NXP Semiconductors

Document Number: AQSUG
Rev. N7.1.1_1.0.0, 03/2017

Android™ Quick Start Guide

1 Overview

This document guides you through the processes of downloading and running this release package. It only explains how to download and run the default release image with default configuration. For details on using the release package, see the *Android™ User's Guide* (AUG) included in this release package.

2 Hardware Requirements

The hardware requirements for using this release package are as follows:

Supported system-on-chips (SoCs):

- i.MX 6Dual/6Quad
- i.MX 6Solo/6DualLite
- i.MX 6SoloLite
- i.MX 6SoloX
- i.MX 7Dual
- i.MX 6QuadPlus

Supported boards:

- SABRE-SD board and platform
- SABRE-AI board
- EVK board

Contents

3 Working with the i.MX 6Quad/6QuadPlus/6DualLite SABRE-SD Board and Platform

3.1 Board hardware

The figure below shows the different components of the SABRE-SD board.

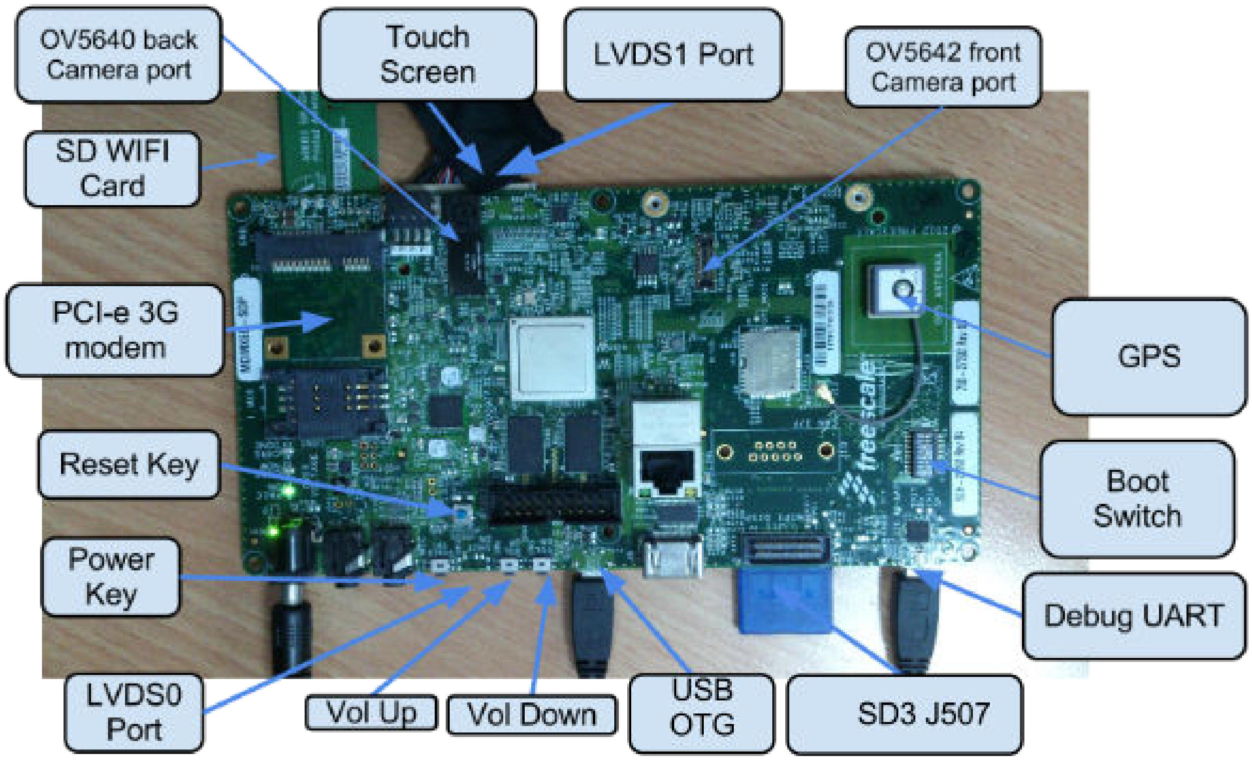


Figure 1. SABRE-SD Board

3.2 Board images

The table below describes the locations of the software images in release_package/android_N7.1.1_1.0.0_image_6dqpsabresd.tar.gz on board partitions.

Table 1. Board images

Image name	Path in release package	Download target
u-boot-imx6q.imx	\	eMMC or SD card first 8 MB section
u-boot-imx6dl.imx	\	eMMC or SD card first 8 MB section
u-boot-imx6qp.imx	\	eMMC or SD Card first 8 MB section
u-boot-imx6q-ldo.imx	\	eMMC or SD card first 8 MB section

Table continues on the next page...

Table 1. Board images (continued)

boot-imx6qp.img	\eMMC	eMMC first partition for i.MX 6QuadPlus
boot-imx6qp.img	\SD	SD Card first partition for i.MX 6QuadPlus
recovery-imx6qp.img	\eMMC	eMMC second partition for i.MX 6QuadPlus
recovery-imx6qp.img	\SD	SD Card second partition for i.MX 6QuadPlus
boot-imx6q.img	\eMMC	eMMC first partition for 800 MB or 1 GB Hz i.MX 6Dual/6Quad
boot-imx6q-ldo.img	\eMMC	eMMC first partition for 1.2 G Hz i.MX 6Dual/6Quad
boot-imx6dl.img	\eMMC	eMMC first partition for i.MX 6DualLite
boot-imx6q.img	\SD	SD card first partition for 800 MB or 1 GB Hz i.MX 6Dual/6Quad
boot-imx6q-ldo.img	\SD	SD card first partition for 1.2 GB Hz i.MX 6Dual/6Quad
boot-imx6dl.img	\SD	SD card first partition for i.MX 6DualLite
recovery-imx6q.img	\eMMC	eMMC second partition for 800 MB or 1 GB Hz i.MX 6Dual/6Quad
recovery-imx6q-ldo.img	\eMMC	eMMC second partition for 1.2 GB Hz i.MX 6Dual/6Quad
recovery-imx6dl.img	\eMMC	eMMC second partition for i.MX 6DualLite
recovery-imx6q.img	\SD	SD card second partition for 800 MB or 1 GB Hz i.MX 6Dual/6Quad
recovery-imx6q-ldo.img	\SD	SD card second partition for 1.2 GB Hz i.MX 6Dual/6Quad
recovery-imx6dl.img	\SD	SD card second partition for i.MX 6DualLite
system.img	\eMMC	eMMC 5th partition
system.img	\SD	SD card 5th partition

3.3 Flashing board images

The board images can be flashed to the target board by using the MFGTool. The release package includes MFGTool for i.MX 6Dual/6Quad/6QuadPlus, i.MX 6Solo/6DualLite, i.MX 6SoloX, i.MX 6SoloLite, and i.MX 7Dual in android_N7.1.1_1.0.0_tools.tar.gz. The MFGTool is mfgtools.tar.gz.

NOTE

The MFGTool only works in the Windows® Operating System (OS) environment.

Perform the following steps to flash the board images:

NOTE

The steps given below take i.MX 6Dual/6Quad as the example SoC. For i.MX 6Solo/6DualLite, replace 'MX6Q' with 'MX6DL' and '6q' with '6dl. For i.MX 6QuadPlus, replace 'MX6Q' with 'MX6QP' and '6q' with '6qp.'

1. Unzip the mfgtools.tar.gz file to a selected location. The directory is named MFGTool-Dir in this example.
2. If the directory does not exist, create the *android/sabresd* directory under *MFGTool-Dir/Profiles/Linux/OS Firmware/files/*.
3. Copy the following files from either android_N7.1.1_1.0.0_image_6dqpsabresd.tar.gz to the *MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/sabresd* path.

- u-boot-imx6q.imx
- eMMC/boot-imx6q.img
- eMMC/system.img
- eMMC/recovery-imx6q.img

NOTE

Do not replace any other files in the files directory and the OS Firmware directory. To flash images for the i.MX 6DualLite/6Solo SABRE-SD boards, replace the name "imx6q" in step 3 with "imx6dl".

To flash the images for the 1.2 GB Hz i.MX 6Dual/6Quad SABRE-SD boards, replace the name "imx6q" in step 3 with "imx6q-ldo".

4. No dedicated boot dips are reserved for serial flash mode on SABRE-SD board. Therefore, a tricky method is used to enter serial flash mode. Change the SABRE-SD SW6 (boot) to 00001100 (from 1-8 bit) to enter flash mode.
5. Power on the board. Using the USB cable on the SABRE-SD OTG port, connect a computer running the Windows OS to the SABRE-SD board.

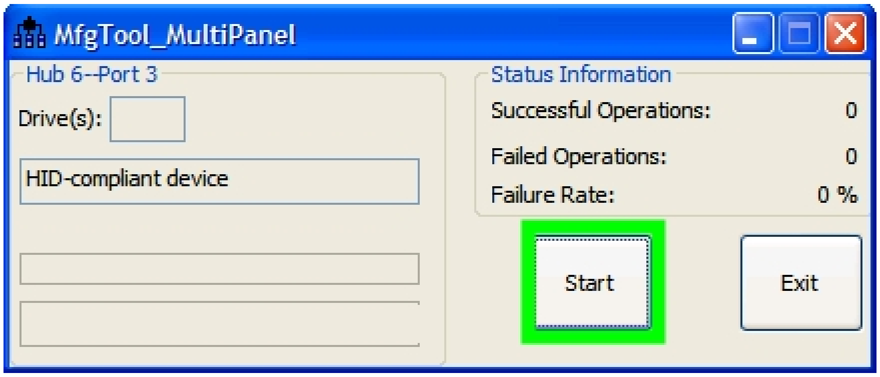
NOTE

There are two USB micro ports on the SABRE-SD board: USB to UART and USB OTG. USB to UART is referred to as debug UART, and the USB OTG is referred to as USB in the hardware image above. The debug UART can be used to monitor the log of the hardware boot processing.

6. Double-click the file *.vbs according to the target device as shown in the following table.

Target device and boot storage	VBS file
i.MX 6Dual/6Quad (800 M Hz or 1 G Hz) SABRE-SD eMMC	mfgtool2-android-mx6q-sabresd-emmc.vbs
i.MX 6Dual/6Quad (1.2 G Hz) SABRE-SD eMMC	mfgtool2-android-mx6q-sabresd-emmc-1.2g.vbs
i.MX 6Dual/6Quad (800 M Hz or 1 G Hz) SABRE-SD SD	mfgtool2-android-mx6q-sabresd-sd.vbs
i.MX 6Dual/6Quad (1.2 G Hz) SABRE-SD SD	mfgtool2-android-mx6q-sabresd-sd-1.2g.vbs
i.MX 6Solo/6DualLite SABRE-SD eMMC	mfgtool2-android-mx6dl-sabresd-emmc.vbs
i.MX 6Solo/6DualLite SABRE-SD SD	mfgtool2-android-mx6dl-sabresd-sd.vbs

7. Click **Start** to start flashing images.



HID-compliant device

Status Information

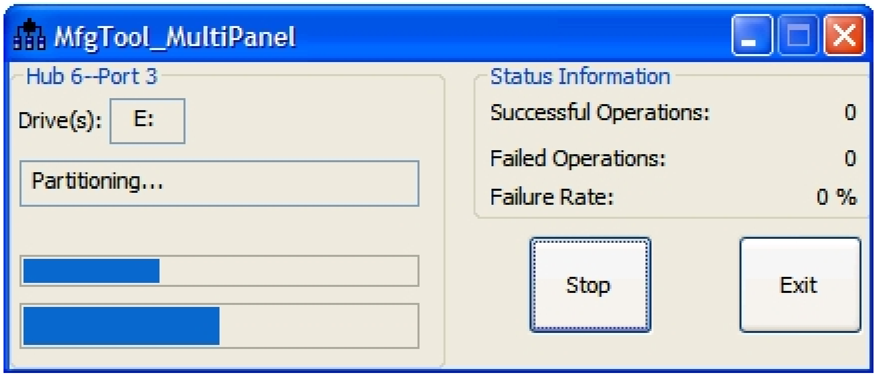
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 2. Starting flash

The figure below shows flashing in progress and the flash status. The flash may take one to two minutes depending on the host machine.



Hub 6--Port 3

Drive(s): E:

Partitioning...

Status Information

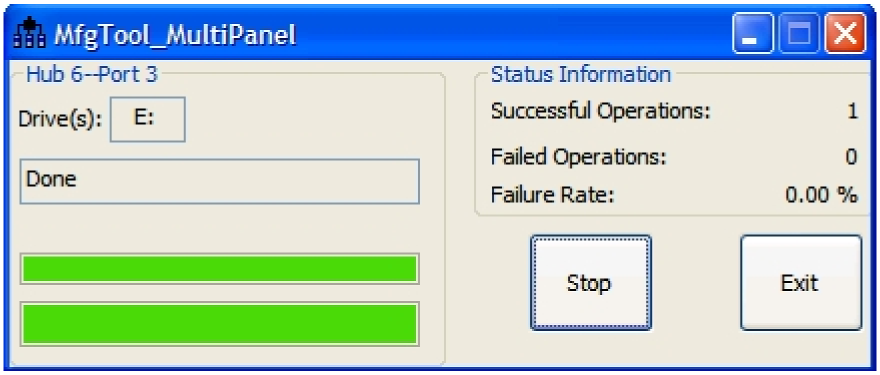
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 3. Download status

The figure below shows the tool once the flash is complete.



Hub 6--Port 3

Drive(s): E:

Done

Status Information

Successful Operations: 1

Failed Operations: 0

Failure Rate: 0.00 %

Figure 4. Download complete

8. Click **Stop**.
9. Change "Boot Switch(SW6)" to 11100110 (from 1 bit to 8 bit) to switch the board back to eMMC 4-bit boot mode.
   Change "Boot Switch(SW6)" to 11010110 (from 1 bit to 8 bit) to switch the board back to eMMC 8-bit boot mode.
   Change "Boot Switch(SW6)" to 01000010 (from 1-8 bit) to switch the board back to SD Card boot mode.

3.4 Booting

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

There are three kinds of display configurations supported in this release between LVDS display pannels and HDMI output.

- To enable the LVDS1 display, see Section Booting with single display: LVDS display.
- To enable single HDMI display, see Section Booting with single display: HDMI display.
- To enable LVDS1 and HDMI output dual display feature, see Section Booting with dual displays: LVDS and HDMI displays.

NOTE

There are two LVDS ports in SABRE SD hardware: LVDS0 and LVDS1. LVDS1 is the primary display in this release. The LVDS1 port is a nearby miniPCIe interface (see the SABRE-SD board image above).

3.4.1 Booting with single display: LVDS display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc2
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
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system information. To avoid this, "androidboot.selinux=disabled" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc2
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 androidboot.selinux=permissive
androidboot.dm_verity=disable
U-Boot > saveenv
```

3.4.2 Booting with single display: HDMI display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc2
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb1:off
video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale cma=448M
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc2
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb1:off
video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale cma=448M
androidboot.selinux=permissive androidboot.dm_verity=disabled
U-Boot > saveenv
```

3.4.3 Booting with dual displays: LVDS and HDMI displays

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc2
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=ldb,bpp=32 video=mxcfb1:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale
cma=448M
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc2
U-Boot > setenv bootargs console=ttymxc0,115200 androidboot.console=ttymxc0 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=ldb,bpp=32 video=mxcfb1:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale
cma=448M androidboot.selinux=permissive androidboot.dm_verity=disabled
U-Boot > saveenv
```

3.5 Board reboot

After completing download and setup, reboot the board and wait for the Android platform to boot up.

Figure 5. Android Nougat image

4 Working with the i.MX 6QuadPlus/6Quad/6DualLite SABRE-AI Platform

4.1 Board hardware

The following figure shows the different components of the i.MX 6Quad/6DualLite SABRE-AI board.

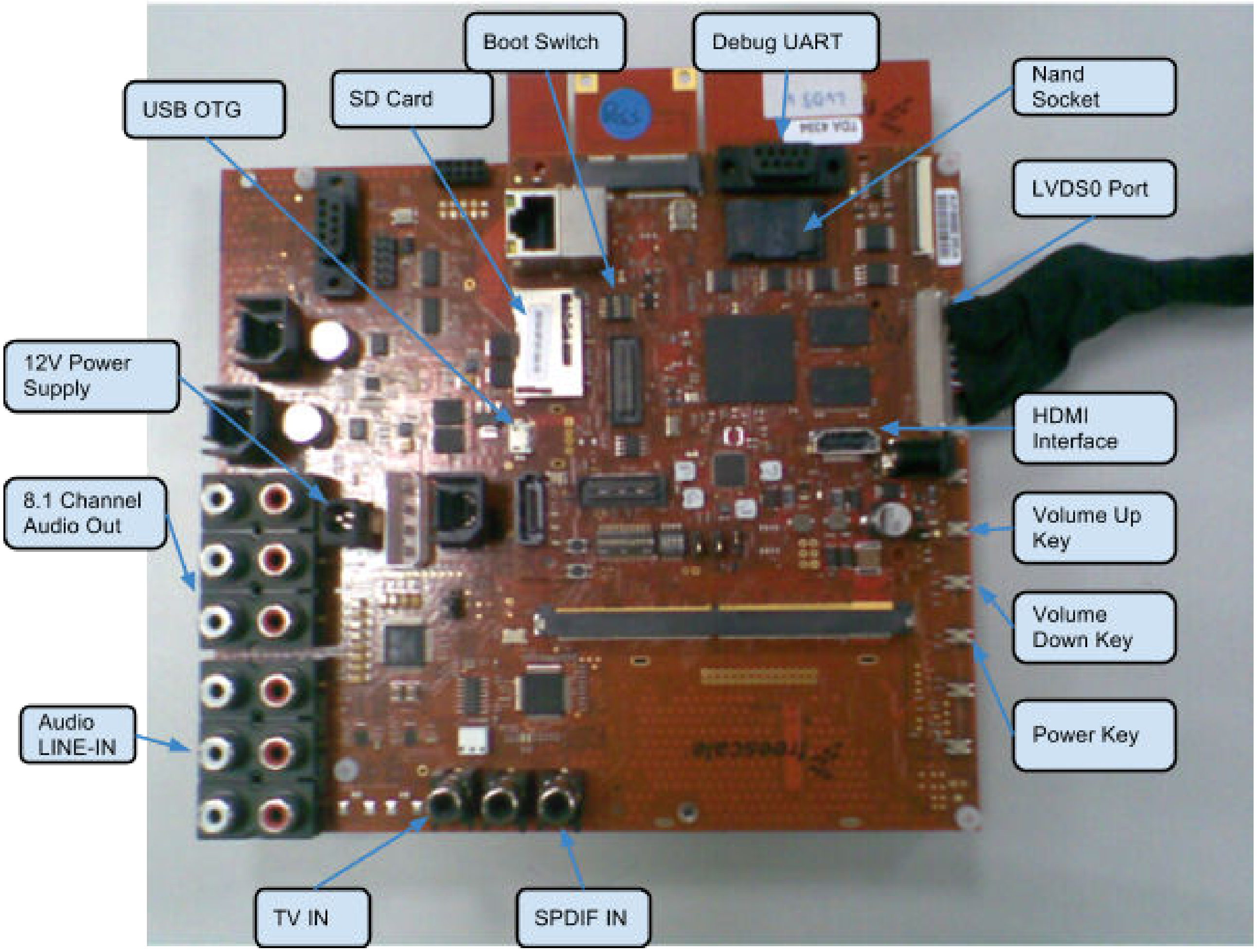


Figure 6. i.MX 6Quad/6DualLite SABRE-AI board

The following figure shows the different components of the i.MX 6QuadPlus SABRE-AI board.

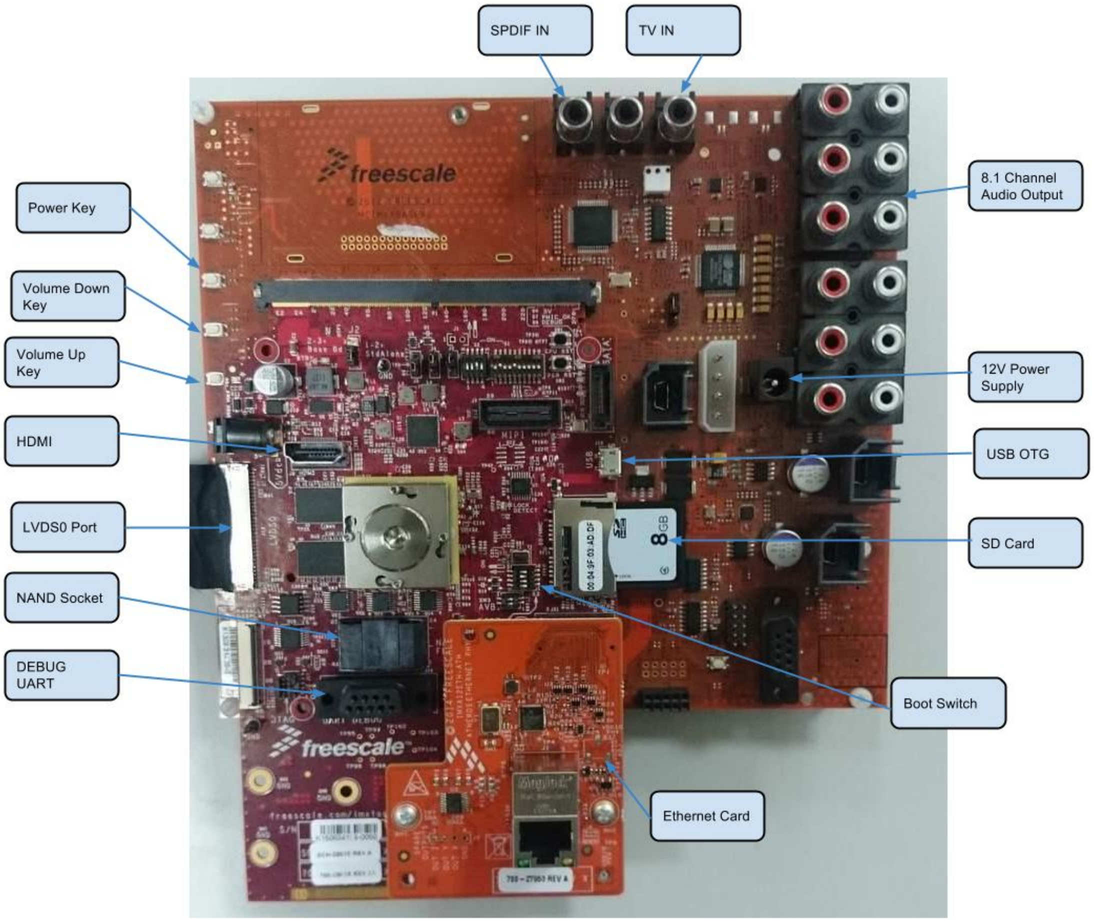


Figure 7. i.MX 6QuadPlus SABRE-AI Board

4.2 Board images

The table below describes the location in the board partitions of the software images in android_N7.1.1_1.0.0_image_6dqpsabreauto.tar.gz on board partitions.

Table 2. Board images

Image name	Path in release package	Download target
u-boot-imx6q.imx u-boot-imx6dl.imx u-boot-imx6qp.imx	\	SD first 8 MB block
boot-imx6q.img boot-imx6dl.img boot-imx6qp.img	\SD	SD 1st partition
recovery-imx6q.img	\SD	SD 2nd partition

Table continues on the next page...

Table 2. Board images (continued)

recovery-imx6dl.img recovery-imx6qp.img		
system.img	\SD	SD 5th partition
u-boot-imx6dl-nand.imx u-boot-imx6q-nand.imx u-boot-imx6qp-nand.imx	\	NAND 1st 64 MB MTD partition
boot-imx6q-nand.img boot-imx6dl-nand.img boot-imx6qp-nand.img	\NAND	NAND 2nd 16 MB MTD partition
recovery-imx6q-nand.img recovery-imx6dl-nand.img recovery-imx6qp-nand.img	\NAND	NAND 3rd 16 MB MTD partition
android_root.img	\NAND	UBIFS Volume for 4th MTD partition

4.3 Flashing board images

The board images can be flashed to the target board by using the MFGTool. The release package includes MFGTool for i.MX 6Dual/6Quad/6QuadPlus, i.MX 6Solo/6DualLite, i.MX 6SoloX, i.MX 6SoloLite, i.MX 6QuadPlus, and i.MX 7Dual in android_N7.1.1_1.0.0_tools.tar.gz. The MFGTool is mfgtools.tar.gz.

NOTE

The MFGTool only works in Windows OS environment.

Perform the following steps to flash the board images:

NOTE

The steps given below take i.MX 6Dual/6Quad as the example SoC. For i.MX 6Solo/6DualLite, replace 'MX6Q' with 'MX6DL' and '6q' with '6dl.' For i.MX 6QuadPlus, replace 'MX6Q' with 'MX6QP' and '6q' with '6qp.'

1. Unzip the mfgtools.tar.gz file to a selected location. The directory is named MFGTool-Dir in this example.
2. If the directory does not exist, create the "*android*" directory under the *MFGTool-Dir/Profiles/Linux/OS Firmware/files* directory.
3. Copy following files from either *release_package/android_N7.1.1_1.0.0_image_6dqpsabreauto.tar.gz* to the *MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/sabreauto* path.
   - u-boot-imx6q.imx
   - SD/boot-imx6q.img
   - SD/system.img
   - SD/recovery-imx6q.img

   NOTE

   Do not replace any other files in the files directory and the OS Firmware directory. When using the NAND boot, the files are in “nand” folder. Use android_root.img instead of system.img. The bootloader should be u-boot-mx6q-nand.bin.

   To flash images for the i.MX 6DualLite/6Solo SABRE-AI boards, replace the name "imx6q" in step 3 with "imx6dl".

To flash images for i.MX 6QuadPlus SABRE-AI boards, replace the name "imx6q" in Step 3 with "imx6qp" for i.MX 6QuadPlus SABRE-AI boards.

4. Change SABRE-AI S3 (boot mode) to 0101 (from 1 bit to 4 bit) to enter flash mode.
5. Power on the board. Using USB cable on the SABRE-AI OTG port, connect a computer running Windows OS to the SABRE-AI board.

NOTE

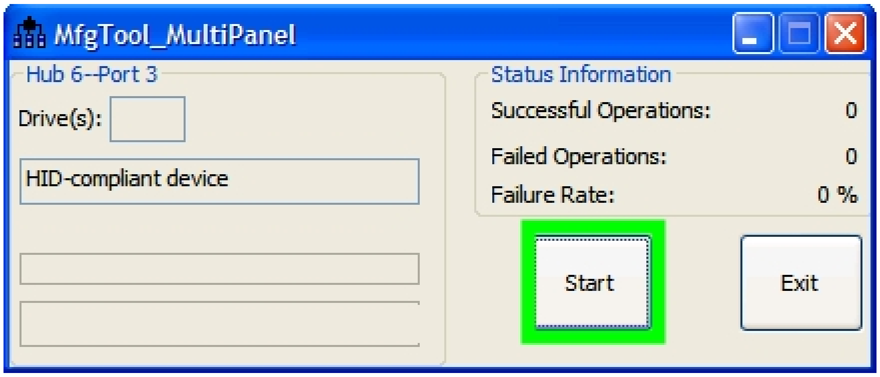
The USB micro port in SABRE-AI is J10.

6. Double-click the *.vbs file according to the target device as shown in the following table.

Table 3. SABRE-AI VBS file

Target device	VBS file
i.MX 6Dual/6Quad SABRE-AI SD	mfgtool2-android-mx6q-sabreauto-sdcard.vbs
i.MX 6Dual/6Quad SABRE-AI NAND	mfgtool2-android-mx6q-sabreauto-nand.vbs
i.MX 6Solo/6DualLite SABRE-AI SD	mfgtool2-android-mx6dl-sabreauto-sdcard.vbs
i.MX 6Solo/6DualLite SABRE-AI NAND	mfgtool2-android-mx6dl-sabreauto-nand.vbs
i.MX 6QuadPlus SABRE-AI SD	mfgtool2-android-mx6qp-sabreauto-sdcard.vbs
i.MX 6QuadPlus SABRE-AI NAND	mfgtool2-android-mx6qp-sabreauto-nand.vbs

7. Click **Start** to start flashing images.



Drive(s):

HID-compliant device

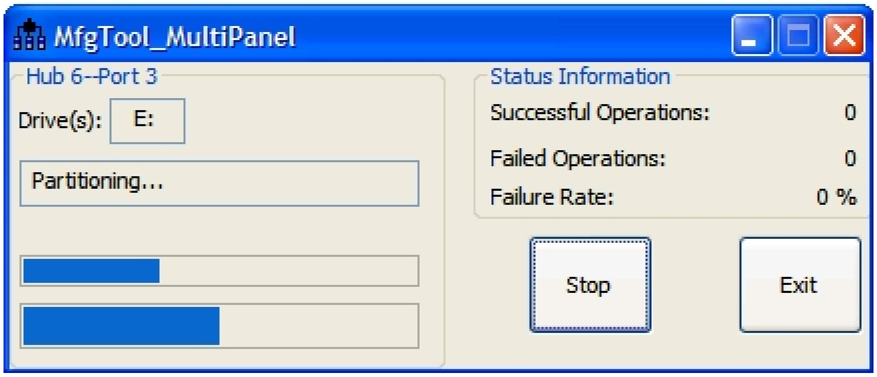
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 8. Starting flash

The figure below shows flashing in progress, and the status bar shows the flash status. The flash may take one to two minutes depending on the host machine.



Hub 6--Port 3

Drive(s): E:

Partitioning...

Status Information

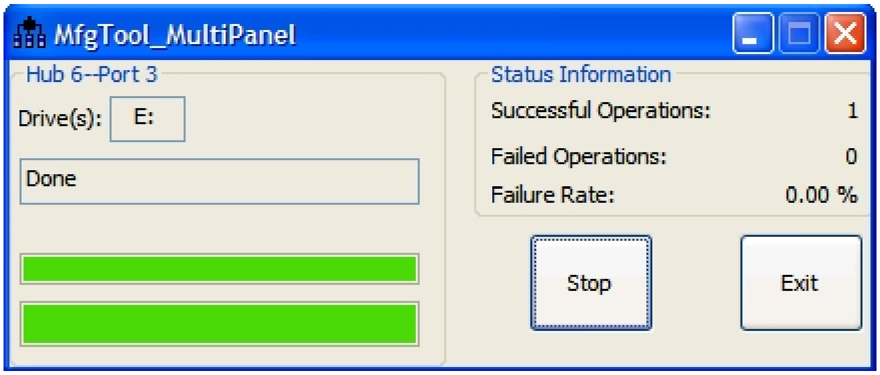
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 9. Download status

The figure below shows the tool once the flash is complete.



Hub 6--Port 3

Drive(s): E:

Done

Status Information

Successful Operations: 1

Failed Operations: 0

Failure Rate: 0.00 %

Figure 10. Download complete

8. Click **Stop**.
9. Change the board boot switch to (S3, S2,S1) 0010, 0010,0100100000 (from 1 bit) to boot from SD on CPU Board .

   Change the board boot switch to (S3, S2,S1) 0010, 0001,0001000000 (from 1 bit) to boot from NAND.

4.4 Booting

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

There are two hardware displays supported in this release: one LVDS display panel and one HDMI output.

- To enable the LVDS0 display, see Section Booting with single display: LVDS display.
- To enable the HDMI display, see Section Booting with single display: HDMI display.
- To enable LVDS0 and HDMI output dual display feature, see Section Booting with dual displays: LVDS and HDMI displays.

NOTE

There are two LVDS ports in SabreAuto hardware, LVDS0 and LVDS1. LVDS0 is taken as the primary display in this release. The LVDS0 port is in the CPU board. The LVDS1 is in the base board.

4.4.1 Booting with single display: LVDS display

To boot from SD, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc1
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 galcore.contiguousSize=20132659
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc1
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 galcore.contiguousSize=201326592
androidboot.selinux=permissive androidboot.dm_verity=disable
U-Boot > saveenv
```

To boot from NAND, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd 'nand read 0x12000000 0x4000000 0x1000000;boota 0x12000000'
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 galcore.contiguousSize=201326592
mtdparts=gpmi-nand:64m(bootloader),16m(bootimg),16m(recovery),1m(misc),-(root) ubi.mtd=5
buildvariant=use
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux feature, which detect the system's information. To avoid this, "androidboot.selinux=permissive" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd 'nand read 0x12000000 0x4000000 0x1000000;boota 0x12000000'
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 galcore.contiguousSize=201326592
mtdparts=gpmi-nand:64m(bootloader),16m(bootimg),16m(recovery),1m(misc),-(root) ubi.mtd=5
buildvariant=user androidboot.selinux=permissive
U-Boot > saveenv
```

4.4.2 Booting with single display: HDMI display

To boot from SD, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc1
U-Boot > setenv bootargs console=ttymxc3,115200 androidboot.console=ttymxc3 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb1:off
video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale cma=512M
galcore.contiguousSize=201326592
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc1
U-Boot > setenv bootargs console=ttymxc3,115200 androidboot.console=ttymxc3 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb1:off
```

```
video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale cma=512M
galcore.contiguousSize=201326592 androidboot.selinux=permissive
androidboot.dm_verity=disabled
U-Boot > saveenv
```

To boot from NAND, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd 'nand read 0x12000000 0x4000000 0x1000000;boota 0x12000000'
U-Boot > setenv bootargs console=ttymxc3,115200 init=/init video=mxcfb0:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off vmalloc=128M
androidboot.console=ttymxc3 consoleblank=0 androidboot.hardware=freescale cma=512M
galcore.contiguousSize=201326592 mtdparts=gpmi-nand:64m(bootloader),16m(bootimg),
16m(recovery),1m(misc),-(root) ubi.mtd=5 buildvariant=user
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux feature, which detect the system's information. To avoid this, "androidboot.selinux=permissive" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd 'nand read 0x12000000 0x4000000 0x1000000;boota 0x12000000'
U-Boot > setenv bootargs console=ttymxc3,115200 init=/init video=mxcfb0:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb1:off video=mxcfb2:off video=mxcfb3:off vmalloc=128M
androidboot.console=ttymxc3 consoleblank=0 androidboot.hardware=freescale cma=512M
galcore.contiguousSize=201326592 mtdparts=gpmi-nand:64m(bootloader),16m(bootimg),
16m(recovery),1m(misc),-(root) ubi.mtd=5 buildvariant=user androidboot.selinux=permissive
U-Boot > saveenv
```

4.4.3 Booting with dual displays: LVDS and HDMI displays

To boot from SD, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc1
U-Boot > setenv bootargs console=ttymxc3,115200 androidboot.console=ttymxc3 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=ldb,bpp=32 video=mxcfb1:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale
cma=512M galcore.contiguousSize=201326592
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc1
U-Boot > setenv bootargs console=ttymxc3,115200 androidboot.console=ttymxc3 consoleblank=0
vmalloc=128M init=/init video=mxcfb0:dev=ldb,bpp=32 video=mxcfb1:dev=hdmi,
1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off androidboot.hardware=freescale
cma=512M galcore.contiguousSize=201326592 androidboot.selinux=permissive
androidboot.dm_verity=disabled
U-Boot > saveenv
```

To boot from NAND, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd 'nand read 0x12000000 0x4000000 0x1000000;boota 0x12000000'
U-Boot > setenv bootargs console=ttymxc3,115200 init=/init video=mxcfb0:dev=ldb,bpp=32
video=mxcfb1:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off vmalloc=128M
androidboot.console=ttymxc3 consoleblank=0 androidboot.hardware=freescale cma=512M
galcore.contiguousSize=201326592 mtdparts=gpmi-nand:64m(bootloader),16m(bootimg),
16m(recovery),1m(misc),-(root) ubi.mtd=5 buildvariant=user
U-Boot > saveenv
```

With settings above, the Android platform does not start the shell console. It enables the default Android selinux feature, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" needs to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd 'nand read 0x12000000 0x4000000 0x1000000;boota 0x12000000'
U-Boot > setenv bootargs console=ttymxc3,115200 init=/init video=mxcfb0:dev=ldb,bpp=32
video=mxcfb1:dev=hdmi,1920x1080M@60,bpp=32 video=mxcfb2:off video=mxcfb3:off vmalloc=128M
androidboot.console=ttymxc3 consoleblank=0 androidboot.hardware=freescale cma=512M
galcore.contiguousSize=201326592 mtdparts=gpmi-nand:64m(bootloader),16m(bootimg),
16m(recovery),1m(misc),-(root) ubi.mtd=5 buildvariant=user androidboot.selinux=permissive
U-Boot > saveenv
```

4.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

Figure 11. Android Nougat image

5 Working with the i.MX 6 SoloLite EVK Board

5.1 Board hardware

The figure below shows the different components of the SoloLite EVK board.

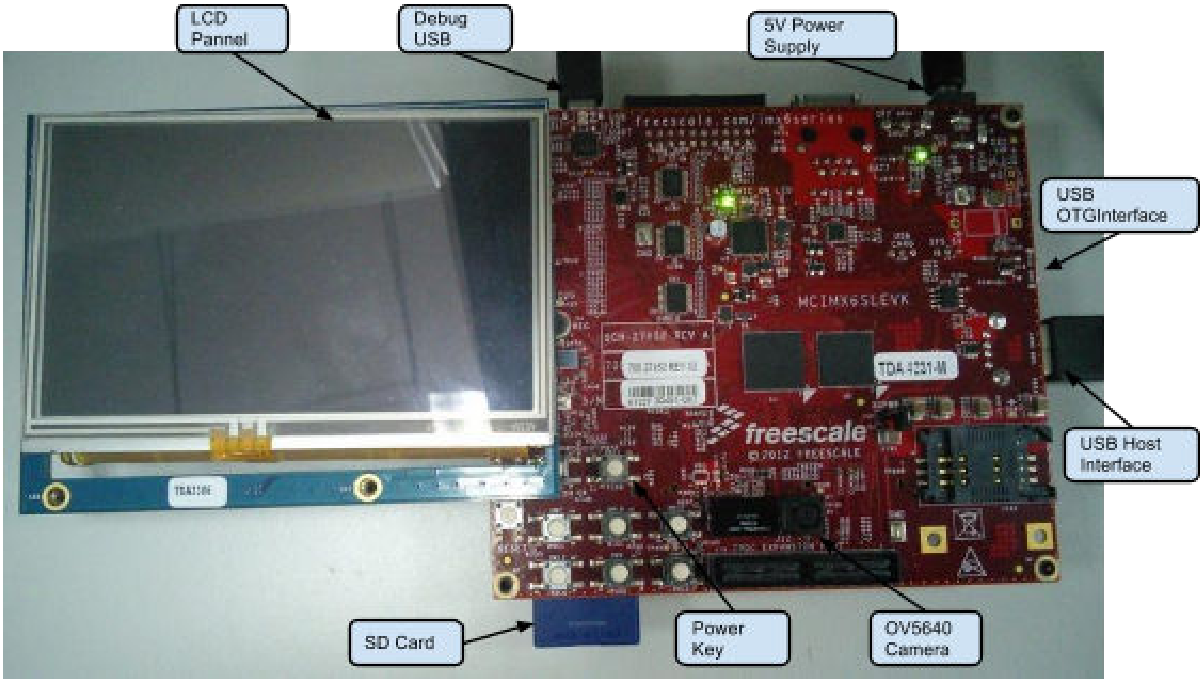


Figure 12. i.MX 6 SoloLite EVK board

5.2 Board images

The table below describes the location in the board partitions of the software images in android_N7.1.1_1.0.0_image_6slevk.tar.gz on board partitions.

Table 4. Board images

Image name	Path in release package	Download target
u-boot-imx6sl.imx	\	SD first 8 MB block
boot-imx6sl.img	\SD	SD 1st partition
recovery-imx6sl.img	\SD	SD 2nd partition
system.img	\SD	SD 5th partition

5.3 Flashing board images

The board images can be flashed to the target board by using the MFGTool. The release package includes MFGTool for i.MX 6Dual/6Quad, i.MX 6Solo/6DualLite, i.MX 6SoloX, i.MX 6SoloLite, i.MX 6QuadPlus, and i.MX 7Dual in android_N7.1.1_1.0.0_tools.tar.gz. The MFGTool is mfgtools.tar.gz.

NOTE

The MFGTool only works in Windows OS environment.

Perform the following steps to flash the board images:

1. Unzip the mfgtools.tar.gz file to a selected location. The directory is named MFGTool-Dir in this example.
2. If the directory does not exist, create the "android/evk" directory under the MFGTool-Dir/Profiles/Linux/OS Firmware/ files path.

3. Copy following files from release_package/android_N7.1.1_1.0.0_image_6slevk.tar.gz to your MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/evk directory.
    - u-boot-imx6sl.imx
    - SD/boot-imx6sl.img
    - SD/system.img
    - SD/recovery-imx6sl.img

NOTE

Do not replace other files in files directory and OS firmware directory.

4. Change the i.MX 6SoloLite-EVK board's S1 (boot mode) to 10 (from 1 bit to 2 bit) to enter flash mode.
5. Power on the board. Use USB cable on the i.MX 6SoloLite-EVK board OTG port, and connect a computer running Windows OS with the i.MX 6SoloLite-EVK board.

NOTE

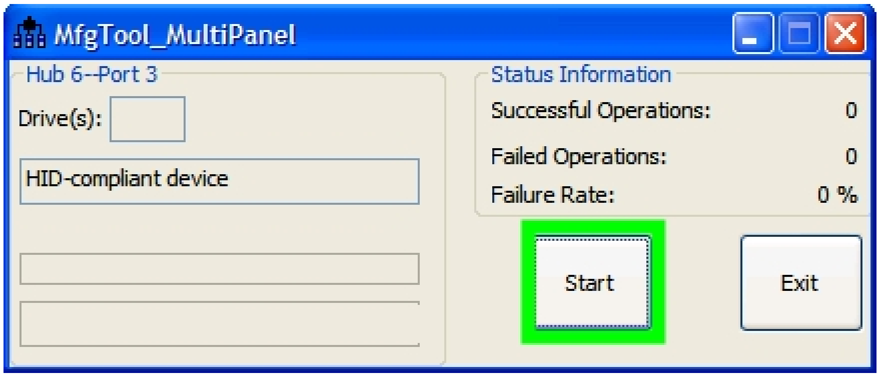
There are two USB micro ports in i.MX 6SoloLite-EVK board: USB to UART, USB OTG. The USB to UART can be referenced as debug UART, and the USB OTG can be referenced as USB in the hardware image above. The debug UART can be used to watch the log of the hardware boot processing.

6. Double-click the *.vbs file according to the target device as shown in the following table.

Table 5. MFGTool VBS file

Target device and boot storage	VBS file
i.MX 6SoloLite EVK SD	mfgtool2-android-mx6sl-evk-sd.vbs

7. Click Start to start flashing images.



Hub 6--Port 3

Drive(s):

HID-compliant device

Status Information

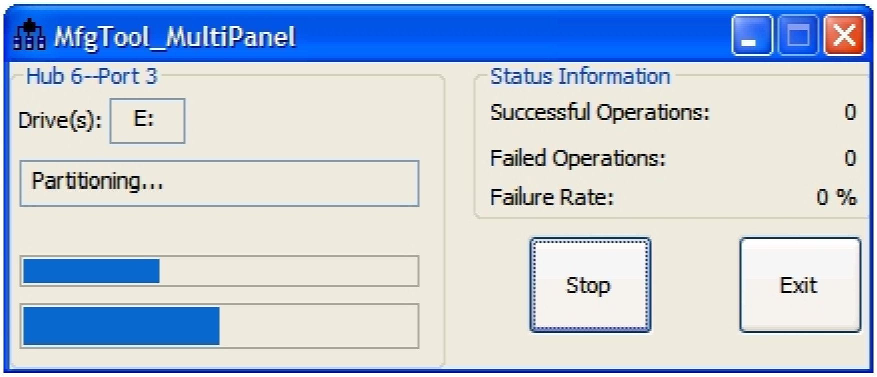
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 13. Starting flash

The figure below shows flashing in progress, and the status bar shows the flash status. The flash may take one to two minutes depending on the host machine.



Drive(s): E:

Partitioning...

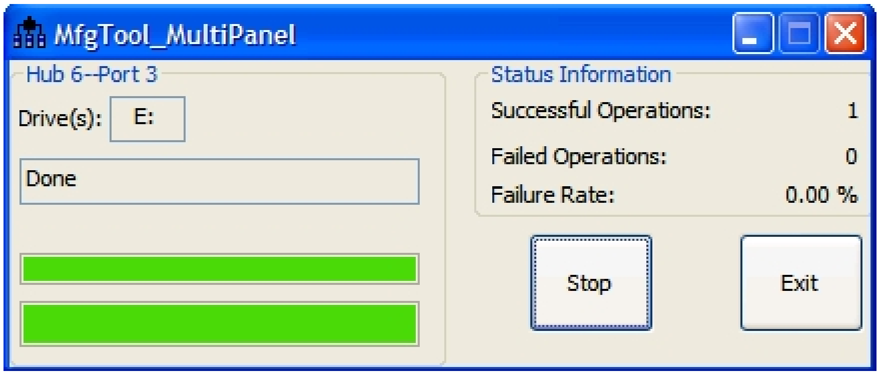
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 14. Download status

The figure below shows the tool when the flash is complete.



Drive(s): E:

Done

Successful Operations: 1

Failed Operations: 0

Failure Rate: 0.00 %

Figure 15. Download complete

8. Click Stop and disconnect the USB cable.
9. Change S1 (boot mode) to 01 (from 1 bit to 2 bit). Change "Boot Switch(SW3,4,5)" to 01000000 (from 1 bit to 8 bit) 00101100 (from 1 bit to 8 bit) 00000000 (from 1 bit to 8 bit) to switch the board back to SD1 boot mode.

5.4 Booting with single display: LCD display

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc1
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc1
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale androidboot.selinux=permissive
androidboot.dm_verity=disabled
U-Boot > saveenv
```

5.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

Figure 16. Android Nougat image

6 Working with the i.MX 6SoloX SABRE-SD Board

6.1 Board hardware

The figure below shows the different components of the i.MX 6SoloX SABRE-SD board.

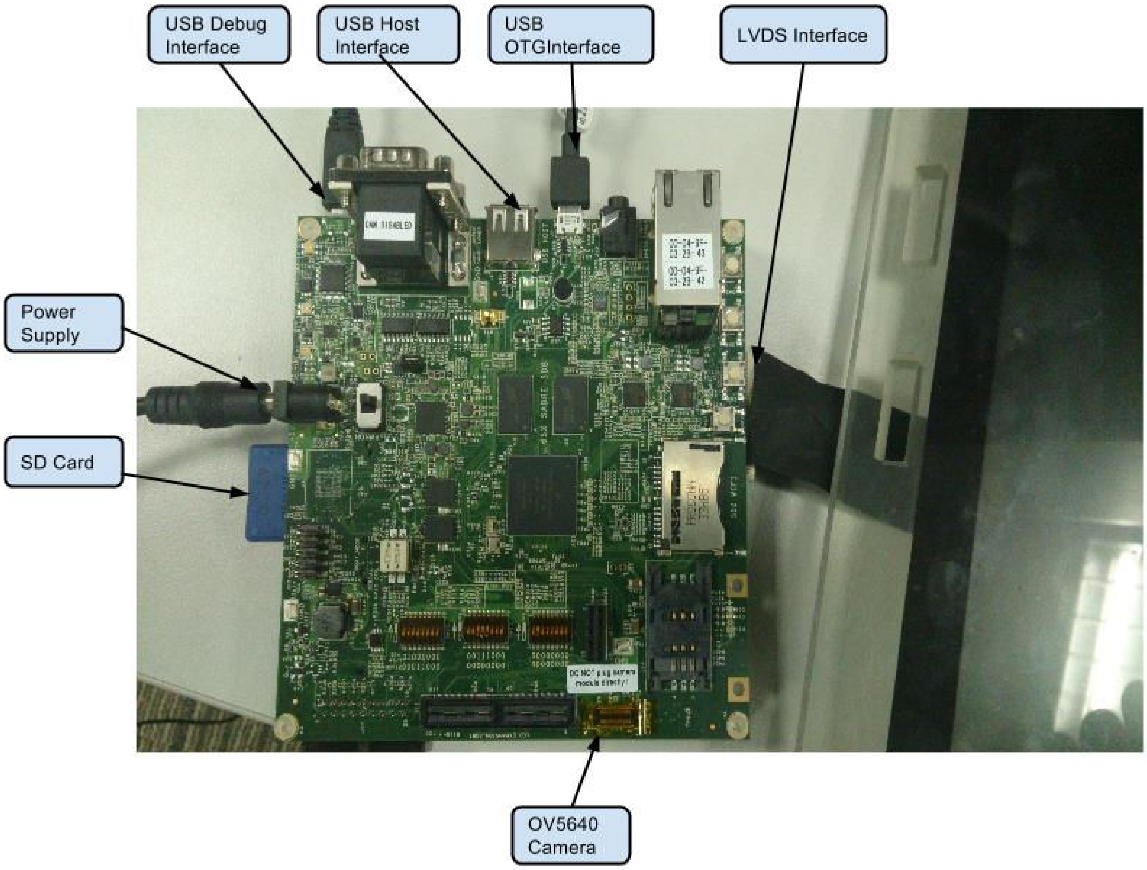


Figure 17. i.MX 6SoloX SABRE-SD board

6.2 Board images

The table below describes the location in the board partitions of the software images in android_N7.1.1_1.0.0_image_6sxsabresd.tar.gz on board partitions.

Table 6. Board images

Image name	Path in release package	Download target
u-boot-imx6sx.imx	\	SD first 8 MB block
boot-imx6sx.img	\SD	SD 1st partition
recovery-imx6sx.img	\SD	SD 2nd partition
system.img	\SD	SD 5th partition

6.3 Flashing board images

The board images can be flashed to the target board by using the MFGTool. The release package includes MFGTool for i.MX 6Dual/6Quad, i.MX 6Solo/6DualLite, i.MX 6SoloX, i.MX 6SoloLite, i.MX 6QuadPlus, and i.MX 7Dual in android_N7.1.1_1.0.0_tools.tar.gz. The MFGTool is mfgtools.tar.gz.

NOTE

The MFGTool only works in Windows OS environment.

Perform the following steps to flash the board images:

1. Unzip the mfgtools.tar.gz file to a selected location. The directory is named MFGTool-Dir in this example.
2. If the directory does not exist, create the "android/sabresd" directory under the MFGTool-Dir/Profiles/Linux/OS Firmware/files path.
3. Copy following files from release_pakcage/android_N7.1.1_1.0.0_image_6sxsabresd.tar.gz to your MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/sabresd path.
    - u-boot-imx6sx.imx
    - SD/boot-imx6sx.img
    - SD/system.img
    - SD/recovery-imx6sx.img

NOTE

Do not replace other files in files directory and OS firmware directory.

4. Change the the i.MX SoloX SABRE-SD board's S1 (boot mode) to 10 (from 1-2 bit) to enter flash mode.
5. Power on the board. Use the USB cable on the OTG port of the i.MX SoloX SABRE-SD board, and connect a computer running Windows OS with the i.MX SoloX SABRE-SD board.

NOTE

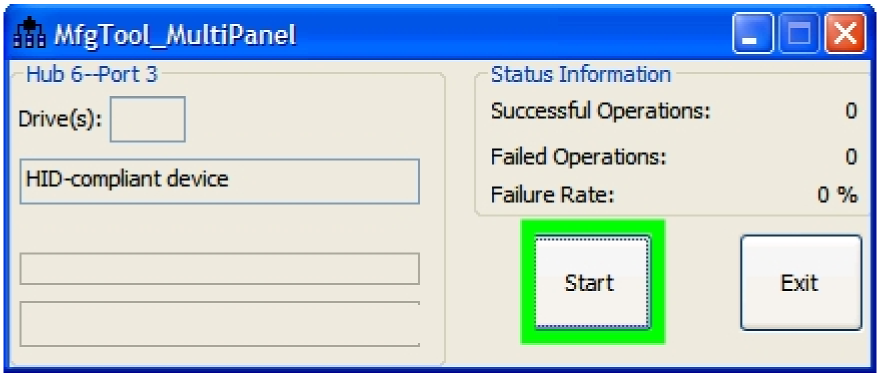
There are two USB micro ports on the i.MX SoloX SABRE-SD board: USB to UART, USB OTG. The USB to UART can be referenced as debug UART, and the USB OTG can be referenced as USB in the hardware image above. The debug UART can be used to watch the log of the hardware boot processing.

6. Double-click the *.vbs file according to the target device as shown in the following table.

Table 7. MFGTool VBS file

Target device and boot storage	VBS file
i.MX SoloX SABRE-SD SD	mfgtool2-android-mx6sx-sabresd-sdcard.vbs

7. Click Start to start flashing images.



Drive(s):

HID-compliant device

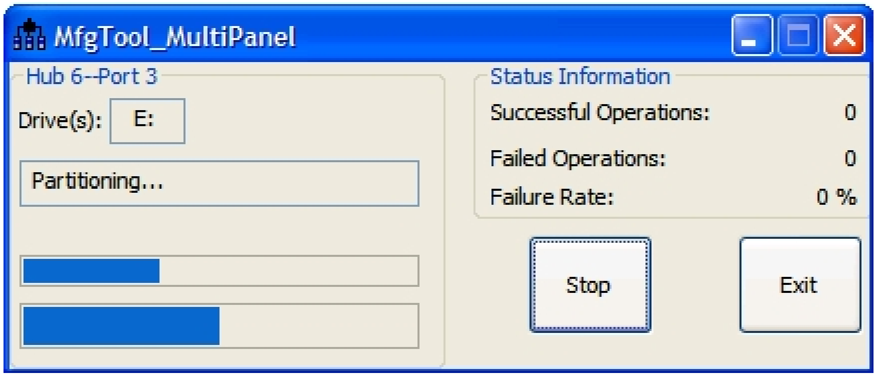
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 18. Starting flash

The figure below shows flashing in progress, and the status bar shows the flash status. The flash may take one to two minutes depending on the host machine.



Drive(s): E:

Partitioning...

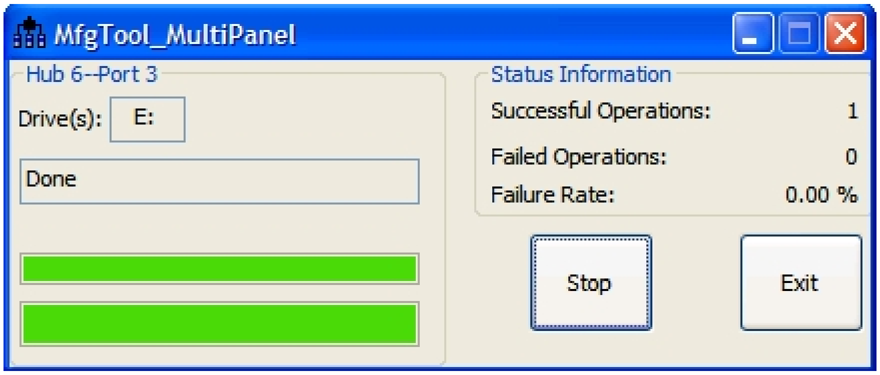
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 19. Download status

The figure below shows the tool when the flash is complete.



Drive(s): E:

Done

Successful Operations: 1

Failed Operations: 0

Failure Rate: 0.00 %

Figure 20. Download complete

8. Click Stop and disconnect the USB cable.
9. Change S1 (boot mode) to 01 (from 1 bit to 2 bit). Change “Boot Switch (SW10,11,12)” to 00000000 (from 1 bit to 8 bit), 00111000 (from 1 bit to 8 bit), 01000000 (from 1 bit to 8 bit) to switch the board back to SD4 boot mode.

6.4 Booting with single display: LVDS display

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
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
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
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
androidboot.selinux=permissive androidboot.dm_verity=disabled
U-Boot > saveenv
```

6.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

Figure 21. Android Nougat image

7 Working with the i.MX 6SoloX SABRE-AI Board

7.1 Board hardware

The figure below shows the different components of the i.MX 6SoloX SABRE-AI board.

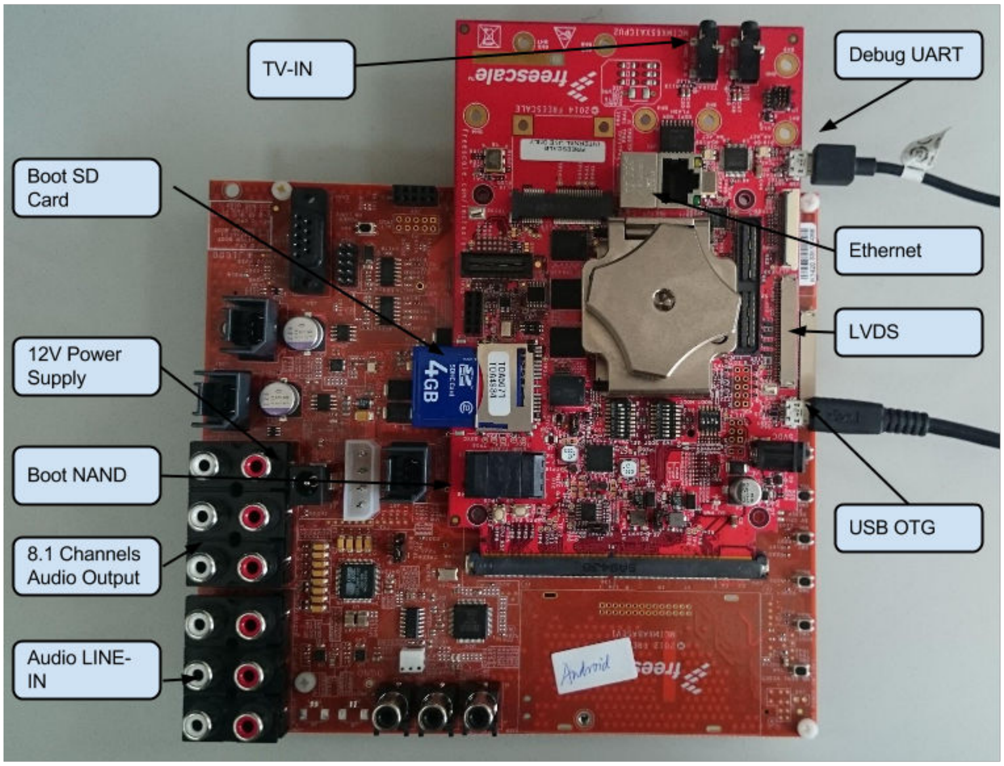


Figure 22. i.MX 6SoloX SABRE-AI board

7.2 Board images

The table below describes the location in the board partitions of the SD card and NAND images in android_N7.1.1_1.0.0_image_6sxqsabreauto.tar.gz.

Table 8. Board images

Image name	Path in release package	Download target
u-boot-imx6sx.imx	\	SD first 8 MB block
boot-imx6sx.img	\SD	SD 1st partition
recovery-imx6sx.img	\SD	SD 2nd partition
system.img	\SD	SD 5th partition
u-boot-imx6sx-nand.i mx	\	NAND 1st 64 MB MTD partition
boot-imx6sx.img	\NAND	NAND 2nd 16 MB MTD partition
recovery-imx6sx.img	\NAND	NAND 3rd 16 MB MTD partition
android_root.img	\NAND	UBIFS volume for 4th MTD partition

7.3 Flashing board images

The board images can be flashed to the target board by using the MFGTool. The release package includes MFGTool for i.MX 6Dual/6Quad, i.MX 6Solo/6DualLite, i.MX 6SoloX, i.MX 6SoloLite, i.MX 6QuadPlus, and i.MX 7Dual in android_N7.1.1_1.0.0_tools.tar.gz. The MFGTool is mfgtools.tar.gz.

NOTE

The MFGTool only works in the Windows OS environment.

Perform the following steps to flash the board images:

1. Unzip the mfgtools.tar.gz file to a selected location. The directory is named MFGTool-Dir in this example.
2. If the directory does not exist, create the "android/sabreauto" directory under the MFGTool-Dir/Profiles/Linux/OS Firmware/files path.
3. Copy following files from either release_package/android_N7.1.1_1.0.0_image_6sxsabreseauto.tar.gz to the MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/sabreauto path.
    - u-boot-imx6sx.imx
    - SD/boot-imx6sx.img
    - SD/system.img
    - SD/recovery-imx6sx.img

    NOTE

    - Do not replace other files in files directory and OS firmware directory.
    - When the NAND boot is used, the files are in the “nand” folder, and you can use android_root.img instead system.img. The bootloader should be u-boot-imx6sx-nand.imx. Change S3,S4 to 00000000,00000001.
    - Change the SABRE-AI S1 (boot mode) to 0101 (from 1 bit to 4 bit) to enter flash mode. Change SW3,SW4 (boot configuration) to 00001100, 01000010 to boot from SD.

4. Power on the board. Use the USB cable on the OTG port of the i.MX SoloX SABRE-AI board, and connect a computer running Windows OS with the i.MX SoloX SABRE-AI board.

    NOTE

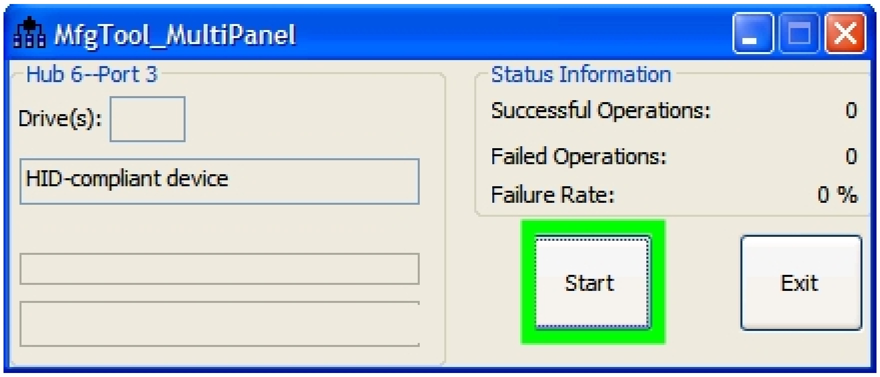
    The USB micro port on the SABRE-AI board is J10.

5. Double-click the *.vbs file according to the target device as shown in the following table.

Table 9. MFGTool VBS file

Target device and boot storage	VBS file
i.MX SoloX SABRE-AI SD	mfgtool2-android-mx6sx-sabreauto-sdcard.vbs
i.MX 6SoloX SABRE-AI NAND	mfgtool2-android-mx6sx-sabreauto-nand.vbs

6. Click Start to start flashing images.

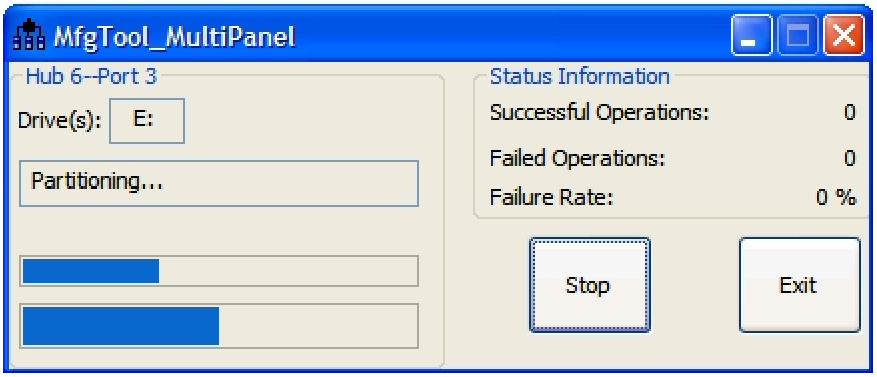


Hub 6--Port 3
Drive(s):
HID-compliant device

Status Information
Successful Operations: 0
Failed Operations: 0
Failure Rate: 0 %

Figure 23. Starting flash

The figure below shows flashing in progress, and the status bar shows the flash status. The flash may take one to two minutes depending on the host machine.

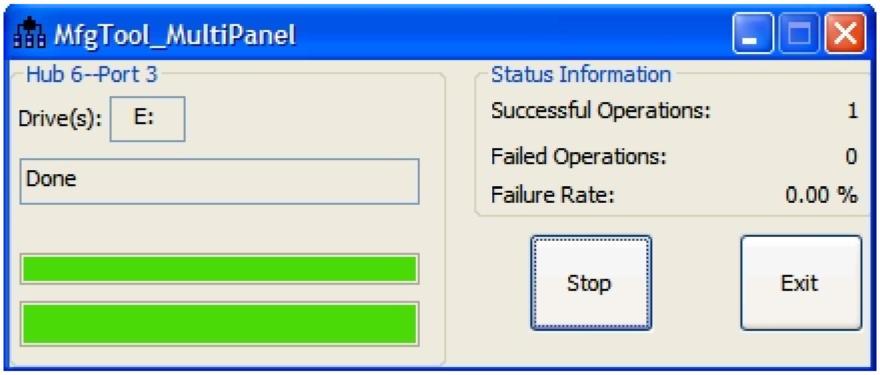


Hub 6--Port 3
Drive(s): E:
Partitioning...

Status Information
Successful Operations: 0
Failed Operations: 0
Failure Rate: 0 %

Figure 24. Download status

The figure below shows the tool when the flash is complete.



Hub 6--Port 3
Drive(s): E:
Done

Status Information
Successful Operations: 1
Failed Operations: 0
Failure Rate: 0.00 %

Figure 25. Download complete

7. Click Stop and disconnect the USB cable.
8. Change “Boot Switch(S1)” to 0010 (1 bit to 4 bit) to switch the board back to SD boot mode. Change the board boot switch to (S1, SW3, SW4) 0010, 00000000,00000001 (from 1 bit) to boot from NAND.

7.4 Booting with single display: LVDS display

When the Android system image is programmed and the Boot Switch(es) is configured, the system is ready to be powered on.

There are two LVDS ports on the SABRE-AI board, LVDS0, and LVDS1. LVDS0 is taken as the primarly display in this release. The LVDS0 port is on the CPU board. The LVDS1 is on the base board.

To boot from SD, set the U-Boot environment variables as follows

```
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=512M
galcore.contiguousSize=201326592
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
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=512M
galcore.contiguousSize=201326592 androidboot.selinux=permissive
androidboot.dm_verity=disabled
U-Boot > saveenv
```

To boot from NAND, set the U-Boot environment variables as follows:

```
U-Boot > setenv bootcmd 'nand read 0x80800000 0x4000000 0x1000000;boota 0x80800000'
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 cma=512M
galcore.contiguousSize=201326592
androidboot.console=ttymxc0 consoleblank=0 androidboot.hardware=freescale mtdparts=gpmi-nand:
64m(bootloader),16m(bootimg),16m(recovery),-(root) ubi.mtd=5
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux feature, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd 'nand read 0x80800000 0x4000000 0x1000000;boota 0x80800000'
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 cma=512M
galcore.contiguousSize=201326592
androidboot.console=ttymxc0 consoleblank=0 androidboot.hardware=freescale mtdparts=gpmi-nand:
64m(bootloader),16m(bootimg),16m(recovery),-(root) ubi.mtd=5 androidboot.selinux=permissive
U-Boot > saveenv
```

7.5 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

Figure 26. Android Nougat image

8 Working with the i.MX 7Dual SABRE-SD Board

8.1 Board hardware

The figure below shows the different components of the i.MX 7Dual SABRE-SD board.

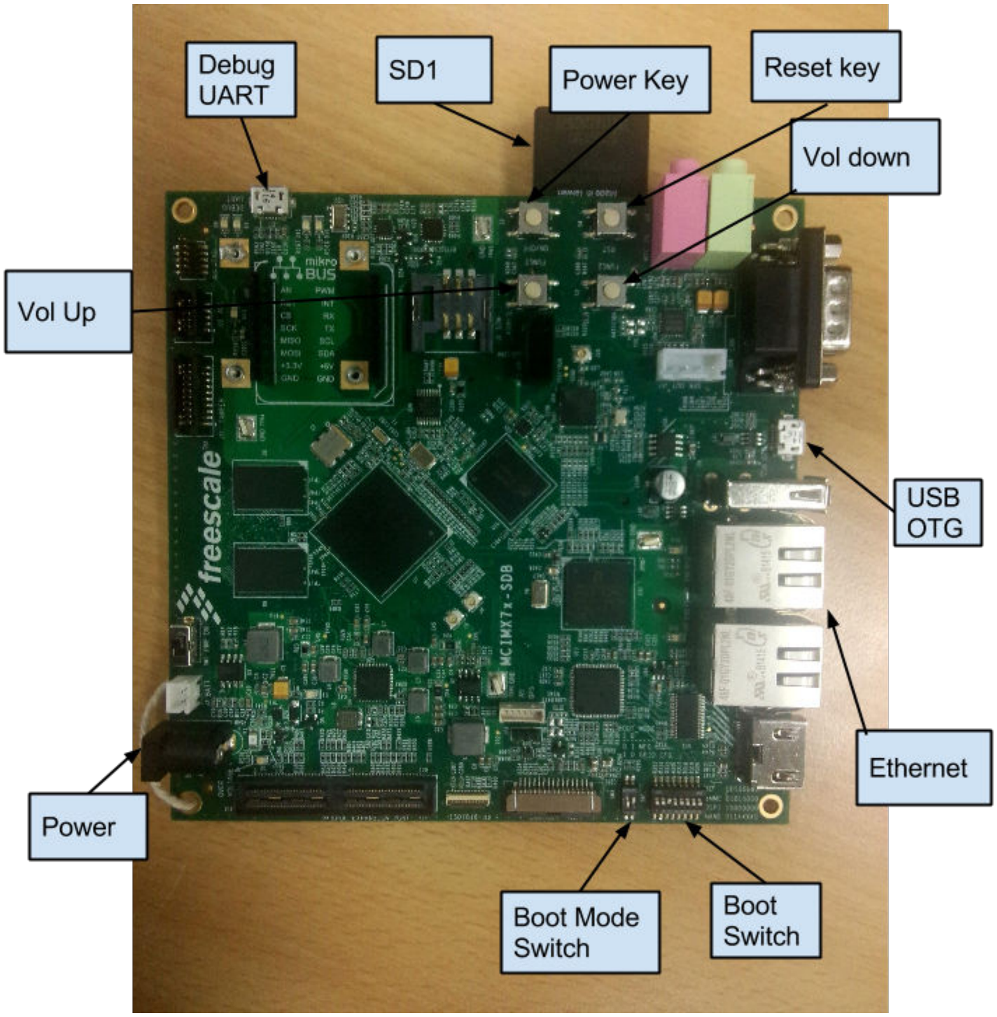


Figure 27. i.MX 7Dual SABRE-SD board

8.2 Board images

The table below describes the location in the board partitions of the software images in android_N7.1.1_1.0.0_image_7dsabresd.tar.gz.

Table 10. Board images

Image name	Path in release package	Download target
u-boot-imx7d.imx u-boot-imx7d-epdc.imx	\	SD first 8 MB block
boot-imx7d.img boot-imx7d-epdc.img	\SD	SD 1st partition
recovery-imx7d.img recovery-imx7d-epdc.img	\SD	SD 2nd partition
system.img	\SD	SD 5th partition

8.3 Flashing board images

The board images can be flashed to the target board by using the MFGTool. The release package includes MFGTool for i.MX 6Dual/6Quad, i.MX 6Solo/6DualLite, i.MX 6SoloX, i.MX 6SoloLite, i.MX 6QuadPlus, and i.MX 7Dual in android_N7.1.1_1.0.0_tools.tar.gz. The MFGTool is mfgtools.tar.gz.

NOTE

The MFGTool only works in Windows OS environment.

Perform the following steps to flash the board images:

1. Unzip the mfgtools.tar.gz file to a selected location. The directory is named MFGTool-Dir in this example.
2. If the directory does not exist, create the "android/sabresd" directory under the MFGTool-Dir/Profiles/Linux/OS Firmware/files path.
3. Copy following files from release_package/android_N7.1.1_1.0.0_image_7dsabresd.tar.gz to your MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/sabresd directory.
    - u-boot-imx7d.imx
    - SD/boot-imx7d-epdc.img
    - SD/boot-imx7d.img
    - SD/system.img
    - SD/recovery-imx7d.img
    - SD/recovery-imx7d-epdc.img

NOTE

Do not replace other files in files directory and OS firmware directory.

4. Change the board's SW3 (boot mode) to 01 (from 1 bit to 2 bit) to enter flash mode.
5. Power on the board. Use USB cable on the board OTG port, and connect a computer running Windows OS with the board.

NOTE

There are two USB micro ports in i.MX 7Dual SABRE-SD board: USB to UART, USB OTG. The USB to UART can be referenced as debug UART, and the USB OTG can be referenced as USB in the hardware image above. The debug UART can be used to watch the log of the hardware boot processing.

The SD card should be plugged in after the board is powered on.

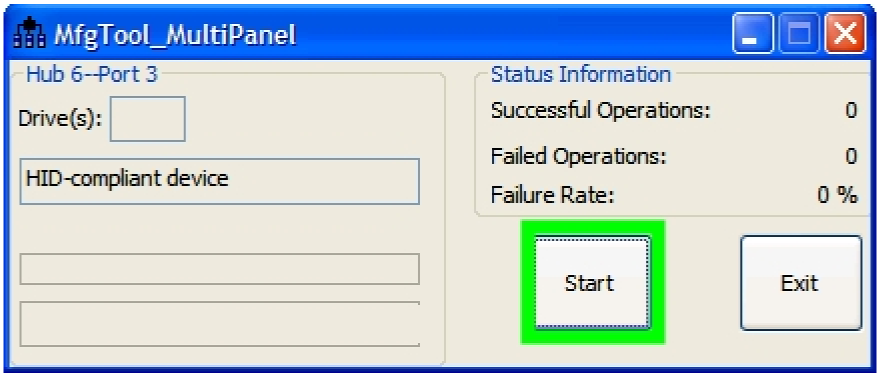
To enable the E-Ink display, copy the SD/boot-imx7d-epdc.img and SD/recovery-imx7d-epdc.img files to MFGTool-Dir/Profiles/Linux/OS Firmware/files/android/sabresd/. Rename them as boot-imx7d.img and recovery-imx7d.img.

6. Double-click the *.vbs file according to the target device as shown in the following table.

Table 11. MFGTool VBS file

Target device and boot storage	VBS file
i.MX 7Dual SABRE-SD	mfgtool2-android-mx7d-sabresd-sd. vbs

7. Click Start to start flashing images.



Hub 6--Port 3

Drive(s):

HID-compliant device

Status Information

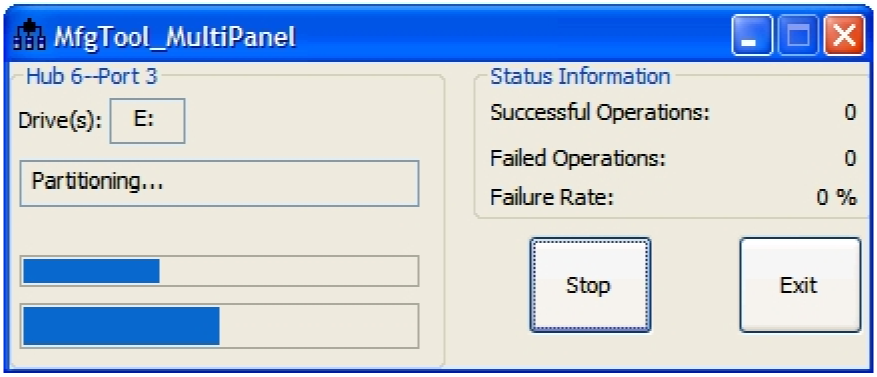
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 28. Starting flash

The figure below shows flashing in progress, and the status bar shows the flash status. The flash may take one to two minutes depending on the host machine.



Hub 6--Port 3

Drive(s): E:

Partitioning...

Status Information

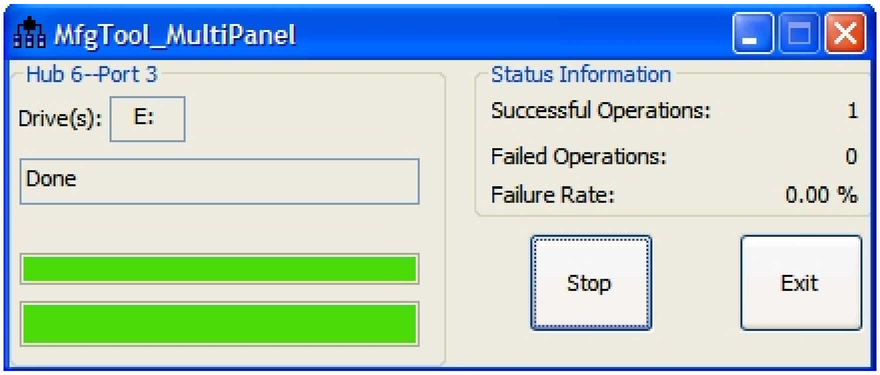
Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Figure 29. Download status

The figure below shows the tool when the flash is complete.



Hub 6--Port 3

Drive(s): E:

Done

Status Information

Successful Operations: 1

Failed Operations: 0

Failure Rate: 0.00 %

Figure 30. Download complete

8. Click Stop and disconnect the USB cable.
9. Change SW3 (boot mode) to 10 (from 1 bit to 2 bit). Change SW2 to switch the board back to SD4 boot mode 00100000 (SD), 01010000 (eMMC), 011XXXX0 (NAND).

8.4 Booting with single display: LVDS display or HDMI display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

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

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system's information. To avoid this, "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" need to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc0
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale androidboot.selinux=permissive
androidboot.dm_verity=disabled
U-Boot > saveenv
```

8.5 Booting with EInk Display

In the U-Boot prompt, set the U-Boot environment variables as shown below:

```
U-Boot > setenv bootcmd boota mmc0
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale max17135:pass=2,vcom=-2370000
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. It enables the default Android selinux and dm_verity security features, which restrict users to change the system and detect the system information. To avoid this, append "androidboot.selinux=permissive" and "androidboot.dm_verity=disabled" to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv bootcmd boota mmc0
U-Boot > setenv bootargs console=ttymxc0,115200 init=/init androidboot.console=ttymxc0
consoleblank=0 androidboot.hardware=freescale max17135:pass=2,vcom=-2370000
androidboot.selinux=permissive androidboot.dm_verity=disabled
U-Boot > saveenv
```

8.6 Board reboot

After you have completed download and setup, reboot the board and wait for the Android platform to boot up.

Figure 31. Android Nougat image

9 Revision History

Table 12. Revision history

Revision number	Date	Substantive changes
N7.1.1_1.0.0	03/2017	Initial release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. ìTypicalî parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including ìtypicals,î must be validated for each customer application by customerís technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Document Number: AQSUG
Rev. N7.1.1_1.0.0
03/2017