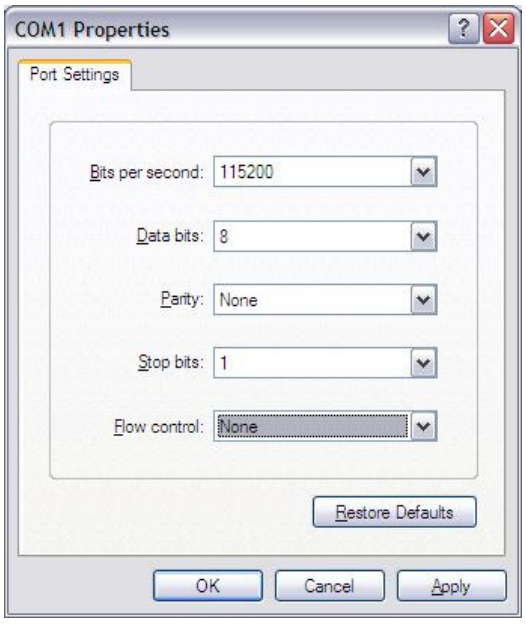

Figure 6 -1 HyperTerminal Settings for Terminal Setup

3. After the bootloader is programmed on SD card, press “POWER” key to power up the board. The bootloader prompt is displayed on the terminal screen.

6.2 Download by U-Boot

6.2.1 MMC/SD

1. To clean up the environments stored on MMC/SD, do as the following in U-Boot console:

```
MX53-LOCO U-Boot > mmc dev 0

MX53-LOCO U-Boot > mmc write 0x70100000 0x600 0x10
```

2. Power up the board and set the U-Boot environment variables as needed. For example,

```
MX53-LOCO U-Boot > setenv serverip 10.192.225.216
MX53-LOCO U-Boot > setenv bootfile uImage
MX53-LOCO U-Boot > saveenv
```

3. Copy uImage to `tftp` server. Then download it to RAM:

```
MX53-LOCO U-Boot > dhcp
```

4. Query the information about MMC/SD card. Two slots are found. The device number of slot1 (Micro SD) is 0. The device number of slot 3 is 1.

```
MX53-LOCO U-Boot > mmc list
FSL_ESDHC: 0
FSL_ESDHC: 1
MX53-LOCO U-Boot > mmc dev 0
mmc0 is current device
MX53-LOCO U-Boot > mmc part

Partition Map for UNKNOWN device 0  --   Partition Type: DOS

Partition     Start Sector     Num Sectors     Type
    1                20480          307200      83
```

5. Select active mmc slot:

```
MX53-LOCO U-Boot > mmc dev 0
```

6. Check the usage of "`mmc`" command. The "`blk#`" is equal to "`<the offset of read/write>/<block length of the card>`". The "`cnt`" is equal to "`<the size of read/write>/<block length of the card>`".

```
MX53-LOCO U-Boot > help mmc
mmc - MMC sub system

Usage:
mmc read addr blk# cnt
mmc write addr blk# cnt
mmc erase blk# cnt
mmc rescan
mmc part - lists available partition on current mmc device
mmc dev [dev] [part] - show or set current mmc device [partition]
```

```
mmc bootpart [dev] [part] - show or set boot partition
mmc list - lists available devices
```

7. Program the kernel uImage into MMC/SD. For example, the below command writes the image with the size 0x300000 from `${loadaddr}` to the offset 0x100000 of the MMC/SD card. Here 0x800 =0x100000/512, 0x1800=0x300000/512. The block size of this card is 512. This example assumes the kernel image is less than 0x300000 bytes. If the kernel image exceeds 0x300000, enlarge the image length.

```
## Write the kernel image to Micro-SD card on slot 1
mmc write ${loadaddr} 0x800 0x1800

## Write the kernel image to SD  card on slot 3
MX53-LOCO U-Boot > mmc dev 1
mmc1 is current device
MX53-LOCO U-Boot > mmc write ${loadaddr} 0x800 0x1800

MMC write: dev # 1, block # 2048, count 6144 ... 6144 blocks write: OK
```

8. Boot up the system through RFS in Micro SD card with VGA output:

```
MX53-LOCO U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs}
              console=tty1 root=/dev/mmcblk0p1 rootwait rw
              video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga'
MX53-LOCO U-Boot > setenv bootcmd_mmc 'run bootargs_base
              bootargs_mmc;mmc dev 0;mmc read ${loadaddr} 0x800
              0x1800;bootm'
MX53-LOCO U-Boot > setenv bootcmd 'run bootcmd_mmc'
MX53-LOCO U-Boot > saveenv
MX53-LOCO U-Boot > reset
```

9. To boot up the system through RFS in SD card using slot 3 with LVDS0, change the command as follows:

```
MX53-LOCO U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs}
console=tty1 root=/dev/mmcblk1p1 rootwait rw video=mxcdi0fb:RGB666,XGA
di0_primary ldb=di0'
MX53-LOCO U-Boot > setenv bootcmd_mmc 'run bootargs_base bootargs_mmc;mmc
dev 1;mmc read ${loadaddr} 0x800 0x1800;bootm'
MX53-LOCO U-Boot > setenv bootcmd 'run bootcmd_mmc'
MX53-LOCO U-Boot > saveenv
MX53-LOCO U-Boot > reset
```

10. To program the rootfs to MMC/SD, See Section 6.3 “Use i.MX53 as Host Server to Create rootfs” or section “Using a Linux Host to Set Up an SD/MMC card”.

6.2.2 SATA

1. Connect a SATA device to the START board. An additional power supply will be needed for the SATA device.

2. Power up the board and set the U-Boot environment variables as needed. For example,

```
MX53-LOCO U-Boot > setenv serverip 10.192.225.216
MX53-LOCO U-Boot > setenv bootfile uImage
MX53-LOCO U-Boot > saveenv
```

3. Copy uImage to tftp server. Then download it to RAM:

```
MX53-LOCO U-Boot > dhcp
```

4. Query SATA information:

```
MX53-LOCO U-Boot > sata info


SATA device 0: Model: Hitachi HTS545032B9A300 Firm: PB3OC60G Ser#:
090311PB0300QKG2EUZA

            Type: Hard Disk

            Supports 48-bit addressing

Capacity: 305245.3 MB = 298.0 GB (625142448 x 512)
```

5. Program the kernel uImage into SATA. For example, the below command writes the image with the size 0x300000 from ${loadaddr} to the offset 0x100000 of the SATA. Here 0x800 =0x100000/512, 0x1800=0x300000/512. The block size of this card is 512. This example assumes the kernel image is less than 0x300000 bytes. If the kernel image exceeds 0x300000, enlarge the image length.

```
MX53-LOCO U-Boot > sata write ${loadaddr} 0x800 0x1800

SATA write: device 0 block # 2048, count 6144 ... 6144 blocks written:
OK
```

6. Boot up the system through RFS in SATA:

```
MX53-LOCO U-Boot > setenv bootargs_sata 'setenv bootargs ${bootargs}
             console=tty1 root=/dev/sda1 rootwait rw
             video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga'
MX53-LOCO U-Boot > setenv bootcmd_sata 'run bootargs_base
             bootargs_sata;sata read  ${loadaddr} 0x800 0x1800;bootm'
MX53-LOCO U-Boot > setenv bootcmd 'run bootcmd_sata'
MX53-LOCO U-Boot > saveenv
MX53-LOCO U-Boot > reset
```

6.2.3 U-Boot Configurations

The U-Boot “print” command can be used to check environment variable values. The “setenv” command can be used to set environment variable values. See the U-Boot user guide for details.

6.3 Use i.MX53 as Host Server to Create rootfs

Linux provides multiple methods to program images to the storage device. This section describes how to use the i.MX53 START as Linux Host server to create the rootfs on MMC/SD card or SATA device.

1. Boot from NFS or other storage. Check partitions information:

```
root@freescale ~$ cat /proc/partitions
```

2. To create a partition in MMC/SD Slot 0, use the `fdisk` command in the Linux console:

```
root@freescale ~$ fdisk /dev/mmcblk0
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that the previous content
won't be recoverable.

The number of cylinders for this disk is set to 124368.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/mmcblk0: 4075 MB, 4075290624 bytes
4 heads, 16 sectors/track, 124368 cylinders
Units = cylinders of 64 * 512 = 32768 bytes

        Device Boot      Start         End      Blocks  Id System
```

3. If creating a partition on a SATA device, the command can be changed to "`fdisk /dev/sda`"

4. As described in Chapter 4, the rootfs partition should be located after kernel image; the first 0x800000 bytes can be reserved for MBR, bootloader, and kernel sections.

   From the above log, the `Units` of current MMC/SD card is 32768 bytes. The beginning cylinder of the first partition can be set as "0x300000/32768 = 96." The last cylinder can be set according to the `rootfs` size. Create a new partition by typing:

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-124368, default 1): 96
```

```
Last cylinder or +size or +sizeM or +sizeK (96-124368, default 124368):
Using default value 124368

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-r mmcblk0:ead partition table
 p1
```

5. Format the MMC/SD partitions as types `ext3` or `ext4` type. For example, to use `ext3`:

```
root@freescale ~$ mkfs.ext3 /dev/mmcblk0p1
mke2fs 1.41.4 (27-Jan-2009)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
248992 inodes, 994184 blocks
49709 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1019215872
31 block groups
32768 blocks per group, 32768 fragments per group
8032 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

6. Copy the `rootfs` contents to the MMC/SD card (copy the `rootfs.ext2` to NFS `rootfs`)

```
mount -t ext2 -o loop /rootfs.ext2 /mnt/cdrom
cd /mnt
mkdir mmcblk0p1
mount -t ext3 /dev/mmcblk0p1 /mnt/mmcblk0p1/
cp -rf /mnt/cdrom/* /mnt/mmcblk0p1/
umount /mnt/mmcblk0p1
umount /mnt/cdrom
```

7. Type `sync` to write the contents to MMC/SD.

8. Type `poweroff` to power down the system. Follow the instructions in Chapter 7 to boot the image from MMC/SD card.

7. Using a Linux Host to Set Up an SD/MMC Card

7.1.1 Requirements

An SD/MMC card reader, like a USB card reader, is required. It will be used to transfer the boot loader and kernel images to initialize the partition table and copy the root file system. To simplify the instructions, it is assumed that a 4GB SD/MMC card is used.

Any Linux distribution can be used for the following procedure. It is recommended to use a Linux distribution that LTIB has been tested against (like Fedora, or Ubuntu).

The Linux kernel running on the Linux host will assign a device node to the SD/MMC card reader. The kernel might decide the device node name or udev rules might be used. In the following instructions, it is assumed that udev is not used.

To identify the device node assigned to the SD/MMC card, enter the command:

```
$ cat /proc/partitions
major minor  #blocks  name
   8     0   78125000 sda
   8     1   75095811 sda1
   8     2          1 sda2
   8     5    3028221 sda5
   8    32  488386584 sdc
   8    33  488386552 sdc1
   8    16    3921920 sdb
   8    18    3905535 sdb1
```

In this example, the device node assigned is `/dev/sdb` (a block is 1kB large).

7.1.2 Copying the Boot Loader Image

Enter the following command to copy the U-Boot image to the SD/MMC card (note that this operation will delete the partition table present on the media):

```
$ sudo dd if=u-boot.bin of=/dev/sdb bs=512 && sync && sync
```

To update U-Boot to another version, run the following command:

```
$ sudo dd if= u-boot.bin  of=/dev/sdb bs=512 seek=2 skip=2&& sync && sync
```

The first 1 KB, that includes the partition table, will be preserved.

7.1.3 Copying the Kernel Image

The following command will copy the kernel image to the SD/MMC card:

```
$ sudo dd if= uImage of=/dev/sdb bs=512 seek=2048 && sync && sync
```

This will copy the `uImage` to the media at offset 1 MB.

7.1.4 Copying the File System (rootfs)

A partition table must be first created. If a partition already exists and it is big enough for the file system you want to deploy, then you can skip this step.

To create a partition, at offset 8192 (in sectors of 512 bytes) enter the following command:

```
$ sudo fdisk /dev/sdb
```

Type the following parameters (each followed by **<ENTER>**):

```
u       [switch the unit to sectors instead of cylinders]
d       [repeat this until no partition is reported by the ‘p’ command ]
n       [create a new partition]
p       [create a primary partition]
1       [the first partition]
8192    [starting at offset sector #8192, i.e. 4MB, which leaves enough space
for the kernel, the boot loader and its configuration data]
<enter> [using the default value will create a partition that spans to the
last sector of the medium]
w       [ this writes the partition table to the medium and fdisk exits]
```

The file system format `ext3` or `ext4` is a good option for removable media due to the built-in journaling. Run the following command to format the partition:

```
$ sudo mkfs.ext3 /dev/sdb1
```

Or

```
$ sudo mkfs.ext4 /dev/sdb1
```

Copy the target file system to the partition:

```
$ mkdir /home/user/mountpoint
$ sudo mount /dev/sdb1 /home/user/mountpoint
```

Assume that the root file system files are located in `/home/user/rootfs`**:**

```
$ cd /home/user/rootfs
$ sudo cp –rpa [A–z]* /home/user/mountpoint
```

```
$ sudo umount /home/user/mountpoint
```

The file system content is now on the media.

8. Running the Image on the Target

This chapter explains how to run an image on the target from downloaded device and NFS. These instructions assume that you have downloaded the kernel image using the instructions in Chapter 5 or Chapter 6.

8.1 Run the image from NFS

To boot from NFS, do as follows (attention must be paid to the items marked in blue color. You need to modify them as per your environment or HW information):

1. Press “POWER” key to power up the board.
2. Enter the following commands in the U-Boot prompt:

```
MX53-LOCO U-Boot > setenv serverip 10.192.225.216

MX53-LOCO U-Boot > setenv bootfile uImage

MX53-LOCO U-Boot > setenv nfsroot
10.192.225.216:/data/rootfs_home/rootfs_mx53

MX53-LOCO U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc0,115200'

MX53-LOCO U-Boot > setenv bootargs_nfs 'setenv bootargs ${bootargs}
root=/dev/nfs ip=dhcp nfsroot=${nfsroot},v3,tcp video=mxcdi1fb:GBR24,VGA-XGA
di1_primary vga'

MX53-LOCO U-Boot > setenv bootcmd_net 'run bootargs_base bootargs_nfs;bootm'

MX53-LOCO U-Boot > setenv bootcmd 'dhcp; run bootcmd_net'

MX53-LOCO U-Boot > saveenv

MX53-LOCO U-Boot > run bootcmd
```

NOTE

If your system is using the Seiko touchscreen as a primary device you should use the following configuration:
video=mxcdi0fb:RGB24, SEIKO-WVGA di0_primary

8.2 Run the Image from MMC/SD

To boot the system from MMC/SD flash follow the steps bellow:

1. Press the “POWER” key to power up the board.
2. Assume the kernel image starts from the address 0x100000 byte (the block start address is 0x800). The kernel image size is less than 0x300000 byte. The `rootfs` is located into `/dev/mmcblk0p1` partition. Enter the following commands in the U-Boot prompt:

```
MX53-LOCO U-Boot > setenv bootargs_base 'setenv bootargs
console=ttymxc0,115200'

MX53-LOCO U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs}
root=/dev/mmcblk0p1 rootwait rw video=mxcdi1fb:GBR24,VGA-XGA
di1_primary vga'

MX53-LOCO U-Boot > setenv bootcmd_mmc 'run bootargs_base
bootargs_mmc;mmc dev 0;mmc read ${loadaddr} 0x800 0x1800;bootm'

MX53-LOCO U-Boot > setenv bootcmd 'run bootcmd_mmc'

MX53-LOCO U-Boot > saveenv

MX53-LOCO U-Boot > run bootcmd
```

NOTE

If your system is using the Seiko touchscreen as a primary device you should use the following configuration:
video=mxcdi0fb:RGB24,SEIKO-WVGA di0_primary

8.3 Run the Image from SATA

The following steps may be used to boot the system from SATA:

1. Press the POWER key to power up the board.
2. Assume the kernel image starts from the address 0x100000 byte (the block start address is 0x800). The kernel image size is less than 0x300000 byte. The `rootfs` is located in `/dev/sda1`. Enter the following commands in the U-Boot prompt:

```
MX53-LOCO U-Boot  > setenv bootargs_base 'setenv bootargs
            console=ttymxc0,115200'
```

```
MX53-LOCO U-Boot > setenv bootargs_sata 'setenv bootargs ${bootargs}
              console=tty1 root=/dev/sda1 rootwait rw
              video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga'
MX53-LOCO U-Boot > setenv bootcmd_sata 'run bootargs_base
              bootargs_sata;sata read  ${loadaddr} 0x800 0x1800;bootm'
MX53-LOCO U-Boot > setenv bootcmd 'run bootcmd_sata'
MX53-LOCO U-Boot > saveenv
MX53-LOCO U-Boot > run bootcmd
```