NXP Semiconductors
User's Guide

Document Number: AUG
Rev. O8.0.0_1.0.0, 02/2018

Android™ User's Guide

Contents

1 Overview

This document describes how to build Android Oreo 8.0 platform for the i.MX 6 and i.MX 7 series devices. It provides instructions for:

- Configuring a Linux® OS build machine.
- Downloading, patching, and building the software components that create the Android™ system image.
- Building from sources and using pre-built images.
- Copying the images to boot media.
- Hardware/software configurations for programming the boot media and running the images.

For more information about building the Android platform, see source.android.com/source/building.html.

2 Preparation

The minimum recommended system requirements are as follows:

- 16 GB RAM
- 300 GB hard disk

For any problems on the building process related to the jack server, see the Android website source.android.com/source/jack.html.

2.1 Setting up your computer

To build the Android source files, use a computer running the Linux OS. The Ubuntu 14.04 64bit version and openjdk-8-jdk is the most tested environment for the Android Oreo 8.0 build.

After installing the computer running Linux OS, check whether all the necessary packages are installed for an Android build. See "Setting up your machine" on the Android website source.android.com/source/initializing.html.

In addition to the packages requested on the Android website, the following packages are also needed:

```
$ sudo apt-get install uuid uuid-dev
$ sudo apt-get install zlib1g-dev liblz-dev
$ sudo apt-get install liblzo2-2 liblzo2-dev
$ sudo apt-get install lzop
$ sudo apt-get install git-core curl
$ sudo apt-get install u-boot-tools
$ sudo apt-get install mtd-utils
$ sudo apt-get install android-tools-fsutils
$ sudo apt-get install openjdk-8-jdk
$ sudo apt-get install device-tree-compiler
$ sudo apt-get install gdisk
```

NOTE

If you have trouble installing the JDK in Ubuntu, see How to install misc JDK in Ubuntu for Android build.

Configure git before use. Set the name and email as follows:

- git config --global user.name "First Last"
- git config --global user.email "first.last@company.com"

2.2 Unpacking the Android release package

After you set up a computer running Linux OS, unpack the Android release package by using the following commands:

```
$ cd ~ (or any other directory you like)
$ tar xzvf mx-o8.0.0_1.0.0_ga.tar.gz
```

3 Building the Android platform for i.MX

3.1 Getting Android source code (Android/kernel/U-Boot)

The i.MX Android release source code consists of three parts:

- NXP i.MX public source code, which is maintained in CodeAurora Forum repository.
- AOSP Android public source code, which is maintained in android.googlesource.com.
- NXP i.MX Android proprietary source code package, which is maintained in www.NXP.com.

To generate the i.MX Android release source code build environment, follow the steps below. Assume you had i.MX Android proprietary source code package imx-o8.0.0_1.0.0.tar.gz under the ~/. directory.

```
$ mkdir ~/bin
$ curl https://storage.googleapis.com/git-repo-downloads/repo  > ~/bin/repo
$ chmod a+x ~/bin/repo
$ export PATH=${PATH}:~/bin
$ source ~/imx-o8.0.0_1.0.0_ga/imx_android_setup.sh
# By default, the imx_android_setup.sh script will create the source code build environment
in the folder ~/android_build
# ${MY_ANDROID} will be refered as the i.MX Android source code root directory in all i.MX
Andorid release documentation.
$ export MY_ANDROID=~/android_build
```

3.2 Building Android images

Building the Android image is performed when the source code has been downloaded (Section 3.1) and patched (Section 3.2).

Commands **lunch <buildName-buildType>** to set up the build configuration and **make** to start the build process are executed.

The build configuration command **lunch** can be issued with an argument <Build name>-<Build type> string, such as **lunch sabresd_6dq-userdebug**, or can be issued without the argument presenting a menu of selection.

The Build Name is the Android device name found in the directory ${MY_ANDROID}/device/fsl/. The following table lists the i.MX build names.

Table 1. Build names

Build name	Description
evk_6sl	i.MX 6SoloLite Evaluation Kit
sabreauto_6q	i.MX 6Quad/6DualLite/6QuadPlus SABRE-AI Board
sabresd_6dq	i.MX 6Quad/6DualLite/6QuadPlus SABRE-SD Board and SABRE Platform
sabresd_6sx	i.MX 6SoloX SABRE-SD Board
sabreauto_6sx	i.MX 6SoloX SABRE-AI Board
sabresd_7d	i.MX 7Dual SABRE-SD Board

The build type is used to specify what debug options are provided in the final image. The following table lists the build types.

Table 2. Build types

Build type	Description
user	Production ready image, no debug
userdebug	Production ready image similar with "user" but with root access and debug tools
eng	Development image with debug tools

Android build steps are as follows:

1. Change to the top level build directory.

   ```
   $ cd ${MY_ANDROID}
   ```
2. Set up the environment for building. This only configures the current terminal.

   ```
   $ source build/envsetup.sh
   ```
3. Execute the Android **lunch** command. In this example, the setup is for the production image of i.MX 6DualQuad SABRE Board/Platform device with user type.

   ```
   $ lunch sabresd_6dq-userdebug
   ```

4. Execute the **make** command to generate the image.

```
$ make 2>&1 | tee build-log.txt
```

When the **make** command is complete, the build-log.txt file contains the execution output. Check for any errors.

For BUILD_ID & BUILD_NUMBER changing, update build_id.mk in your ${MY_ANDROID} directory. For details, see the *Android™ Frequently Asked Questions* (AFAQ).

For i.MX 6DualLite SABRE-SD, i.MX 6Dual/6Quad SABRE-SD, and i.MX 6QuadPlus SABRE-SD boards, the same build configuration is used. These two boards share the same kernel/system/recovery images with the exception of the U-Boot image. The following outputs are generated by default in ${MY_ANDROID}/out/target/product/sabresd_6dq:

- root/: root file system (including init, init.rc). Mounted at /.
- system/: Android system binary/libraries. Mounted at /system.
- data/: Android data area. Mounted at /data.
- recovery/: root file system when booting in "recovery" mode. Not used directly.
- boot-imx6q.img: composite image for i.MX 6Dual/6Quad SABRE-SD, which includes the kernel zImage, ramdisk, board's device tree binary, and boot parameters.
- boot-imx6qp.img: composite image for i.MX 6QuadPlus SABRE-SD, which includes the kernel zImage, ramdisk, board's device tree binary, and boot parameters.
- boot-imx6dl.img: composite image for i.MX 6DualLite SABRE-SD, which includes the kernel zImage, ramdisk, board's device tree binary, and boot parameters.
- boot-imx6q-ldo.img: a composite image for i.MX 6Dual/6Quad 1.2 G Hz SABRE-SD, which includes the kernel zImage, ramdisk, board's device tree binary, and boot parameters.
- ramdisk.img: ramdisk image generated from "root/". Not used directly.
- system.img: EXT4 image generated from "system/". It can be programmed to "SYSTEM" partition on SD/eMMC card with "dd".
- recovery-imx6q.img: EXT4 image for i.MX 6Dual/6Quad SABRE-SD, which is generated from "recovery/". Can be programmed to the "RECOVERY" partition on SD/eMMC card with "dd".
- recovery-imx6qp.img: EXT4 image for i.MX 6QuadPlus SABRE-SD, which is generated from "recovery/". Can be programmed to the "RECOVERY" partition on SD/eMMC card with "dd".
- recovery-imx6q-ldo.img: EXT4 image for i.MX 6Dual/6Quad 1.2 G Hz SABRE-SD, which is generated from "recovery/". Can be programmed to "RECOVERY" partition on SD/eMMC card with "dd".
- recovery-imx6dl.img: EXT4 image for i.MX 6DualLite SABRE-SD, which is generated from "recovery/". Can be programmed to the "RECOVERY" partition on SD/eMMC card with "dd".
- partition-table.img: GPT partition table image, used for 8 GB SD card.
- partition-table-14GB.img: GPT partition table image, used for 16 GB SD card.
- partition-table-28GB.img: GPT partition table image, used for 32 GB SD card.
- u-boot-imx6q.imx: U-Boot image with no padding for i.MX 6Dual/6Quad SABRE-SD.
- u-boot-imx6qpimx: U-Boot image with no padding for i.MX 6QuadPlus SABRE-SD.
- u-boot-imx6dl.imx: U-Boot image with no padding for i.MX 6DualLite SABRE-SD.
- vendor.img: vendor image, which holds platform binaries, mounted at /vendor.

NOTE

- To build the U-Boot image separately, see Building U-Boot images.
- To build the kernel uImage separately, see Building a kernel image.
- To build boot.img, see Building boot.img.

The default build is configured for internal eMMC boot storage. See Building Android image for the SD card on the SABRE-SD Board for the configuration steps to make SD card in Slot 3 as the boot storage.

3.2.1 Configuration examples of building i.MX devices

The following table shows examples of using the `lunch` command to set up different i.MX devices. After the desired i.MX device is set up, the `make` command is used to start the build.

Table 3. i.MX device lunch examples

Build name	Description
i.MX 6DualLite/Quad/QuadPlus SABRE-SD Board and Platform	$ lunch sabresd_6dq-userdebug
i.MX 6Quad/DualLite/QuadPlus SABRE-AI Board	$ lunch sabreauto_6q-userdebug
i.MX 6SoloLite EVK Board	$ lunch evk_6sl-userdebug
i.MX 6SoloX SABRE-SD Board	$ lunch sabresd_6sx-userdebug
i.MX 6SoloX SABRE-AI Board	$ lunch sabreauto_6sx-userdebug
i.MX 7Dual SABRE-SD Board	$ lunch sabresd_7d-userdebug

After the `lunch` command is executed, the **make** command is issued:

```
$ make 2>&1 | tee build-log.txt
```

3.2.2 User build mode

A production release Android system image is created by using the **user** Build Type. For configuration options, see Table "Build types" in Section Building Android images.

The notable differences between the **user** and **eng** build types are as follows:

- Limited Android System image access for security reasons.
- Lack of debugging tools.
- Installation modules tagged with user.
- APKs and tools according to product definition files, which are found in PRODUCT_PACKAGES in the sources folder ${MY_ANDROID}/device/fsl/imx6/imx6.mk. To add customized packages, add the package MODULE_NAME or PACKAGE_NAME to this list.
- The properties are set as: `ro.secure=1` and `ro.debuggable=0`.
- `adb` is disabled by default.

The i.MX development tool options are shown below from: IMX6_SW: i.MX 6 Series Software and Development Tool Resources.

Table 4. Development tool options

	i.MX 6Quad	i.MX 6Dual	i.MX 6DualLite	i.MX 6Solo	i.MX 6SoloX	i.MX 6SoloLite	i.MX 7Dual	i.MX 6QuadPlus
SABRE Board for Smart Devices	*	Uses i.MX 6Quad	-	-	*	-	-	*
SABRE Platform for Smart Devices	*	Uses i.MX 6Quad	*	Uses i.MX 6DualLite	-	-	*	-

Table continues on the next page...

Table 4. Development tool options (continued)

	i.MX 6Quad	i.MX 6Dual	i.MX 6DualLite	i.MX 6Solo	i.MX 6SoloX	i.MX 6SoloLite	i.MX 7Dual	i.MX 6QuadPlus
SABRE for Automotive Infotainment	*	Uses i.MX 6Quad	*	Uses i.MX 6DualLite	-	-	-	*
Evaluation Kit	-	-	-	-	-	*	-	-

* Supported through the superset device (i.MX 6Quad is superset to i.MX 6Dual, i.MX 6DualLite is superset to i.MX 6Solo)

Table 5. Android system image production build method 1

i.MX development tool	Description	Image build command
SABRE Board/Platform for Smart Devices	i.MX 6QuadPlus/6Quad, i.MX 6DualLite	$ make PRODUCT-sabresd_6dq-userdebug 2>&1 | tee build-log.txt
	i.MX 6SoloX	$ make PRODUCT-sabresd_6sx-userdebug 2>&1 | tee build-log.txt
	i.MX 7Dual	$ make PRODUCT-sabresd_7d-userdebug 2>&1 | tee build-log.txt
SABRE Board for Automotive Infotainment	i.MX 6Dual/6Quad/6QuadPlus, i.MX 6DualLite	$ make PRODUCT-sabreauto_6q-userdebug 2>&1 | tee build-log.txt
	i.MX 6SoloX	$ make PRODUCT-sabreauto_6sx-userdebug 2>&1 | tee build-log.txt
Evaluation Kit	i.MX 6SoloLite	$ make PRODUCT-evk_6sl-userdebug 2>&1 | tee build-log.txt

Another method to create Android System Images is setting the environment and then issuing the `make` command. To set up the environment, execute the script `${MY_ANDROID}/build/envsetup.sh`. The `lunch` command and configuration argument for the Development Tool is then executed. To start the build, use the `make` command. The following table lists the `lunch` configuration values.

Table 6. Android system image production build method 2

i.MX development tool	Description	Lunch configuration
SABRE Board/Platform for Smart Devices	i.MX 6QuadPlus/6Quad, i.MX 6DualLite	sabresd_6dq-userdebug
	i.MX 6SoloX	sabresd_6sx-userdebug
	i.MX 7Dual	sabresd_7d-userdebug
SABRE for Automotive Infotainment	i.MX 6Dual/6Quad , i.MX 6DualLite	sabreauto_6q-userdebug
	i.MX 6SoloX	sabreauto_6sx-userdebug
Evaluation Kit	i.MX 6SoloLite	evk_6sl-userdebug

An example for the SABRE Board for Smart Devices i.MX 6Dual/Quad is:

```
$ cd ${MY_ANDROID}
$ source build/envsetup.sh
$ lunch sabresd_6dq-userdebug
$ make
```

To create Android platform over-the-air, OTA, and package, the following make target is specified:

```
$ make otapackage
```

For more Android platform building information, see source.android.com/source/building.html.

All the output files to create a bootable Android image should be all created in {MY_ANDROID}/out/target/product/<board>.

3.3 Building U-Boot images

After you set up U-Boot using the steps outlined above, you can find the tool (mkimage) under tools/.

```
$ cd ${MY_ANDROID}/vendor/nxp-opensource/uboot-imx
$ export ARCH=arm
$ export CROSS_COMPILE=${MY_ANDROID}/prebuilts/gcc/linux-x86/arm/arm-linux-
androideabi-4.9/bin/arm-linux-androideabi-
$ make distclean
```

For i.MX 6Quad/6QuadPlus SABRE-SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6qpsabresdandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6qpsabresd_config
$ make
```

For i.MX 6Quad SABRE-SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6qsabresdandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6qsabresd_config
$ make
```

For i.MX 6DualLite SABRE-SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6dlsabresdandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6dlsabresd_config
$ make
```

For i.MX 6Solo SABRE-SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6solosabresdandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6solosabresd_config
$ make
```

For i.MX 6Quad SABRE-AI SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6qsabreautoandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6qsabreauto_config
$ make
```

For i.MX 6QuadPlus SABRE-AI SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6qpsabreautoandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6qpsabreauto_config
$ make
```

For i.MX 6DualLite SABRE-AI SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6dlsabreautoandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6dlsabreauto_config
$ make
```

For i.MX 6Solo SABRE-AI SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6solosabreautoandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6solosabreauto_config
$ make
```

For i.MX 6SoloLite EVK SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6slevkandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6slevk_config
$ make
```

For i.MX 6SoloX SABRE-SD SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6sxsabresdandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6sxsabresd_config
$ make
```

For i.MX 6SoloX SABRE-AI SD:

```
# To build uboot.imx that is used in Android platform
$ make mx6sxsabreautoandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx6sxsabreauto_config
$ make
```

For i.MX 7Dual SABRE-SD SD:

```
# To build uboot.imx that is used in Android platform
$ make mx7dsabresdandroid_config
# To build uboot.img that is used in MFGTOOL
$ make mx7dsabresd_config
$ make
```

"u-boot.imx" is generated if you have a successful build.

NOTE

Any image that should be loaded by U-Boot must have an unique image head. For example, data must be added at the head of the loaded image to tell U-Boot about the image (in other words, whether it's a kernel or ramfs) and how to load the image (in other words, load/execute address). Before loading any image into RAM by U-Boot, you need a tool to add this information and generate a new image which can be recognized by U-Boot. The tool is delivered together with U-Boot. After you set up U-Boot using the steps outlined above, find the tool (mkimage) under tools/. The process of using mkimage to generate an image (for example, kernel image and ramfs image), which is to be loaded by U-Boot, is outlined in the subsequent sections of this document.

3.4 Building a kernel image

Kernel image is automatically built when building the Android root file system.

The following are the default Android build commands to build the kernel image:

```
$ export PATH=${MY_ANDROID}/bootable/bootloader/uboot-imx/tools:$PATH
$ cd ${MY_ANDROID}/vendor/nxp-opensource/kernel_imx
$ echo $ARCH && echo $CROSS_COMPILE
```

Make sure that you have those two environment variables set. If the two variables are not set, set them as follows:

```
$ export ARCH=arm
$ export CROSS_COMPILE=${MY_ANDROID}/prebuilts/gcc/linux-x86/arm/arm-linux-
androideabi-4.9/bin/arm-linux-androideabi-
# Generate ".config" according to default config file under arch/arm/configs.
# To build the kernel image for i.MX 6Dual/Quad, 6QuadPlus, 6DualLite, 6Solo, 6SoloLite,
6SoloX, and 7Dual
$ make imx_v7_android_defconfig
$ make KCFLAGS=-mno-android

# To build the kernel uImage for i.MX 6Dual/Quad, 6QuadPlus, 6DualLite, and 6Solo
$ make uImage LOADADDR=0x10008000 KCFLAGS=-mno-android

# To build the kernel uImage for i.MX 6SoloLite
$ make uImage LOADADDR=0x80008000 KCFLAGS=-mno-android

# To build the kernel uImage for i.MX 6SoloX
$ make uImage LOADADDR=0x80008000 KCFLAGS=-mno-android

# To build the kernel uImage for i.MX 7Dual
$ make uImage LOADADDR=0x80008000 KCFLAGS=-mno-android


# To build the zImage that is used in MFGTOOL
# zImage is under mfgtools\Profiles\Linux\OS Firmware\firmware\
$ make imx_v7_mfg_defconfig
$ make KCFLAGS=-mno-android -j4
```

The kernel images are found in the folders: ${MY_ANDROID}/vendor/nxp-opensource/kernel_imx/arch/arm/boot/zImage and ${MY_ANDROID}/vendor/nxp-opensource/kernel_imx/arch/arm/boot/uImage.

3.5 Building boot.img

boot.img and boota are default booting command and image.

As outlined in Running the Android Platform with a Prebuilt Image, we use boota as the default image to boot, not the uramdisk and uImage we used before.

Use this command to generate boot.img under Android environment:

```
# Boot image for the i.MX 6Dual/6Quad SABRE-SD board
$ cd ${MY_ANDROID}
$ source build/envsetup.sh
$ lunch sabresd_6dq-userdebug
$ make bootimage

# Boot image for the i.MX 6Dual/6Quad/6QuadPlus SABRE-AI board
$ source build/envsetup.sh
$ lunch sabreauto_6q-userdebug
$ make bootimage
```

```
# Boot image for the i.MX 6SoloLite EVK board
$ source build/envsetup.sh
$ lunch evk_6sl-userdebug
$ make bootimage

# Boot image for the i.MX 6SoloX SABRE-SD board
$ source build/envsetup.sh
$ lunch sabresd_6sx-userdebug
$ make bootimage

# Boot image for the i.MX 6SoloX SABRE-AI board
$ source build/envsetup.sh
$ lunch sabreauto_6sx-userdebug
$ make bootimage

# Boot image for i.MX 7Dual SABRE-SD board
$ source build/envsetup.sh
$ lunch sabresd_7d-userdebug
$ make bootimage
```

4 Running the Android Platform with a Prebuilt Image

To test the Android platform before building any code, use the prebuilt images from the following packages and go to "Download Images" and "Boot".

Table 7. Image packages

Image package	Description
android_O8.0.0_1.0.0_image_6qsabresd.tar.gz	Prebuilt image for i.MX 6Dual/6Quad, i.MX 6QuadPlus, and i.MX 6DualLite SABRE-SD board, which includes extended features.
android_O8.0.0_1.0.0_image_6qsabreauto.tar.gz	Prebuilt image for i.MX 6Dual/6Quad, i.MX 6QuadPlus, and i.MX 6DualLite/Solo SABRE-AI board, which includes extended features.
android_O8.0.0_1.0.0_image_6slevk.tar.gz	Prebuilt image for the i.MX 6SoloLite EVK board, which includes extended features.
android_O8.0.0_1.0.0_image_6sxsabresd.tar.gz	Prebuilt image for the i.MX 6SoloX SABRE-SD board, which includes extended features.
android_O8.0.0_1.0.0_image_6sxsabreauto.tar.gz	Prebuilt image for i.MX 6SoloX SABRE-AI board, which includes extended features.
android_O8.0.0_1.0.0_image_7dsabresd.tar.gz	Prebuilt image for i.MX 7Dual SABRE-SD board, which includes extended features.

The following tables list the detailed contents of android_O8.0.0_1.0.0_image_6qsabresd.tar.gz image packages.

The table below shows the prebuilt images to support the system boot from eMMC and SD cards on the i.MX 6Dual/6Quad SABRE-SD board and platform and i.MX 6Solo/6DualLite SABRE-SD platform.

Table 8. Images for i.MX 6 SABRE-SD board and platform eMMC boot

SABRE-SD eMMC image	Description
/u-boot-imx6q.imx	The bootloader (with padding) for 800 MHz or 1 GHz i.MX 6Dual/6Quad SABRE-SD board
/u-boot-imx6dl.imx	The bootloader (with padding) for i.MX 6Solo/6DualLite SABRE-SD board

Table continues on the next page...

Table 8. Images for i.MX 6 SABRE-SD board and platform eMMC boot (continued)

/u-boot-imx6q-ldo.imx	The bootloader (with padding) for 1.2 GHZ i.MX6Dual/6Quad SABRE-SD board
/u-boot-imx6qp.imx	The bootloader (with padding) for i.MX 6Solo/6DualLite SABRE-SD board
/u-boot-imx6qp-ldo.imx	The bootloader (with padding) for 1.2 GHZ i.MX6QuadPlus SABRE-SD board
/partition-table.img	GPT table Image for 8 GB SD card
/partition-table-14GB.img	GPT table Image for 16 GB SD card
/partition-table-28GB.img	GPT table Image for 32 GB SD card
/boot-imx6dl.img	Boot Image for i.MX 6QuadPlus SABRE-SD board
/boot-imx6q.img	Boot Image for 800 MHz or 1 GHz i.MX 6Dual/6Quad SABRE-SD board
/boot-imx6q-ldo.img	Boot Image for 1.2 GHZ i.MX 6Dual/6Quad SABRE-SD board
/boot-imx6qp.img	Boot Image for i.MX 6QuadPlus SABRE-SD board
/boot-imx6qp-ldo.img	Boot Image for 1.2 GHZ i.MX 6QuadPlus SABRE-SD board
/system.img	System Boot Image
/recovery-imx6dl.img	Recovery Image for i.MX 6Solo/6DualLite SABRE-SD board
/recovery-imx6q.img	Recovery Image for 800 MHz or 1 GHz i.MX 6Dual/6Quad SABRE-SD board
/recovery-imx6q-ldo.img	Recovery Image for 1.2 GHZ i.MX 6Dual/6Quad SABRE-SD board
/recovery-imx6qp.img	Recovery Image for i.MX 6QuadPlus SABRE-SD board
/recovery-imx6qp-ldo.img	Recovery Image for 1.2 G Hz i.MX 6QuadPlus SABRE-SD board
/vendor.img	Vendor Image

The following tables list the detailed contents of android_O8.0.0_1.0.0_image_6qsabreauto.tar.gz image packages.

The table below shows the prebuilt images to support the system boot from SD on the i.MX 6Dual/6Quad, i.MX 6QuadPlus, and i.MX 6Solo/6DualLite SABRE-AI boards.

Table 9. Images for i.MX 6 SABRE-AI SD boot

SABRE-AI SD image	Description
/u-boot-imx6q.imx	Bootloader (with padding) for i.MX 6Dual/6Quad SABRE-AI SD boot
/u-boot-imx6dl.imx	Bootloader (with padding) for i.MX 6Solo/6DualLite SABRE-AI SD boot
/u-boot-imx6qp.imx	Bootloader (with padding) for i.MX 6QuadPlus SABRE-AI SD boot
/partition-table.img	GPT table image for 8 GB SD card
/partition-table-14GB.img	GPT table Image for 16 GB SD card
/partition-table-28GB.img	GPT table Image for 32 GB SD card
/boot-imx6q.img /boot-imx6dl.img	Boot Image for SD

Table continues on the next page...

Table 9. Images for i.MX 6 SABRE-AI SD boot (continued)

/boot-imx6qp.img	
/system.img	System Boot Image
/recovery-imx6q.img /recovery-imx6dl.img /recovery-imx6qp.img	Recovery Image
/vendor.img	Vendor Image

The following tables list the detailed contents of android_O8.0.0_1.0.0_image_6slevk.tar.gz image package.

The table below shows the prebuilt images to support the system boot from SD on the i.MX 6SoloLite EVK board.

Table 10. Images for i.MX 6SoloLite EVK SD boot

EVK SD image	Description
/u-boot-imx6sl.imx	Bootloader (with padding) for the i.MX 6SoloLite EVK board
/partition-table.img	GPT table image for 8 GB SD card
/partition-table-14GB.img	GPT table Image for 16 GB SD card
/partition-table-28GB.img	GPT table Image for 32 GB SD card
/boot-imx6sl.img	Boot image for SD
/system.img	System boot image
/recovery-imx6sl.img	Recovery image
/vendor.img	Vendor Image

The following tables list the detailed contents of the android_O8.0.0_1.0.0_image_6sxsabresd.tar.gz image package.

The table below shows the prebuilt images to support the system boot from SD on the i.MX 6SoloX SABRE-SD board.

Table 11. Images for i.MX 6SoloX SABRE-SD SD boot

i.MX 6SoloX SABRE-SD SD image	Description
/u-boot-imx6sx.imx	Bootloader (with padding) for the i.MX 6SoloX SABRE-SD board
/partition-table.img	GPT table image for 8 GB SD card
/partition-table-14GB.img	GPT table Image for 16 GB SD card
/partition-table-28GB.img	GPT table Image for 32 GB SD card
/boot-imx6sx.img	Boot image for the i.MX 6SoloX SABRE-SD board
/system.img	System boot image
/recovery-imx6sx.img	Recovery image for the i.MX 6SoloX SABRE-SD board
/vendor.img	Vendor Image

The following tables list the detailed contents of android_O8.0.0_1.0.0_image_6sxsabreauto.tar.gz image packages.

The table below shows the prebuilt images to support the system boot from SD on the i.MX 6SoloX SABRE-AI boards.

Table 12. Images for i.MX 6SoloX SABRE-AI SD

i.MX 6SoloX SABRE-AI SD image	Description
/u-boot-imx6sx.imx	Bootloader (with padding) for i.MX 6SoloX SABRE-AI SD boot.
/partition-table.img	GPT table image for 8 GB SD card
/partition-table-14GB.img	GPT table Image for 16 GB SD card
/partition-table-28GB.img	GPT table Image for 32 GB SD card
/boot-imx6sx.img	Boot image for SD.
/system.img	System boot Image.
/recovery-imx6sx.img	Recovery image.
/vendor.img	Vendor Image

The following table lists the detailed contents of android_O8.0.0_1.0.0_image_7dsabresd.tar.gz image package.

The table below shows the prebuilt images to support the system boot from SD on i.MX 7Daul SABRE-SD boards.

Table 13. Images for i.MX 7Dual SABRE-SD SD boot

i.MX 7Dual SABRE-SD SD image	Description
/u-boot-imx7d.imx	Bootloader (with padding) for the i.MX 7Dual SABRE-SD board
/partition-table.img	GPT table image for 8 GB SD card
/partition-table-14GB.img	GPT table Image for 16 GB SD card
/partition-table-28GB.img	GPT table Image for 32 GB SD card
/boot-imx7d.img	Boot image for the i.MX 7Dual SABRE-SD board
/system.img	System boot image
/recovery-imx7d.img	Recovery image for the i.MX 7Dual SABRE-SD board
/vendor.img	Vendor Image

NOTE

boot.img is an Android image that stores zImage and ramdisk together. It also stores other information such as the kernel boot command line, machine name.

This information can be configured in android.mk. It is used to override any bootloader default boot arguments without changing it in the bootloader source code.

5 Programming Images

The images from the prebuilt release package or created from source code contain the U-Boot boot loader, system image, and recovery image. At a minium, the storage devices on the development system (MMC/SD or NAND) must be programmed with the U-Boot boot loader. The i.MX 6 series boot process determines what storage device to access based on the switch settings. When the boot loader is loaded and begins execution, the U-Boot environment space is then read to determine how to proceed with the boot process. For U-Boot environment settings, see Section Booting.

The following download methods can be used to write the Android System Image:

- MFGTool or fsl-sdcard-partition.sh to download all images to MMC/SD card.

5.1 System on MMC/SD

The images needed to create an Android system on MMC/SD can either be obtained from the release package or be built from source.

The images needed to create an Android system on MMC/SD are listed below:

- U-Boot image: u-boot.imx
- boot image: boot.img
- Android system image: system.img
- Recovery image: recovery.img
- GPT table image: partition-table.img
- Vendor image: vendor.img

5.1.1 Storage partitions

To create storage partitions, use MFGTool as described in the *Android Quick Start Guide* (AQSUG), or use format tools in the prebuild directory.

The layout of the MMC/SD/TF card for Android system is shown below:

- [Partition type/index] which is defined in the MBR.
- [Name] is only meaningful in the Android platform. Ignore it when creating these partitions.
- [Start Offset] shows where partition is started, unit in MB.

The SYSTEM partition is used to put the built Android system image. The DATA is used to put applications' unpacked codes/data, system configuration database, etc. In normal boot mode, the root file system is mounted from uramdisk. In recovery mode, the root file system is mounted from the RECOVERY partition.

Table 14. Storage partitions

Partition type/index	Name	Start offset	Size	File system	Content
N/A	BOOT Loader	1 KB	1 MB - 1K	N/A	bootloader
1	Boot	1 MB	32 MB	boot.img format, kernel + ramdisk	boot.img
2	Recovery	Follow Boot	32 MB	boot.img format, kernel + ramdisk	recovery.img
3	SYSTEM	Follow Recovery	1536 MB	EXT4. Mount as / system	Android system files under / system/ dir
4	CACHE	Follow SYSTEM.	512 MB	EXT4. Mount as / cache.	Android cache for image store of OTA
5	Misc	Follow CACHE	4 MB	N/A	For recovery store bootloader message, reserve
6	DATAFOOTE R	Follow Misc	2 MB	N/A	For crypto footer of DATA partition encryption
7	METADATA	Follow Dtafootor	2 MB	N/A	For system slide show
8	PRESISTER	Follow Metadata	1 MB	N/A	Option to operate unlock \unlock

Table continues on the next page...

Table 14. Storage partitions (continued)

Partition type/index	Name	Start offset	Size	File system	Content
9	VENDOR	Follow PRESISTER	112 MB	EXT4. Mount as / vendor	vendor.img
10	DATA	Follow VENDOR	Total - Other images	EXT4. Mount at /data	Application data storage for system application, and for internal media partition, in /mnt/sdcard/ dir
11	FBMISC	Follow Data	1 MB	N/A	To store the state of lock \unlock

To create these partitions, use MFGTool described in the *Android Quick Start Guide* (AQSUG), or use script fsl-sdcard-partition.sh in the source code directory.

The script below can be used to partition an SD card as shown in the partition table above:

```
$ cd ${MY_ANDROID}/
$ sudo ./device/fsl/common/tools/fsl-sdcard-partition.sh -f <soc_name> /dev/sdX
# <soc_name> can be as imx6qp, imx6q, imx6dl, imx6sl, imx6sx and imx7d.
```

NOTE

- The minimum size of the SD card is 8 GB.
- If the SD card is 8 GB, use `sudo ./device/fsl/common/tools/fsl-sdcard-partition.sh -f <soc_name> /dev/sdX` to flash images.
- If the SD card is 16 GB, use `sudo ./device/fsl/common/tools/fsl-sdcard-partition.sh -f <soc_name> -c 14 /dev/sdX` to flash images.
- If the SD card is 32 GB, use `sudo ./device/fsl/common/tools/fsl-sdcard-partition.sh -f <soc_name> -c 28 /dev/sdX` to flash images.
- /dev/sdX, the X is the disk index from 'a' to 'z'. That may be different on each computer running Linux OS.
- Unmount all the SD card partitions before running the script.
- Put the related bootloader, boot image, system image, recovery image, GPT table image, and vendor image in your current directory. This script requires to install the simg2img tool on the computer. simg2img is a tool that converts the sparse system image to raw system image on the Linux OS host computer. The android-tools-fsutils package includes the simg2img command for Ubuntu Linux OS.

5.1.2 Downloading images with MFGTool

MFGTool can be used to download all images into a target device. It is a quick and easy tool for downloading images. See *Android™ Quick Start Guide* (AQSUG) for a detailed description of MFGTool.

6 Booting

This chapter describes booting from MMC/SD, TFTP and NFS.

6.1 Booting from MMC/SD

This section describes how to boot from MMC/SD on the SABRE and EVK development tool devices:

- i.MX 6DualQuad/6DualLite/6QuadPlus SABRE-SD board/platform
- i.MX 6DualQuad/6DualLite/6QuadPlus SABRE-AI board
- i.MX 6SoloLite EVK board
- i.MX 6SoloX SABRE-SD board
- i.MX 6SoloX SABRE-AI board
- i.MX 7Dual SABRE-SD board

6.1.1 Booting from MMC/SD on the i.MX 6QuadPlus/6Quad/6DualLite SABRE-SD board

This section contains boot switch information and steps needed to bootup from MMC/SD.

The following table lists the boot switch settings for different boot methods.

Table 15. Boot switch settings

Download mode (MFGTool mode)	(SW6) 00001100 (from 1-8 bit)
eMMC 4-bit (MMC2) boot	(SW6) 11100110 (from 1-8 bit)
eMMC 8-bit (MMC2) boot	(SW6) 11010110 (from 1-8 bit)
SD2 boot	(SW6) 10000010 (from 1-8 bit)
SD3 boot	(SW6) 01000010 (from 1-8 bit)

To boot from eMMC, perform the following operations:

Change the board boot switch to eMMC 4-bit mode and make (SW6) 11100110 (from 1-8 bit). Or change (SW6) 11100110 (from 1-8 bit) for 8-bit boot mode.

The default environment in boot.img is booting from eMMC. To use the default environment in boot.img, use the following command:

```
U-Boot > setenv bootargs
```

To clear the bootargs environment, use the following commands:

```
U-Boot > setenv fastboot_dev mmc2 [eMMC as fastboot deivce]
U-Boot > setenv bootcmd boota mmc2      [Load the boot.img from eMMC]
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init video=mxcfb0:dev=ldb,bpp=32
video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off vmalloc=128M androidboot.console=ttymxc0
consoleblank=0  androidboot.hardware=freescale cma=448M galcore.contiguousSize=33554432
[Optional]
U-Boot > saveenv        #[Save the environments]
```

NOTE

The mmcX value changes depending on the boot mode. These are the correct values:

Hardware port	U-Boot device in software
eMMC	mmc2
SD2	mmc3
SD3	mmc1

bootargs env is an optional setting for boota. The boot.img includes a default bootargs, which is used if there is no definition of the bootargs environment.

Some SoCs on SABRE-SD boards do not have MAC address fused. Therefore, to use FEC in U-Boot, set the following environment:

```
U-Boot > setenv ethaddr 00:04:9f:00:ea:d3            #[setup the
MAC address]
U-Boot > setenv fec_addr 00:04:9f:00:ea:d3           $[setup the
MAC address]
```

To boot from the SD card, perform the following operations:

Change the board boot switch to SD3 boot: (SW6) 01000010 (from 1-8 bit). To clear the bootargs env and set up the booting from SD card in SD slot 3, use the following command:

```
U-Boot > setenv fastboot_dev mmc1  [eMMC as fastboot deivce]
U-Boot > setenv bootcmd boota mmc1         [Load the boot.img from SD card]
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init video=mxcfb0:dev=ldb,bpp=32
video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off vmalloc=128M androidboot.console=ttymxc0
consoleblank=0  androidboot.hardware=freescale cma=448M galcore.contiguousSize=33554432
#[Optional]
U-Boot > saveenv    #[Save the environments]
```

NOTE

bootargs env is an optional setting for booti. The boot.img includes a default bootargs, which is used if there is no definition of the bootargs environment.

Some SoCs on SABRE-SD boards do not have MAC address fused.

Therefore, to use FEC in U-Boot, set the following environment:

```
U-Boot > setenv ethaddr 00:04:9f:00:ea:d3            #[setup the
MAC address]
U-Boot > setenv fec_addr 00:04:9f:00:ea:d3           $[setup the
MAC address]
```

6.1.2 Booting from MMC/SD on the the i.MX 6Dual/Quad/DualLite/QuadPlus SABRE-AI board

This section contains boot switch information and steps needed to bootup from MMC/SD.

The following table lists the boot switch settings for different boot methods on i.MX 6 series SABRE-AI boards.

Table 16. Boot switch settings

Download mode (MFGTool mode)	(S3) 0101 (from 1-4 bit)
SD on CPU Board	(S1) 0100100000 (from 1-10 bit) (S2) 0010 (from 1-4 bit) (S3) 0010 (from 1-4 bit)

To boot from SD, perform the following operations:

Change the board boot switch to (S3, S2, S1) 0010, 0010,0100100000 (from 1 bit).

The default environment in boot.img is booting from SD. To use the default environment in boot.img, use the following command:

```
U-Boot > setenv bootargs
```

To clear the bootargs environment, use the following command:

```
U-Boot > setenv bootcmd boota mmc1      #[Load the boot.img from SD]
U-Boot > setenv bootargs console=ttymxc3,115200 init=/init video=mxcfb0:dev=ldb,bpp=32
video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off vmalloc=128M androidboot.console=ttymxc3
consoleblank=0 androidboot.hardware=freescale cma=512M  #[Optional]
U-Boot > saveenv   #[Save the environments]
```

NOTE

bootargs environment is an optional setting for Android boot. The boot.img includes a default bootargs, which is used if there is no definition of the bootargs environment.

Some SoCs on SABRE-AI boards do not have MAC address fused. Therefore, to use FEC in U-Boot, set the following environment:

```
U-Boot > setenv ethaddr 00:04:9f:00:ea:d3            #[setup the
MAC address]
U-Boot > setenv fec_addr 00:04:9f:00:ea:d3           $[setup the
MAC address]
```

6.1.3 Booting from SD on the i.MX 6SoloLite EVK board

The following table lists the Boot switch settings used to control the boot storage.

Table 17. Boot switch settings

Download mode (MFGTool mode)	SW3: 01000000(from 1-8 bit) SW4: 00101100 (from 1-8 bit) SW5: 00000000(from 1-8 bit) Boot_Mode: 10 (from 1-2 bit)
SD boot	SW3: 01000000(from 1-8 bit) SW4: 00101100 (from 1-8 bit) SW5: 00000000(from 1-8 bit) Boot_Mode: 01 (from 1-2 bit)

To boot from SD, perform the following operations:

Change the board Boot_Mode switch to 01 (from 1-2 bit) and (SW3,4,5) 01000000 0010110000000000 (from 1-8 bit).

The default environment in boot.img is booting from SD. To use default enviroment in boot.img, use the following command:

```
U-Boot > setenv bootargs
```

To clear the bootargs environment, use the following command:

```
U-Boot > setenv bootcmd boota mmc1
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale   #[Optional]
U-Boot > saveenv                      [Save the environments]
```

NOTE

bootargs environment is an optional setting for Android boot. The boot.img file includes a default bootargs, which is used if there is no definition about the bootargs env.

Due to some SoCs on the EVK boards, do not fuse MAC address. Set the following environment to use FEC in U-Boot:

```
U-Boot > setenv ethaddr 00:04:9f:00:ea:d3       [setup the MAC
address]
U-Boot > setenv fec_addr 00:04:9f:00:ea:d3      [setup the MAC
address]
```

6.1.4 Booting from SD on the i.MX 6SoloX SABRE-SD board

This section contains boot switch information and steps needed to bootup from SD.

The following table lists the boot switch settings used to control the boot storage.

Table 18. Boot switch settings

Download mode (MFGTool mode)	SW10: 00000000 (from 1-8 bit) SW11: 00111000 (from 1-8 bit) SW12: 01000000 (from 1-8 bit) Boot_Mode: 10 (from 1-2 bit)
SD boot	SW3: 00000000 (from 1-8 bit) SW4: 00111000 (from 1-8 bit) SW5: 01000000 (from 1-8 bit) Boot_Mode: 01 (from 1-2 bit)

To boot from SD, perform the following operations:

Change the board Boot_Mode switch to 01 (from 1-2 bit) and (SW10, 11, 12) 00000000 00111000 01000000 (from 1-8 bit).

The default environment in boot.img is booting from SD. To use the default environment in boot.img, use the following command:

```
U-Boot > setenv bootargs
```

To clear the bootargs environment, use the following command:

```
U-Boot > setenv bootcmd boota mmc2
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale vmalloc=128M cma=448M
galcore.contiguousSize=33554432     [Optional]
U-Boot > saveenv                         [Save the environments]
```

NOTE

bootargs environment is an optional setting for Android boot. The boot.img file includes a default bootargs, which is used if there is no definition about the bootargs env.

Due to some SoCs on the SABRE-SD boards, do not fuse MAC addres. Set the following environment to use FEC in U-Boot:

```
U-Boot > setenv ethaddr 00:04:9f:00:ea:d3       [setup the MAC
address]
U-Boot > setenv fec_addr 00:04:9f:00:ea:d3      [setup the MAC
address]
```

6.1.5 Booting from SD on the i.MX 6SoloX SABRE-AI board

The following table lists the boot switch settings used to control the boot storage.

Table 19. Boot switch settings

Download mode (MFGTool mode)	SW3: 00001100 (from 1-8 bit)

Table continues on the next page...

Table 19. Boot switch settings (continued)

	SW4: 01000010 (from 1-8 bit) S1 (Boot_Mode): 0101 (from 1-4bit)
SD boot	SW3: 00001100 (from 1-8 bit) SW4: 01000010 (from 1-8 bit) S1 (Boot_Mode): 0010 (from 1-4bit)

To boot from SD, perform the following operations:

Change the board S1 (Boot_Mode) switch to 0010 (from 1-4bit) and (SW3, SW4) 00001100 01000010 (from 1-8 bit).

The default environment in boot.img is booting from SD. To use default env in boot.img, use the following command:

```
U-Boot > setenv bootargs
```

To clear the bootargs environment, use the following command:

```
U-Boot > setenv bootcmd boota mmc0
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale  vmalloc=128M cma=448M
galcore.contiguousSize=33554432 [Optional
U-Boot > saveenv                        [Save the environments]
```

NOTE

bootargs environment is an optional setting for Android boot. The boot.img file includes a default bootargs, which is used if there is no definition about the bootargs env.

Due to some SoCs on the SABRE-SD boards, do not fuse MAC addres. Set the following environment to use FEC in U-Boot:

```
U-Boot > setenv ethaddr 00:04:9f:00:ea:d3       [setup the MAC
address]
U-Boot > setenv fec_addr 00:04:9f:00:ea:d3      [setup the MAC
address]
```

6.1.6 Booting from SD on the i.MX 7Dual SABRE-SD board

The following table lists the boot switch settings used to control the boot storage.

Table 20. Boot switch settings

Download mode (MFGTool mode)	SW4: 00100000 (from 1-8 bit) Boot_Mode: 01 (from 1-2 bit)
SD boot	SW4: 00111000 (from 1-8 bit) Boot_Mode: 10 (from 1-2 bit)

To boot from SD, perform the following operations:

Change the board Boot_Mode switch to 10 (from 1-2 bit) and SW4 00100000 (from 1-8 bit).

The default environment in boot.img is booting from SD. To use the default environment in boot.img, use the following command:

```
U-Boot > setenv bootargs
```

To clear the bootargs environment, use the following command:

```
U-Boot > setenv bootcmd boota mmc0
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale     [Optional]
U-Boot > saveenv                         [Save the environments]
```

NOTE

bootargs environment is an optional setting for Android boot. The boot.img file includes a default bootargs, which is used if there is no definition about the bootargs env.

Due to some SoCs on the SABRE-SD boards, do not fuse MAC addres. Set the following environment to use FEC in U-Boot:

```
U-Boot > setenv ethaddr 00:04:9f:00:ea:d3       [setup the MAC
address]
U-Boot > setenv fec_addr 00:04:9f:00:ea:d3      [setup the MAC
address]
```

6.2 Boot-up configurations

This section explains the common U-Boot environments used for NFS, MMC/SD boot, and kernel command line.

6.2.1 U-Boot environment

If you do not define the bootargs environment, it uses the default bootargs inside the image.

- bootcmd is the first variable to run after U-Boot boot.
- bootargs is the kernel command line, which the bootloader passes to the kernel. As described in Kernel command line (bootargs), bootargs environment is optional for booti. boot.img already has bootargs. If you do not define the bootargs environment, it uses the default bootargs inside the image.

  To use the default environment in boot.img, use the following command to clear the bootargs environment.

  ```
  > setenv bootargs
  ```
- dhcp: get the IP address by BOOTP protocol, and load the kernel image ($bootfile env) from the TFTP server.
- booti:

  booti command parses the boot.img header to get the zImage and ramdisk. It also passes the bootargs as needed (it only passes bootargs in boot.img when it cannot find "bootargs" var in your U-Boot environment). To boot from mmcX, do the following:

  ```
  > booti mmcX
  ```

  To read the boot partition (the partition store boot.img, in this instance, mmcblk0p1), the X is the MMC bus number, which is the hardware MMC bus number, in i.MX 6Dual/6Quad SABRE-SD and i.MX 6Solo/6DualLite SABRE-SD boards. eMMC is mmc3.

  Add partition ID after mmcX.

  ```
  > boota mmcX boot   #  boot is default
  > boota mmcX recovery      # boot from the recovery partition
  ```

  If you have read the boot.img into memory, use this command to boot from

  ```
  > boota 0xXXXXXXXX
  ```
- bootm (only works in for the NFS) starts running the kernel. For other use cases, use booti command.

6.2.2 Kernel command line (bootargs)

Depending on the different booting/usage scenarios, you may need different kernel boot parameters set for bootargs.

Table 21. Kernel boot parameters

Kernel parameter	Description	Typical value	Used when
console	Where to output kernel log by printk.	console=ttymxc0,115200	All use cases.
init	Tells kernel where the init file is located.	init=/init	All use cases. "init" in the Android platform is located in "/" instead of in "/sbin".
androidboot.hardware	Specifies hardware name for this product. Android init process loads the configuration file init.$(androidboot.hardware).rc in the root directory.	androidboot.hardware=freescale	All use cases.

video	Tells kernel/driver which resolution/depth and refresh rate should be used, or tells kernel/driver not to register a framebuffer device for a display device.	video=mxcfb0:dev=ldb,LDB-XGA,if=RGB666,bpp=32
		or
		video=mxcfb1:dev=hdmi, 1920x1080M@60,if=RGB24,bpp=32
		or
		video=mxcfb2:off
			To specify a display framebuffer with:
			video=mxcfb<0,1,2>:dev=<ldb,hdmi>,<LDB-XGA,xres x yresM@fps>,if=<RGB666,RGB24>,bpp=<16,32>
			or
			To disable a display device's framebuffer register with:
			video=mxcfb<0,1,2>:off

vmalloc	vmalloc virtual range size for kernel.	vmalloc=256M	vmalloc=<size>
androidboot.console	The Android shell console. It should be the same as console=.	androidboot.console=ttymxc0	To use the default shell job control, such as Ctrl+C to terminate a running process, set this for the kernel.
fec_mac	Sets up the FEC MAC address.	fec_mac=00:04:9f:00:ea:d3	On SABRE-SD board, the SoC does not have a MAC address fused in. To use FEC, assign this parameter to the kernel.
cma	CMA memory size for GPU/VPU physical memory allocation.	cma=448M galcore.contiguousSize=33554432	It is 448 MB by default.
androidboot.selinux	Argument to disable selinux check and enable serial input when connecting a host	androidboot.selinux=permissive	Android Oreo 8.0 CTS requirement: The serial input should be disabled by default. Setting this argument enables console serial input, which violates the CTS requirement.

Table 21. Kernel boot parameters

Kernel parameter	Description	Typical value	Used when
	computer to the target board's USB UART port. For details about selinux, see Security-Enhanced Linux in Android.		

7 Revision History

Table 22. Revision history

Revision number	Date	Substantive changes
O8.0.0_1.0.0	02/2018	Initial release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AUG
Rev. O8.0.0_1.0.0
02/2018