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

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

freescale™

Contents

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

Chapter 7
Download Images by Bootloader or NFS

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

Chapter 9
Running the Image on the Target

Chapter 1
About This Book

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

Chapter 2
Introduction

The i.MX 6Dual/Quad SABRE-Lite 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-Lite 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

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

This bootloader supports .

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-Lite 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-Lite 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 #d18e3) 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-Lite board, select "Choose your board for u-boot" as "mx6q_sabrelite". 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.

```
--- Choose your board
       board (mx6q_sabrelite)  --->
```

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:

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

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-sabrelite.bin of=u-boot-mx6q-sabrelite-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 0x10008000 -e 0x10008000 -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_sabrelite_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_sabrelite)  --->
```

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-Lite Board

The boot modes of the i.MX 6Dual/Quad SABRE-Lite 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-Lite 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
SW1	ON	OFF

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/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 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.

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-Lite 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-Lite 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.

- 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”.
  - Select "Sabre-lite-SPI_NOR" to program images to SPINOR.
  - Select "Sabre-lite-SPI_NOR" to program images to SPI-NOR.
- 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.
- 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-Lite 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.

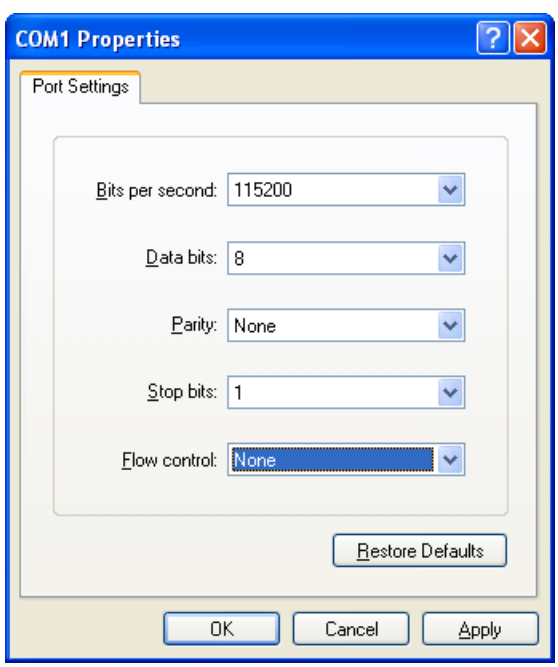

Figure 7-1. HyperTerminal Settings for Terminal Setup

3. Connect the power supply to power up the board. The bootloader prompt is displayed on the terminal screen.

7.2 Download by U-Boot

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

7.2.1 MMC/SD on SD4

The U-Boot bootloader is able to download images from a tftp server into RAM and to write from RAM to the microSD card. The location of the microSD slot is J20. For this operation the Ethernet interface is used on the Sabre-Lite board and U-Boot environment variables are initialized for network communications.

The SPI-NOR FLASH contains U-Boot which is executed upon power on. For this operation Setup boot pin, to boot from SPI-NOR as described on the 4.2 How to Boot From SPI-NOR section.

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. Configure U-Boot environment for network communications. Below is an example. The lines with '#' character are comments and have no effect.

```
### ipaddr : IP address of the Sabre-Lite board
MX6Q SABRELITE U-Boot > setenv ipaddr 192.168.1.10
### gatewayip : IP address of the network gateway
MX6Q SABRELITE U-Boot > setenv gatewayip 192.168.1.254
### netmask :  Netmask of the network
MX6Q SABRELITE U-Boot > setenv netmask 255.255.255.0
### serverip : IP address of host providing tftp server
MX6Q SABRELITE U-Boot > setenv serverip 192.168.1.254
### The ethaddr environment variable must be set, either by reading the internal fuse
which has ### been programmed or from an environment variable.
MX6Q SABRELITE U-Boot > setenv ethaddr 00:01:02:03:04:05
### Write all the environment variables to persistent storage
MX6Q SABRELITE U-Boot > saveenv


### console: Minimum console settings, by default this variable is initialized from the
build
MX6Q SABRELITE U-Boot > setenv bootargs console=ttymxc1,115200
###  loadaddr: RAM address to store the image
MX6Q SABRELITE U-Boot > setenv loadaddr 0x10800000
### bootfile: Name of kernel image on tftp server
MX6Q SABRELITE U-Boot > setenv bootfile uImage
```

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

```
MX6Q SABRELITE U-Boot > dhcp
```

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

```
MX6Q SABRELITE U-Boot >mmc dev 1
MX6Q SABRELITE U-Boot >mmcinfo
```

4. 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 SABRELITE 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
```

5. 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 SABRELITE U-Boot >mmc dev 1
MX6Q SABRELITE U-Boot >mmc write 0x10800000 0x800 0x2000
```

6. To program the root file system on the microSD card, refer to section “Use i.MX6Q as Host Server to Create rootfs” or Using a Linux Host to Set Up an SD/MMC Card
7. Refer to section Run the Image from MMC/SD for configuring the U-Boot environment variables to boot from the microSD card using the uImage Linux kernel and root file system.

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.

7.4 Use i.MX 6Dual/Quad SABRE-Lite board 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.MX 6Dual/Quad SABRE-Lite as Linux Host server to create the rootfs on MMC/SD cardor SATA device. The example below is SD card. Device file node name needs to be changed for 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 4, 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. As described in Flash Memory Map, the rootfs partition should be located after kernel image; the first 0x800000 bytes can be reserved for MBR, bootloader, and kernel sections. From the log shown above, 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-read mmcblk0 partition table

 p1
```

4. 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.
```

5. 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 -af /mnt/cdrom/* /mnt/mmcblk0p1/
umount /mnt/mmcblk0p1
umount /mnt/cdrom
```

6. Type sync to write the contents to MMC/SD.
7. Type poweroff to power down the system. Follow the instructions in Using a Linux Host to Set Up an SD/MMC Card to boot the image from MMC/SD card.

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-Lite 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.

On the i.MX 6Dual/Quad SABRE-Lite, u-boot cannot be booted from an SD/MMC card. However, an SD/MMC card in slot 4 can be used to re-program an i.MX 6Dual/Quad SABRE-Lite u-boot image in SPI-NOR using the following commands:

```
MX6Q SABRELITE U-Boot > mmc dev 1
MX6Q SABRELITE U-Boot > mmc read 0x10800000 0 200
MX6Q SABRELITE U-Boot > sf probe 1
MX6Q SABRELITE U-Boot > sf erase 0 0x40000
MX6Q SABRELITE U-Boot > sf write 0x10800000 0 0x40000
```

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 SABRELITE U-Boot > setenv serverip 10.192.225.216    (*)
MX6Q SABRELITE U-Boot > setenv bootfile uImage    (*)
MX6Q SABRELITE U-Boot > setenv nfsroot /data/rootfs_home/rootfs_mx6    (*)
MX6Q SABRELITE U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc1,115200'
### LVDS
MX6Q SABRELITE 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 SABRELITE 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 SABRELITE U-Boot > setenv bootcmd_net 'run bootargs_base bootargs_nfs;bootm'
MX6Q SABRELITE U-Boot > setenv bootcmd 'dhcp; run bootcmd_net'
MX6Q SABRELITE 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 start from the address 0x100000 byte (the block start address is 0x800). The kernel image size is less than 0x400000 byte. The rootfs is located into /dev/mmcblk0p1 partition. Enter the following commands in the U-Boot prompt:

```
 MX6Q SABRELITE U-Boot > setenv bootcmd_mmc 'run bootargs_base bootargs_mmc;mmc dev 3;mmc
read
 ${loadaddr} 0x800 0x2000;bootm'
```

NOTE

The steps listed in this section were verified with 1 SD card plugged in the board during system boot-up and loading the kernel and filesystem from the micro-SD slot. Boot-up from the regular SD slot was also confirmed; this would require a slight modification to the u-boot “bootcmd_mmc” environment variable to use “mmc dev 0” instead of "mmc dev 1"

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™