i.MX 6Dual/6Quad SABRE-AI Linux User's Guide

Document Number: 924-76373
Rev. 12.04.01, 05/2012

freescale™

Contents

Section Number **Title** **Page**

Chapter 1
About This Book

Chapter 2
Introduction

Chapter 3
Building the Linux Platform

Chapter 4
How to Boot the i.MX 6Dual/Quad SABRE-AI Board

Chapter 1
About This Book

This document explains how to build and install the Freescale Linux BSP on the i.MX 6Dual/Quad SABRE-AI board. All steps needed to get the i.MX 6Dual/Quad SABRE-AI 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.

1.1 References

i.MX 6Dual/Quad SABRE-AI Linux Reference Manual.

Chapter 2
Introduction

The i.MX 6Dual/Quad SABRE-AI Linux BSP is a collection of binary, source code, and support files that can be used to create a U-boot boot loader, Linux kernel image, and a root file system for i.MX 6Dual/Quad SABRE-AI board. For the steps on how to run and configure a new Ubuntu rootfs please reference How to setup ubuntu rootfs.

2.1 Boot Loader

The i.MX 6Dual/Quad SABRE-AI Linux delivery package contains the following U-Boot bootloader binary:

L3.0.15_12.04.01_ER_images_MX6X/u-boot-mx6q-sabreauto.bin

This bootloader supports SD/MMC, SPI-NOR, NAND, Parallel NOR.

2.2 Linux Kernel image

This Freescale i.MX BSP contains a pre-built kernel image based on the 3.0.15 version of the Linux kernel. The i.MX 6Dual/Quad SABRE-AI kernel image is found at the following location:

L3.0.15_12.04.01_ER_images_MX6X/uImage

2.3 Gnome mobile Root File System

The root file system package provides busybox, common libraries, and other fundamental elements.

The i.MX 6Dual/Quad SABRE-AI BSP package contains the following rootfs file system:

L3.0.15_12.04.01_ER_images_MX6X/rootfs.ext2.gz

The rootfs.ext2.gz file system includes Freescale specific libraries and gnome GUI. It can be mounted as NFS, (refer to Setting rootfs for NFS) or its contents can be stored on a boot media, such as Secure Digital (SD) card.

2.4 Ubuntu demo rootfs

An Ubuntu demo rootfs (11.10 Oneiric) with demo applications is provided for demo purpose.

The login credentials are User: linaro and Password: linaro.

Chapter 3
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.

Please note that not all of the steps are required for every boot mode. The only required steps are in Setting Up the Linux Host and Installing and Building LTIB.

3.1 Setting Up the Linux Host

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

3.2 Installing and Building LTIB

NOTE

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, in a location such as /home/user/:

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

   This command installs LTIB to your directory.

3. Build LTIB:

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

4. Select platform to **Freescale iMX reference boards** and exit, saving the changes. At the next menu, select platform type as imx6x and package profile. Exit and save the new configuration. Please note that only the profiles of **Min profile, FSL gnome release packages** and **mfg firmware profile** pass build tests.

NOTE

You can use the ./ltib -m selectype command to change the profile after the first selection.

5. To build U-Boot for i.MX 6Dual/Quad SABRE-AI board, select "Choose your board for u-boot" as "mx6q_sabreauto". Please note this option is only for U-Boot. For the kernel image, the current default kernel configuration builds a single image that works for all i.MX6 boards except i.MX6 SoloLite boards. i.MX6 SoloLite boards use 'MX6S'

```
--- Choose your board
       board (mx6q_sabreauto)  --->
```

6. Change the kernel from Linux 2.6.38-imx to Linux 3.0.15-imx

```
kernel (Linux 3.0.15-imx) --->
```

7. Close the configuration screen saving the changes.
8. Run the following command:

```
./ltib
```

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

9. Some other useful ltib commands are:

```
./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>
```

3.3 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 imx6x/rootfs.ext2.gz  /tools
cd /tools
gunzip rootfs.ext2.gz
mount -o loop -t ext2 rootfs.ext2 /mnt/rootfs
cp -a /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 -a /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:

```
%vi /etc/exports
       edit this file by adding the export directory, for example:
       /tools/rootfs *(rw,no_root_squash)
       save and exit
%exportfs -a
```

3.4 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 imx6x/uImage /tftpboot
```

or

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

3.5 How to generate no-padding U-Boot

To generate no-padding U-Boot, run the following command:

```
sudo dd if=u-boot-mx6q-sabreauto.bin of=u-boot-mx6q-sabreauto-no-padding.bin bs=512 skip=2
```

3.6 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 shown 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 0x10800000 -e 0x10800000 -n
"Linux-$(KERNELRELEASE);" -d zImage uImage
```

Note: Replace KERNELRELEASE with the appropriate kernel version for your image. For example, 3.0.15-151-xxxx.

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

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

```
make ARCH=arm
CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-
linaro-toolchain/bin/arm-none-linux-gnueabi- distclean

make ARCH=arm
CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-
linaro-toolchain/bin/arm-none-linux-gnueabi- mx6q_sabreauto_config


make ARCH=arm
CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-
linaro-toolchain/bin/arm-none-linux-gnueabi-
```

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

```
make ARCH=arm
CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-
linaro-toolchain/bin/arm-none-linux-gnueabi- imx6_defconfig

make ARCH=arm
CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-
linaro-toolchain/bin/arm-none-linux-gnueabi- uImage
```

3.8 How to setup ubuntu rootfs

To setup ubuntu rootfs, perform the following actions:

- Follow the instructions of Chapter 8 to create an SD card with a valid U-boot, kernel, and root file system partition without any content and with format only.
- Mount SD Card and uncompress the root file system files:

```
sudo mount /dev/mmcblk0p1 /mnt/hd
cd /mnt/hd
```

```
sudo tar --numeric-owner -xzvf /<path>/oneiric.tgz
```

NOTE

The option “--numeric-owner” may not be available if you are using busybox. Using this option is mandatory so please make sure you use the full blown version of tar.

- Boot with oneiric rootfs. Log in as **linaro** (not root), the password is **linaro**.
- Install FSL packages. For this you should already have copied all the *.deb files that come along with the Released BSP to the Target Board. The deb files of each release can be located in the demo image package:

```
sudo dpkg \--force-architecture \-i *.deb
depmod
```

- Flush data to SD card and reboot:

```
sync
sudo halt
```

- Oneiric demo rootfs comes from Linaro release. It can be downloaded from https://wiki.linaro.org/Boards/MX6QSabreLite. Then apply the following changes:
  - To reserve the DMA buffer for video playback, ensure /proc/sys/vm/lowmem_reserve_ratio value as 1. This setting can be added into /etc/rc.local:

```
echo 1 1 > /proc/sys/vm/lowmem_reserve_ratio
```

3.9 Build Manufacturing Firmware

Please setup LTIB environment and then configure Firmware build profile.

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

Choose correct item as shown below:

```
--- Choose the platform type
       Selection (imx6x)  --->
--- Choose the packages profile
       Selection (mfg firmware profile)--->
```

In "Freescale iMX6x Based Boards" section, choose the board information as in the following example:

```
--- Choose your board for u-boot
                  board (mx6q_sabreauto)  --->
```

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/.

Chapter 4
How to Boot the i.MX 6Dual/Quad SABRE-AI Board

The boot modes of the i.MX 6Dual/Quad SABRE-AI board are controlled by the boot configuration DIP switches on the board. To locate the boot configuration switches refer to the i.MX 6Dual/Quad SABRE-AI Hardware User's Guide. The following sections list basic boot setup configurations only.

4.1 How to Enter Serial Download Mode for MFG Tool

Table below shows the boot switch settings which are used to enter serial download mode for MFG tool. If bootimage is not validated in boot media, system will enter serial download mode.

Table 4-1. the boot switch setup for MFG tool

Switch	D1	D2	D3	D4
SW7	OFF	ON	OFF	OFF

4.2 How to Boot From SD Card from Slot3

The following table shows the dip settings for SD boot on CPU board:

Table 4-2. the boot switch setup for SD boot on CPU board

Switch	D1	D2	D3	D4	D5	D6	D7	D8
SW1	X	X	X	OFF	ON	X	X	X
SW2	X	OFF	ON	OFF	-	-	-	-
SW3	OFF	OFF	ON	OFF	-	-	-	-

The following table shows the dip settings for MMC boot on CPU board:

Table 4-3. the boot switch setup for MMC boot on CPU board

Switch	D1	D2	D3	D4	D5	D6	D7	D8
SW1	X	X	X	OFF	ON	X	X	X
SW2	X	ON	ON	OFF	-	-	-	-
SW3	OFF	OFF	ON	OFF	-	-	-	-

4.3 How to Boot From NAND

The following table shows the dip settings for NAND boot:

Table 4-4. the boot switch setup for NAND

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
SW1	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF
SW2	OFF	OFF	OFF	ON	-	-	-	-	-	-
SW3	OFF	OFF	ON	OFF	-	-	-	-	-	-

4.4 How to Boot From SPI-NOR

Table below shows the boot switch settings to boot from SPI-NOR

Table 4-5. the boot switch setup for SPI-NOR boot.

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
S1	X	X	X	X	X	X	X	X	X	X
S2	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
SW7	OFF	OFF	ON	OFF	-	-	-	-	-	-

4.5 How to Boot from WEIM(Parallel) NOR

The following table shows the dip settings for NOR boot:

Table 4-6. boot switch setup for WEIM NOR

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
SW1	X	X	X	OFF	ON	X	X	X	X	X
SW2	X	OFF	OFF	OFF	-	-	-	-	-	-
SW3	OFF	OFF	ON	OFF	-	-	-	-		

NOTE

SPI and WEIM NOR has pins conflicts. Both cannot be used at the same config. Default u-boot configuration is set to SPI NOR.

In order to boot from WEIM NOR, enable the config CONFIG_CMD_WEIMNOR in the u-boot. This will disable SPI NOR.

Chapter 5
Flash Memory Map

This chapter describes the software layout in MMC/SD cards.

This information may be useful for understanding subsequent sections about image download.

5.1 MMC/SD Memory Map

The MMC/SD scheme is different from the NAND and NOR flash which are deployed in the BSP software. The MMC/SD 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 after the 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.

5.2 NAND flash memory map

NAND flash scheme is configured by kernel command line.

For example:

```
mtdparts=gpmi-nfc:20m(boot),200m(test, -(user)
```

Chapter 6
Downloading Images Using MFG Tool

This chapter describes the procedure for using the MFG tool to download images to the different devices.

6.1 Installing the MFG Tools

Unzip Mfgtools-Rel-12.04.01_ER_MX6Q_UPDATER.tar.gz

6.2 Usage

Read the MFG tool documentation in the "Document" folder, before using the MFG tool. The MFG tool follows the instructions in "Profiles\MX6Q Linux Update\OS Firmware \ucl.xml" to execute program operations. The user must read and update ucl.xml to understand the operations before using the MFG tool.

Follow these instructions to use the i.MX 6Dual/Quad SABRE-AI MFG tool:

- Connect a USB cable from a PC to the USB OTG port on the board.
- Connect UART to PC for console output. Open a Terminal emulator program.
- Set boot pin to Mfgtools mode. Refer to How to enter serial download mode.
- The default profile of the manufacturing tool assumes 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. You can also modify profile to support other formats.
  - >cd *your_rootfs_dir*
  - >tar -cjf rootfs.tar.bz2 *
- You can specify your images in two ways: The first is by editing "Profiles\MX6Q Linux Update\OS Firmware\ucl.xml" to modify the file path or flash operations according to your usage. You can modify them for i.MX 6Dual/Quad SABRE-AI programming. After the modification is completed, save the changes and exit. Another way is by copying your files in "Profiles\MX6Q Linux Update\OS Firmware \files" directory. You can replace the files inside this folder.

NOTE

You will find u-boot-<board>.bin and uImage binaries in "Profiles\MX6Q 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" and power on the board. If this is the first time connecting an i.MX6Q board with the MFG tool, System will automatically install HID driver for you.
- Select the appropriate USB port in the sheet "USB Ports". Or Click "Scan" button.
- Under the “Options” menu, choose “Configuration”. Select the appropriate profile under the tab labeled "Profiles." In the “Operations" section there is a column labeled “Options”.
- 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 below. When you see "Update Complete" in the notification bar, press the red, Stop, button to finish.
- "Sabre-ARD-SPI_NOR & SD card" to program u-boot on the SPI NOR and Kernel and rootfs in the SD card.
- The manufacturing tool may sometimes report an error message when it is downloading the file system in an SD card. This can be caused by insufficient space in the SD card due to a small partition size. To fix this, unzip the file "Profiles \MX6Q Linux Update\OS Firmware\mksdcard.sh.tar" and then modify the script to increase the size of the partition and create more partitions according to your file system requirements. After the modification is done, tar the script again.

Chapter 7
Download Images by Bootloader or NFS

7.1 Setup Terminal

The i.MX 6Dual/Quad SABRE-AI 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 below.

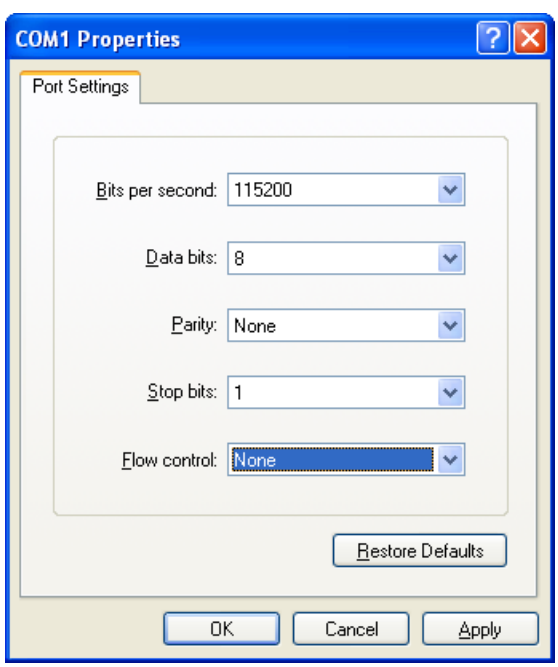

Figure 7-1. HyperTerminal Settings for Terminal Setup

7.2 Download by U-Boot

The following section describes how to download by U-boot.

7.2.1 MMC/SD on SD3(J14 CPU Board)

To enter the U-Boot prompt, press any key before the U-Boot environment variable, "bootdelay", is down counted and before it times out. The default setting is 3 seconds.

1. To restore the environment variables stored on MMC/SD to their defaults, do the following in U-Boot console:

```
MX6Q SABREAUTO U-Boot >mmc dev 2
MX6Q SABREAUTO U-Boot >mmc write 0x10800000 0x600 0x10
```

   or

```
MX6Q SABREAUTO U-Boot >destoryenv
```

2. Configure U-Boot environment for network communications. Below is an example. The lines with '#' character are comments and have no effect.
3. Copy uImage to tftp server. Then download it to RAM:

```
MX6Q SABREAUTO U-Boot > dhcp
```

4. Query the information about MMC/SD card in slot 3.

```
MX6Q SABREAUTO U-Boot >mmc dev 2
MX6Q SABREAUTO U-Boot >mmcinfo
```

5. 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>".

```
MX6Q SABREAUTO 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
```

6. Program the kernel uImage located in RAM at ${loadaddr} into the microSD. For example the command to write the image with the size 0x400000 from ${loadaddr} to the offset of 0x100000 of the microSD card. Refer to the following examples for the definition of the mmc Parameters.

```
blk# = (microSD Offset)/(SD block length) = 0x100000/512 = 0x800

cnt = (image Size)/(SD block length) = 0x400000/512 = 0x2000
```

This example assumes the kernel image is less than 0x400000. If the kernel image exceeds 0x400000, increase the image length. After issuing the tftp command, filesize U-Boot environment variable is set with the number of bytes transferred. This can be checked to determine the correct size needed for the calculation. Use U-Boot command printenv to see the value.

```
MX6Q SABREAUTO U-Boot >mmc dev 2
MX6Q SABREAUTO U-Boot >mmc write 0x10800000 0x800 0x2000
```

7. Boot up the system through rootfs in SD card via HannStar LVDS:

```
MX6Q SABREAUTO U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/
mmcblk0 rootwait rw video=mxcfb0:dev=ldb,LDB-XGA,if=RGB666 ip=dhcp'
MX6Q SABREAUTO U-Boot >setenv bootcmd_mmc 'run bootargs_base bootargs_mmc;mmc dev 2;mmc
read ${loadaddr} 0x800 0x2000;bootm'
MX6Q SABREAUTO U-Boot > setenv bootcmd 'run bootcmd_mmc'
MX6Q SABREAUTO U-Boot > saveenv
```

7.3 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.

Chapter 8
Using a Linux Host to Set Up an SD/MMC Card

This chapter describes the steps to prepare an SD/MMC card to boot up an i.MX 6Dual/Quad SABRE-AI board.

8.1 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).

8.2 Copying the Boot Loader Image

Enter the following command to copy the U-Boot image to the SD/MMC card:

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

This assumes a pre-built u-boot image delivered with the BSP or built from the u-boot source code. If using a non-padded u-boot image, "skip=2" should be omitted from the above command line. The first 1 KB of the SD/MMC card, that includes the partition table, will be preserved.

8.3 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 conv=fsync
```

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

8.4 Copying the File System (rootfs)

First, a partition table must be 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 16384 (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]
16384   [starting at offset sector #16384, 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
```

NOTE

This may take several minutes depending on the size of your rootfs.

The file system content is now on the media.

Chapter 9
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 Downloading Images Using MFG Tool or Download Images by Bootloader or NFS or Using a Linux Host to Set Up an SD/MMC Card. If you have not setup your Serial Terminal yet, please refer to Setup Terminal .

9.1 Run the image from NFS

To boot from NFS, do as follows:

1. Power on the board.
2. Enter the following commands in the U-Boot prompt:

```
MX6Q SABREAUTO U-Boot > setenv serverip 10.192.225.216    (*)
MX6Q SABREAUTO U-Boot > setenv bootfile uImage    (*)
MX6Q SABREAUTO U-Boot > setenv nfsroot /data/rootfs_home/rootfs_mx6    (*)
MX6Q SABREAUTO U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc3,115200'
### LVDS
MX6Q SABREAUTO U-Boot > setenv bootargs_nfs 'setenv bootargs ${bootargs} root=/dev/nfs
ip=dhcp
nfsroot=${serverip}:${nfsroot},v3,tcp video=mxcfb0:dev=ldb,LDB-XGA,if=RGB666'    (*)
### HDMI
MX6Q SABREAUTO U-Boot > setenv bootargs_nfs 'setenv bootargs ${bootargs} root=/dev/nfs
ip=dhcp nfsroot=${serverip}:${nfsroot},v3,tcp video=mxcfb1:dev=ldb,LDB-XGA,if=RGB666
video=mxcfb0:dev=hdmi,1920x1080M@60,if=RGB24' (*)
MX6Q SABREAUTO U-Boot > setenv bootcmd_net 'run bootargs_base bootargs_nfs;bootm'
MX6Q SABREAUTO U-Boot > setenv bootcmd 'dhcp; run bootcmd_net'
MX6Q SABREAUTO U-Boot > saveenv
```

NOTE

If MAC address has not burned into fuse, you must set MAC address to use network in uboot.

```
setenv ethaddr xx:xx:xx:xx:xx:xx
```

9.2 Run the Image from MMC/SD

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

1. Power on 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 0x400000 byte. Enter the following commands in the U-Boot prompt:

```
MX6Q SABREAUTO U-Boot > setenv loadaddr 0x10800000
MX6Q SABREAUTO U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc3,115200'
MX6Q SABREAUTO U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs} root=/dev/
mmcblk0p1 rootwait rw video=mxcfb1:dev=ldb,LDB-XGA,if=RGB666 video=mxcfb0:dev=hdmi,
1920x1080M@60,if=RGB24 ip=dhcp'
MX6Q SABREAUTO U-Boot > setenv bootcmd_mmc 'run bootargs_base bootargs_mmc;mmc dev
2;mmc read ${loadaddr} 0x800 0x2000;bootm'
MX6Q SABREAUTO U-Boot > setenv bootcmd 'run bootcmd_mmc'
MX6Q SABREAUTO U-Boot > saveenv
MX6Q SABREAUTO U-Boot > run bootcmd
```

9.3 Run the Image from NAND

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

1. Power up the board.
2. Assume the kernel image starts from the address 0x1400000 byte (the block start address is 0x800). The kernel image size is less than 0x400000 byte. The rootfs is located in /dev/mtd2. Enter the following commands in the U-Boot prompt:

```
MX6Q SABREAUTO U-Boot > setenv loadaddr 0x10800000
MX6Q SABREAUTO U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc3,115200'
MX6Q SABREAUTO U-Boot > setenv bootargs_nand 'setenv bootargs ${bootargs} ubi.mtd=2
root=ubi0:rootfs rootfstype=ubifs rootwait rw mtdparts=gpmi-nand:20m(boot),20m(kernel),
1024m(rootfs),-(user) gpmi_debug_init'
MX6Q SABREAUTO U-Boot > setenv bootcmd_nand 'run bootargs_base bootargs_nand;nand read $
{loadaddr} 0x1400000 0x400000;bootm'
MX6Q SABREAUTO U-Boot > setenv bootcmd 'run bootcmd_nand'
MX6Q SABREAUTO U-Boot > run bootcmd
```

Chapter 10
How To Check Current CPU Frequency

Scaling governors has been used in the Linux kernel to set the CPU frequency. CPU frequencies can be scaled automatically depending on the system load, in response to ACPI events, or manually by userspace programs. For more information about governors, read governors.txt from http://kernel.org/doc/Documentation/cpu-freq/governors.txt

Following are some of the frequently used commands

To get the available Scaling governors

```
cat /sys/devices/system/cpu/*/cpufreq/cpuinfo_cur_freq
```

To check the current CPU frequency:

```
cat /sys/devices/system/cpu/*/cpufreq/cpuinfo_cur_freq
```

Frequency will be displayed depending on the governor set.

To check the maximum frequency:

```
cat /sys/devices/system/cpu/*/cpufreq/cpuinfo_max_freq
```

To check the minimum frequency:

```
cat /sys/devices/system/cpu/*/cpufreq/cpuinfo_min_freq
```

To set constant CPU frequency: Set the scaling governor to userspace and set the desired frequency.

```
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Kernel is preconfigured to support only certain frequencies. The list of frequencies currently supported can be obtained from

```
cat /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state
```

Set always to Max Frequency:

```
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

10.1 How To Change the Core Operating Frequency From 1Ghz to 850 Mhz

In order to display the current supported frequencies:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state
```

Check whether 850Mhz is listed in the table. If it is not, then kernel will chose the next nearest max frequency in this case it will be 996 MHz.

```
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor

echo 852000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_set_speed
```

In order to add the supported frequency, please check the kernel source arch/arm/mach-mx6/cpu_op-mx6.c and it involves few mandatory steps.

10.2 How to check frequency from registers

The following command will help to determine that the default PLL1 (which is used by the CPU clock):

```
/unit_tests/memtool -32 0x020C8000 1
```

For example for 792MHz the _hex_ value will be '80002042', the 'div_select' divisor is specified in bits [6:0] and the PLL1 frequency is computed like so: Frequency = OSC clk (which is 24MHz) * div_select/2.0

…which will work out to 792 (MHz).

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 China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: 924-76373
Rev. 12.04.01, 05/2012

freescale™