AQSUG

Android Quick Start Guide

Rev. android-12.1.0_1.0.0 —   **User guide**
24 October 2022

Document information

Information	Content
Keywords	Android, i.MX, android-12.1.0_1.0.0
Abstract	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.

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 8M Mini
- i.MX 8M Nano
- i.MX 8M Plus
- i.MX 8M Quad
- i.MX 8ULP
- i.MX 8QuadMax (Silicon Revision B0)
- i.MX 8QuadXPlus (Silicon Revision B0 and Silicon Revision C0)

Supported boards:

- EVK board and Platform
- MEK board and Platform

3   Working with the i.MX 8M Mini EVK Board

3.1   Board hardware

The figure below shows the different components of the i.MX 8M Mini EVK LPDDR4 board.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

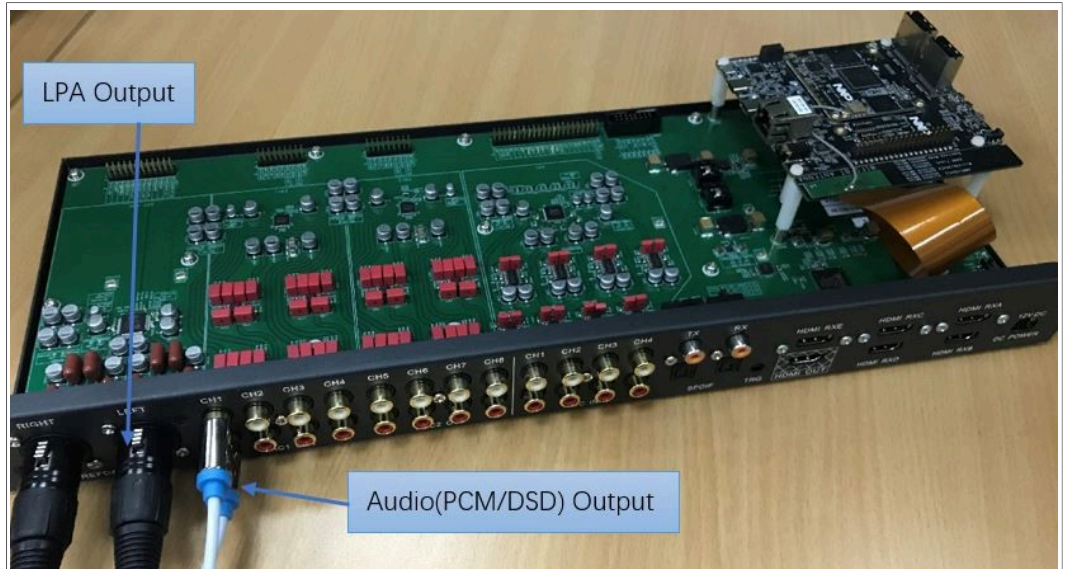**2 / 65**

Figure 1. i.MX 8M Mini EVK LPDDR4 board

Figure 2. i.MX 8M Mini EVK with audio board

Figure 3. i.MX Mini SAS cable with DSI-to-HDMI adapter

Figure 4. MX8-DSI-OLED1 MIPI panel

AQSUG

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

4 / 65

Figure 5. OV5640 CSI MIPI camera

Note:

- *i.MX 8M Mini EVK LPDDR4 Rev. C board and i.MX 8M Mini EVK DDR4 Rev. C board are supported in this release.*
- *To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "MIPI DSI" port.*
- *To test the MIPI panel display, connect the i.MX MIPI panel to the "MIPI DSI" port.*
- *To test the camera, connect the OV5640 CSI MIPI camera to the "MIPI CSI" port.*
- *For i.MX 8M Mini EVK LPDDR4 board, BT and Wi-Fi functions are supported.*
- *For i.MX 8M Mini EVK DDR4 board, BT and Wi-Fi functions are not supported.*
- *"i.MX 8M Mini EVK REV C" indicates the revision of the base board.*

3.2 Board images

The table below describes the location in the board partitions of the software images in `android-12.1.0_1.0.0_image_8mmevk.tar.gz`.

Table 1. Board images

Image name	Download target
`spl-imx8mm-dual.bin`	33 KB offset of MMC for i.MX 8M Mini EVK LPDDR4 board.
`spl-imx8mm-trusty-dual.bin`	33 KB offset of MMC for i.MX 8M Mini EVK LPDDR4 board.
`spl-imx8mm-trusty-secure-unlock-dual.bin`	33K offset of mmc for i.MX 8M Mini EVK LPDDR4 board.
`bootloader-imx8mm-dual.img`	`bootloader_a` and `bootloader_b` partitions on the i.MX 8M Mini EVK LPDDR4 board.
`bootloader-imx8mm-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions on the i.MX 8M Mini EVK LPDDR4 board.
`bootloader-imx8mm-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions on i.MX 8M Mini EVK LPDDR4 board.
`u-boot-imx8mm.imx`	33 KB offset of MMC for a board with LPDDR4 on it.
`u-boot-imx8mm-evk-uuu.imx`	The bootloader used by UUU for i.MX 8M Mini board with LPDDR4 on it. It is not flashed to MMC.
`u-boot-imx8mm-ddr4.imx`	33 KB offset of SD card for a board with DDR4 on it.
`u-boot-imx8mm-ddr4-evk-uuu.imx`	Bootloader used by UUU for i.MX 8M Mini board with DDR4 on it. It is not flashed to the SD card.
`imx8mm_mcu_demo.img`	5120 KB offset of MMC.
`partition-table.img`	0 offset of MMC. It is used for single bootloader condition and the target storage device should be larger than 13 GB.

Table 1. Board images...*continued*

Image name	Download target
`partition-table-dual.img`	0 offset of MMC. It is used for dual bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 offset of MMC. It is used for single bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 offset of MMC. It is used for dual bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions. This is the AOSP GKI boot image.
`boot-imx.img`	`boot_a` and `boot_b` partitions. This is the boot image built with i.MX kernel tree for debugging.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions when doing VTS test with GSI system image.
`vbmeta-imx8mm.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output on i.MX 8M Mini EVK LPDDR4 board.
`vbmeta-imx8mm-m4.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output and audio playback based on Cortex-M4 FreeRTOS on i.MX 8M Mini EVK LPDDR4 board.
`vbmeta-imx8mm-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support rm67199 MIPI panel output on i.MX 8M Mini EVK LPDDR4 board.
`vbmeta-imx8mm-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions to support rm67191 MIPI panel output on i.MX 8M Mini EVK LPDDR4 board.
`vbmeta-imx8mm-ddr4.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output on i.MX 8M Mini EVK DDR4 board.
`system.img`	Logical partition `system_a` and logical partition `system_b` in super partition.
`system_ext.img`	Logical partition `system_ext_a` and logical partition `system_ext_b` in super partition.
`vendor.img`	Logical partition `vendor_a` and logical partition `vendor_b` in super partition.
`product.img`	Logical partition `product_a` and logical partition `product_b` in super partition.
`super.img`	Super partition.
`dtbo-imx8mm.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output on i.MX 8M Mini EVK LPDDR4 board.
`dtbo-imx8mm-m4.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output and audio playback based on Cortex-M4 FreeRTOS on i.MX 8M Mini EVK LPDDR4 board.
`dtbo-imx8mm-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions to support RM67199 MIPI panel output on i.MX 8M Mini EVK LPDDR4 board.
`dtbo-imx8mm-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions to support RM67191 MIPI panel output on i.MX 8M Mini EVK LPDDR4 board.
`dtbo-imx8mm-ddr4.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output playback on i.MX 8M Mini EVK DDR4 board.

Table 1. Board images...*continued*

Image name	Download target
`rpmb_key_test.bin`	Prebuilt test RPMB key, which can be used to set the RPMB key as fixed 32 bytes 0x00.
`testkey_public_rsa4096.bin`	Prebuilt AVB public key, which is extracted from the default AVB private key.

3.3 Flashing board images

The board image files can be flashed into the target board using Universal Update Utility (UUU).

For the UUU binary file, download it from GitHub: uuu release page on GitHub.

To achieve more flexibility, two script files are provided to invoke UUU to automatically flash all Android images.

- `uuu_imx_android_flash.sh` for Linux OS
- `uuu_imx_android_flash.bat` for Windows OS

For this release, these two scripts are validated on UUU 1.4.182 version. Download corresponding version from GitHub:

- For Linux OS, download the file named `uuu`.
- For Windows OS, download the file named `uuu.exe`.

Because the two script files will directly invoke UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from GitHub as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   For Rev. C boards, change the first four bits of board's sw1101 to 1010 (from 1-4 bit) to enter serial download mode.
3. Power on the board. Use the USB cable on the board USB 2.0 Type-C port to connect your PC with the board.
4. Decompress `release_package/android-12.1.0_1.0.0_ image_8mmevk.tar.gz`. The package contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Mini board, related options are described as follows.

Table 2. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8M Mini EVK, it should be `imx8mm`. This option is **mandatory**.
`-a`	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.

AQSUG
User guide
All information provided in this document is subject to legal disclaimers.
Rev. android-12.1.0_1.0.0 — 24 October 2022
© 2022 NXP B.V. All rights reserved.

7 / 65

Table 2. Options for uuu_imx_android_flash tool...*continued*

Option	Description
-b	Only flashes slot b. If this option and -a option are not used, slots a and b are both flashed.
-c card_size	Specifies which partition table image file to flash. For i.MX 8M Mini EVK, it can be followed with "28". If this option is not used, default partition-table.img or partition-table-dual.img is flashed.
-m	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
-u uboot_feature	Flashes U-Boot or SPL&bootloader image with uboot_feature in their names. For i.MX 8M Mini EVK LPDDR4, it can be dual, trusty-dual and "trusty-secure-unlock-dual". If this option is not used, the default u-boot-imx8mm.imx is flashed. For i.MX 8M Mini EVK DDR4, it should be ddr4.
-d dtb_feature	Flashes DTBO and vbmeta images with dtb_feature in their names. For i.MX 8M Mini EVK LPDDR4, it can be m4, mipi-panel and mipi-panel-rm67191. If this option is not used, default dtbo-imx8mm.img and vbmeta-imx8mm.img are flashed. For i.MX 8M Mini EVK DDR4, it should be ddr4.
-e	Erases user data after images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. For uuu_imx_android_flash.bat, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
-t target_dev	Specifies the target device. For i.MX 8M Mini EVK, it can be emmc and sd. If this option is not used, images are flashed to eMMC.
-daemon	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and execute to fastboot mode. This option is used for development.
-dryrun	Only generates a UUU script but not execute UUU with this script.
-usb usb_path	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

Obviously, -m and -d m4 should be used together.

• On Linux system, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx8mm -a -e -u
  trusty-dual
```

• On Windows system, open the command-line interface in administrator mode. The corresponding command is as follows:

```
> uuu_imx_android_flash.bat -f imx8mm -a -e -u trusty-dual
```

When the command above is executed, spl-imx8mm-trusty-dual.bin is flashed, bootloader-imx8mm-trusty-dual.img with other default images are flashed into eMMC slot a for i.MX 8M Mini EVK LPDDR4 bord..

AQSUG

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

8 / 65

Note:

- *`-u` followed with a parameter and containing `trusty` cannot be used together with `-t sd`, because Trusty OS cannot boot from SD card.*
- *To flash SD card, execute the tool with `-t sd`. To flash eMMC, it does not need to use `-t` option.*
- *If your SD card is 16 GB or uses onboard eMMC as the boot device, it does not need to use `-c` option.*
- *If your SD card is 32 GB, execute the tool with `-c 28`.*
- *For i.MX 8M Mini EVK LPDDR4 board:*
  - *To test dual bootloaders, execute the tool with `-u dual`.*
  - *To test Trusty OS and dual-bootloader both enabled condition, execute the tool with `-u trusty-dual`.*
  - *To test the demostration implementation of secure unlock, execute the tool with `-u trusty-secure-unlock-dual`. For secure unlock details, see the i.MX Android Security User's Guide (ASUG).*
  - *To test MIPI-DSI to HDMI output, it does not need to use `-d` option. `dtbo-imx8mm.img` is flashed in this condition to support Wi-Fi expansion card and this image does not support to be booted from the SD card.*
  - *To test RM67199 MIPI panel output, execute the tool with `-d mipi-panel`.*
  - *To test RM67191 MIPI panel output, execute the tool with `-d mipi-panel-rm67191`.*
  - *To test support MIPI-DSI-to-HDMI output and audio playback based on Cortex-M4 FreeRTOS, execute the tool with `-m` and `-d m4`.*
- *For i.MX 8M Mini EVK DDR4 board: execute the command with `-u ddr4 -d ddr4 -t sd`.*
- *If `uuu_imx_android_flash.bat` is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:*

```
> net use z: \\192.168.1.1\daily_images
```

"z" in the command represents an available drive letter. It can be other available drive letter.

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, the command window displays information indicating that images are already flashed.
   Note:
   If the target device has a DOS partition table on it, the flash process fails for the first time. Push the reset key on the board and execute the flash script again.
7. Power off the board.
8. Change boot device as eMMC or SD card.
   For Rev. C boards:
   - Change sw1101 to 0110110010 and change sw1102 to 0001101000 if you want to boot from SD card.
   - Change sw1101 to 0110110001 and change sw1102 to 0001010100 if you want to boot from eMMC.

3.4 Booting

After downloading the images, reboot the board using the power on/off switch.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

9 / 65

3.4.1 Booting with Single MIPI-DSI-to-HDMI or MIPI panel display

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

• i.MX 8M Mini EVK LPDDR4 Board:

```
U-Boot > setenv bootargs console=ttymxc1,115200
 earlycon=ec_imx6q,0x30890000,115200 init=/init
 cma=800M@0x400M-0xb80M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware
 loop.max_part=7 transparent_hugepage=never
 moal.mod_para=wifi_mod_para_sd8987.conf
 androidboot.lcd_density=240 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

3.4.2 Booting with Single MIPI-DSI-to-HDMI display and audio playback based on Cortex-M4 FreeRTOS

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

• i.MX 8M Mini EVK LPDDR4 Board:

```
U-Boot > setenv bootargs console=ttymxc1,115200
 earlycon=ec_imx6q,0x30890000,115200 init=/init
 cma=800M@0x400M-0xb80M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware
 loop.max_part=7 transparent_hugepage=never
 moal.mod_para=wifi_mod_para_sd8987.conf
 androidboot.lcd_density=240 bootconfig
U-Boot > setenv bootcmd "bootmcu && boota"
U-Boot > saveenv
```

Note:

To use other dtbo images, do not add `bootmcu" to "bootcmd`. The following command can recover `bootcmd`:

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

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

3.5 Board reboot

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

4 Working with the i.MX 8M Nano EVK Board

4.1 Board hardware

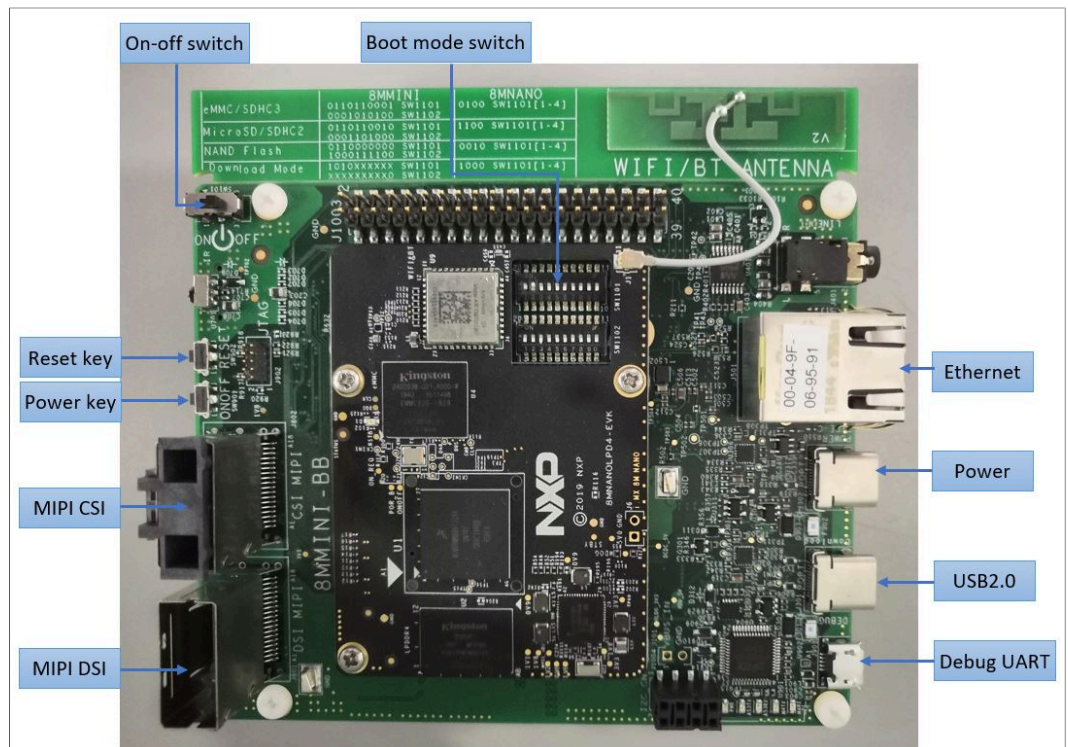The figure below shows the different components of the i.MX 8M Nano EVK board.

Figure 6. i.MX 8M Nano EVK board

Figure 7. i.MX mini SAS cable with DSI-to-HDMI adapter

AQSUG

User guide

All information provided in this document is subject to legal disclaimers.

Rev. android-12.1.0_1.0.0 — 24 October 2022

© 2022 NXP B.V. All rights reserved.

11 / 65

Figure 8. MX8-DSI-OLED1 MIPI panel

Figure 9. OV5640 CSI MIPI camera

Note:

- *i.MX 8M Nano EVK LPDDR4 board and i.MX 8M Nano EVK DDR4 board are supported in this release.*
- *To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "MIPI DSI" port.*
- *To test the MIPI panel display, connect the i.MX MIPI panel to the "MIPI DSI" port.*
- *To test the camera, connect the OV5640 CSI MIPI camera to the "MIPI CSI" port.*
- *For i.MX 8M Nano EVK LPDDR4 board, Bluetooth/Wi-Fi functions are supported.*
- *For i.MX 8M Nano EVK DDR4 board, Bluetooth/Wi-Fi functions are not maintained.*

4.2 Board images

The table below describes the location in the board partitions of the software images in `android-12.1.0_1.0.0_image_8mnevk.tar.gz`.

Table 3. Board images

Image name	Download target
`spl-imx8mn-dual.bin`	32 KB offset of MMC for i.MX 8M Nano EVK LPDDR4 board.
`spl-imx8mn-trusty-dual.bin`	32 KB offset of MMC for i.MX 8M Nano EVK LPDDR4 board.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

12 / 65

Table 3. Board images...*continued*

Image name	Download target
`spl-imx8mn-trusty-secure-unlock-dual.bin`	32K offset of mmc for i.MX 8M Nano EVK LPDDR4 board.
`bootloader-imx8mn-dual.img`	`bootloader_a` and `bootloader_b` partitions on i.MX 8M Nano EVK LPDDR4 board.
`bootloader-imx8mn-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions on i.MX 8M Nano EVK LPDDR4 board.
`bootloader-imx8mn-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partition on i.MX 8M Nano EVK LPDDR4 board.
`u-boot-imx8mn.imx`	32 KB offset of MMC for i.MX 8M Nano EVK LPDDR4 board.
`u-boot-imx8mn-ddr4.imx`	32 KB offset of MMC for i.MX 8M Nano EVK DDR4 board.
`u-boot-imx8mn-evk-uuu.imx`	Bootloader used by UUU for i.MX 8M Nano EVK LPDDR4 board. It is not flashed to MMC.
`u-boot-imx8mn-ddr4-evk-uuu.imx`	Bootloader used by UUU for i.MX 8M Nano EVK DDR4 board. It is not flashed to MMC.
`imx8mn_mcu_demo.img`	5120 KB offset of MMC.
`partition-table.img`	0 KB offset of MMC. It is used for single bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 KB offset of MMC. It is used for dual-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 KB offset of MMC. It is used for single-bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 KB offset of MMC. It is used for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions. This is the AOSP GKI boot image.
`boot-imx.img`	`boot_a` and `boot_b` partitions. This is the boot image built with i.MX kernel tree for debugging.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions when doing VTS test with GSI system image.
`vbmeta-imx8mn.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output on i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-rpmsg.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output and MCU image on i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support rm67199 MIPI panel output on i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions to support rm67191 MIPI panel output on i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-ddr4.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output on i.MX 8M Nano EVK DDR4 board.
`vbmeta-imx8mn-ddr4-rpmsg.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output and MCU image on i.MX 8M Nano EVK DDR4 board.

Table 3. Board images...*continued*

Image name	Download target
`vbmeta-imx8mn-ddr4-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support RM67199 MIPI panel output on i.MX 8M Nano EVK DDR4 board.
`vbmeta-imx8mn-ddr4-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions to support RM67191 MIPI panel output on i.MX 8M Nano EVK DDR4 board.
`system.img`	Logical partition `system_a` and logical partition `system_b` in super partition.
`system_ext.img`	Logical partition `system_ext_a` and logical partition `system_ext_b` in super partition.
`vendor.img`	Logical partition `vendor_a` and logical partition `vendor_b` in super partition.
`product.img`	Logical partition `product_a` and logical partition `product_b` in super partition.
`super.img`	Super partition.
`dtbo-imx8mn.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output on i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-rpmsg.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output and MCU image on i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions to support rm67199 MIPI panel output on i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions to support rm67191 MIPI panel output on i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-ddr4.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output on i.MX 8M Nano EVK DDR4 board.
`dtbo-imx8mn-ddr4-rpmsg.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output and MCU image on i.MX 8M Nano EVK DDR4 board.
`dtbo-imx8mn-ddr4-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions to support rm67199 MIPI panel output on i.MX 8M Nano EVK DDR4 board.
`dtbo-imx8mn-ddr4-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions to support rm67191 MIPI panel output on i.MX 8M Nano EVK DDR4 board.
`rpmb_key_test.bin`	Prebuilt test RPMB key. It can be used to set the RPMB key as fixed 32 bytes 0x00.
`testkey_public_rsa4096.bin`	Prebuilt AVB public key. It is extracted from the default AVB private key.

4.3 Flashing board images

The board image files can be flashed into the target board using Universal Update Utility (UUU).

For the UUU binary file, download it from GitHub: uuu release page on GitHub.

To achieve more flexibility, two script files are provided to invoke UUU to automatically flash all Android images.

- `uuu_imx_android_flash.sh` for Linux OS
- `uuu_imx_android_flash.bat` for Windows OS

AQSUG

User guide

All information provided in this document is subject to legal disclaimers.

Rev. android-12.1.0_1.0.0 — 24 October 2022

© 2022 NXP B.V. All rights reserved.

14 / 65

For this release, these two scripts are validated on UUU 1.4.182 version. Download corresponding version from GitHub:

- For Linux OS, download the file named `uuu`.
- For Windows OS, download the file named `uuu.exe`.

Because the two script files will directly invoke UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from GitHub as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   For Rev. C2 boards, change the first four bits of board's sw1101 to 1000 (from 1-4 bit) to enter serial download mode.
3. Power on the board. Use the USB cable on the board OTG port to connect your PC with the board.
4. Decompress `release_package/android-12.1.0_1.0.0_ image_8mnevk.tar.gz`. The package contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Nano board, related options are described as follows.

Table 4. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8M Nano EVK, it should be `imx8mn`. This option is **mandatory**.
`-a`	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b. If this option and `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8M Nano EVK, it can be followed with "28". If this option is not used, default `partition-table.img` or `partition-table-dual.img` is flashed.
`-m`	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader image with `uboot_feature` in their names. For i.MX 8M Nano EVK LPDDR4 board, it can be `dual`, `trusty-dual` and `trusty-secure-unlock-dual`. For i.MX 8M Nano EVK DDR4, it should be `ddr4`. If this option is not used, the default `u-boot-imx8mn.imx` is flashed.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8M Nano EVK LPDDR4 board, it can be `rpmsg`, `mipi-panel` or `mipi-panel-rm67191`. For i.MX 8M Nano EVK DDR4 board, it can be `ddr4`, `ddr4-rpmsg`, `ddr4-mipi-panel` or `ddr4-mipi-panel-rm67191`. If this option is not used, the default `dtbo-imx8mn.img` and `vbmeta-imx8mn.img` are flashed.

Table 4. Options for uuu_imx_android_flash tool...*continued*

Option	Description
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8M Nano EVK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and execute to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not execute UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

Obviously, `-m` should be used together with `-d rpmsg` or `-d ddr4-rpmsg`.

- On Linux system, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx8mn -a -e -u
  trusty-dual
```

- On Windows system, open the command-line interface in administrator mode. The corresponding command is as follows:

```
> uuu_imx_android_flash.bat -f imx8mn -a -e -u trusty-dual
```

When the command above is executed, `spl-imx8mn-trusty-dual.bin` is flashed, `bootloader-imx8mn-trusty-dual.img` with other default images are flashed into eMMC slot a for i.MX 8M Nano EVK LPDDR4 board.

Note:

- *`-u` followed with a parameter and containing `trusty` cannot be used together with `-t sd`, because Trusty OS cannot boot from SD card.*
- *To flash SD card, execute the tool with `-t sd`. To flash eMMC, it does not need to use `-t` option.*
- *If your SD card is 16 GB or uses onboard eMMC as the boot device, it does not need to use `-c` option.*
- *If your SD card is 32 GB, execute the tool with `-c 28`.*
- *For i.MX 8M Nano EVK LPDDR4 board:*
  - *To test dual bootloaders, execute the tool with `-u dual`.*
  - *To test Trusty OS and dual-bootloader both enabled condition, execute the tool with `-u trusty-dual`.*
  - *To test the demostration implementation of secure unlock, execute the tool with `-u trusty-secure-unlock-dual`. For secure unlock details, see the i.MX Android Security User's Guide (ASUG).*
  - *To test MIPI-DSI to HDMI output, it does not need to use the `-d` option.*
  - *To test MIPI-DSI to HDMI output and MCU image, execute the tool with `-d rpmsg`.*

- – *To test RM67199 MIPI panel output, execute the tool with* `-d mipi-panel.`
- – *To test RM67191 MIPI panel output, execute the tool with* `-d mipi-panel-rm67191.`
  - *For i.MX 8M Nano EVK DDR4 board:*
    - – *To test MIPI-DSI to HDMI output, execute the tool with* `-u ddr4 -d ddr4.`
    - – *To test DSI to HDMI output and MCU image, execute the wool with* `-u ddr4 -d ddr4-rpmsg.`
    - – *To test RM67199 MIPI panel output, execute the tool with* `-u ddr4 -d ddr4-mipi-panel.`
    - – *To test RM67191 MIPI panel output, execute the tool with* `-u ddr4 -d ddr4-mipi-panel-rm67191.`
  - *If* `uuu_imx_android_flash.bat` *is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:*

```
> net use z: \\192.168.1.1\daily_images
```

"z" in the command represents an available drive letter. It can be other available drive letter.

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, the command window displays information indicating that images are already flashed.
   Note:
   If the target device has a DOS partition table on it, the flash process fails for the first time. Push the reset key on the board and execute the flash script again.
7. Power off the board.
8. Change boot device as eMMC or SD card.
   For Rev. C boards:
   - Change SW1101 first four bits ([1-4]) to 0100 to boot from eMMC.
   - Change SW1101 first four bits ([1-4]) to 1100 to boot from SD card.

4.4 Booting

After downloading the images, boot the board by connecting it to the power supply.

4.4.1 Booting with single MIPI-DSI-to-HDMI/MIPI panel display

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

- i.MX 8M Nano EVK Board:

```
U-Boot > setenv bootargs console=ttymxc1,115200
 earlycon=ec_imx6q,0x30890000,115200 init=/init
 cma=800M@0x400M-0xb80M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware
 transparent_hugepage=never loop.max_part=7
 moal.mod_para=wifi_mod_para_sd8987.conf
 androidboot.lcd_density=240 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append
`androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
```

```
U-Boot > saveenv
```

4.4.2 Booting with single MIPI-DSI-to-HDMI display with an MCU image

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

• i.MX 8M Nano EVK Board:

```
U-Boot > setenv bootargs console=ttymxc1,115200
 earlycon=ec_imx6q,0x30890000,115200 init=/init
 cma=800M@0x400M-0xb80M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware
 transparent_hugepage=never loop.max_part=7
 moal.mod_para=wifi_mod_para_sd8987.conf
 androidboot.lcd_density=240 bootconfig
U-Boot > setenv bootcmd "bootmcu && boota"
U-Boot > saveenv
```

Note:

To use other dtbo images, do not add `bootmcu` *to* `bootcmd`*. The following command can recover bootcmd:*

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

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs 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.

5 Working with the i.MX 8M Plus EVK Board

5.1 Board hardware

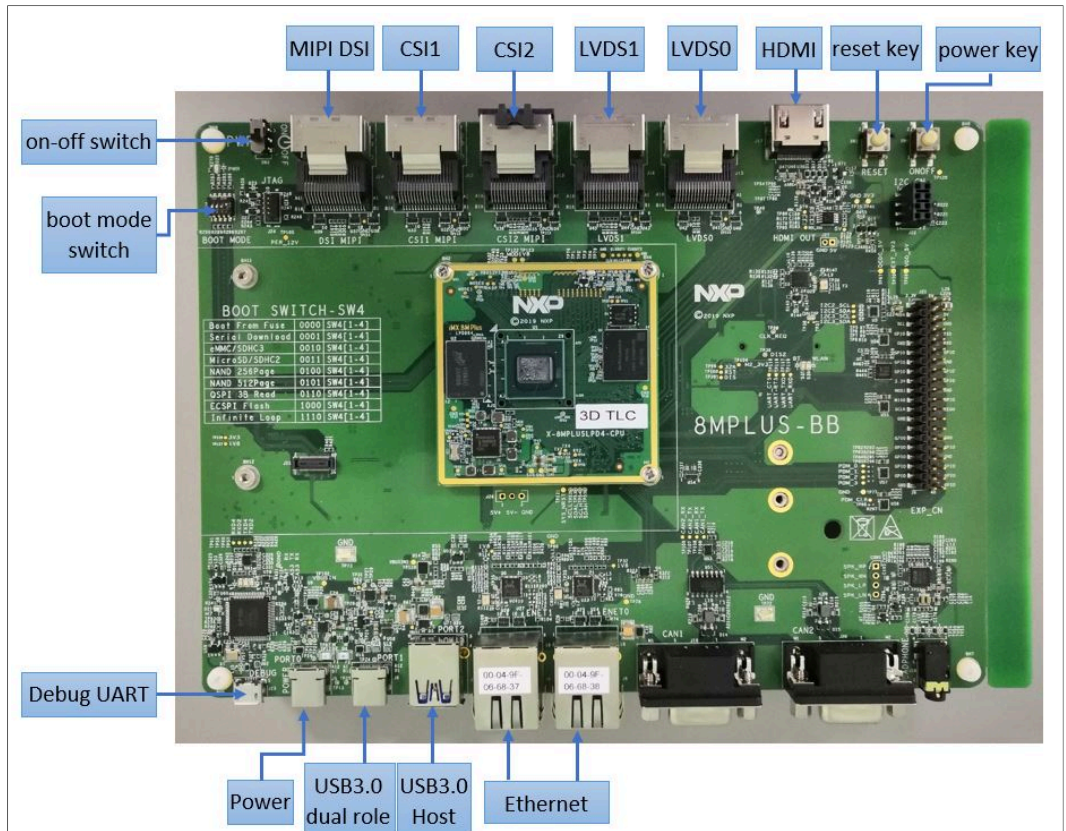The figure below shows the different components of the i.MX 8M Plus EVK board.

Figure 10.   i.MX 8M Plus EVK board

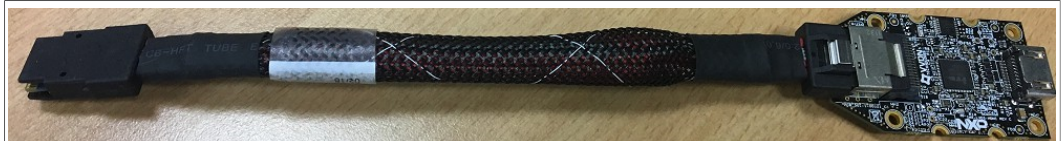**Figure 11. i.MX mini SAS cable with DSI-to-HDMI adapter**

Figure 12. i.MX mini SAS cable with LVDS-to-HDMI adapter

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

19 / 65

Figure 13. MX8-DSI-OLED1 MIPI panel

Figure 14. i.MX LVDS panel

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

21 / 65
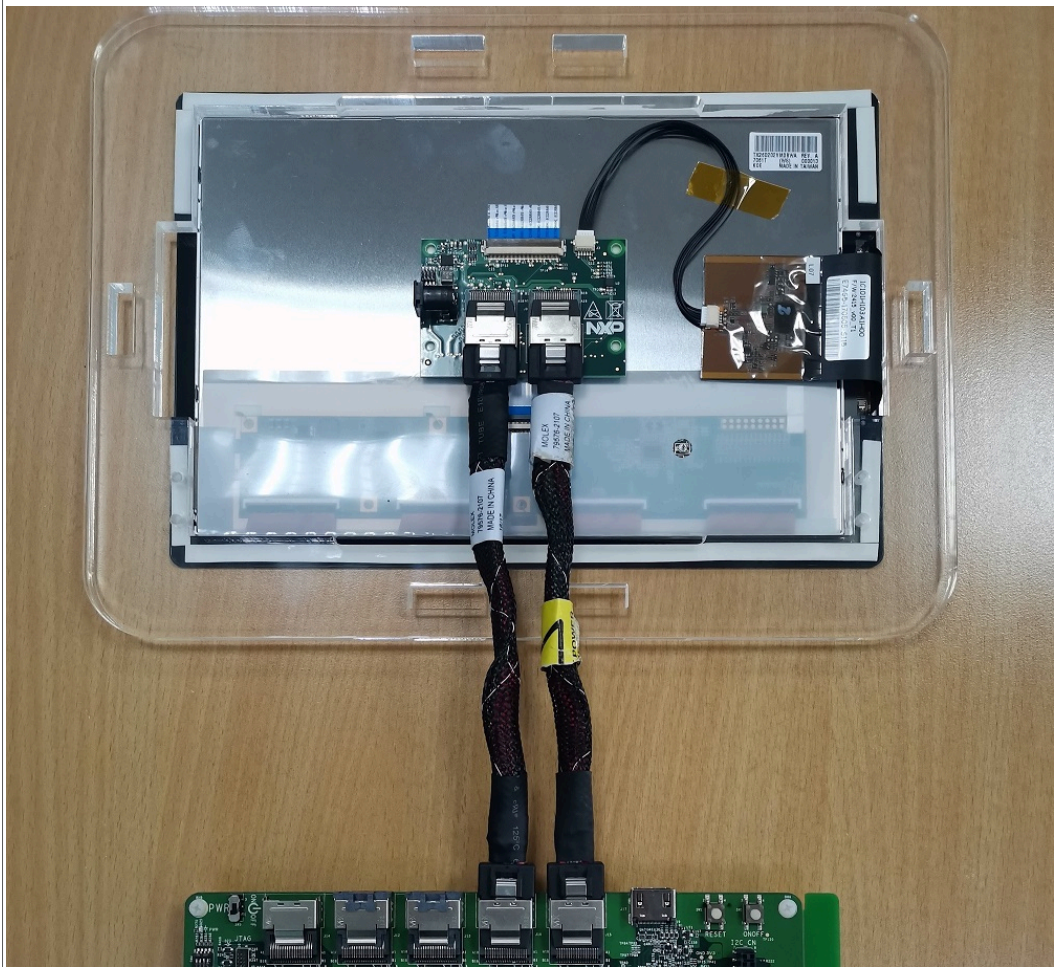
Figure 15. i.MX dual channel LVDS to HDMI adapter

Figure 16. OV5640 CSI MIPI camera

Figure 17. Basler CSI MIPI camera

Figure 18. Os08a20 CSI MIPI Camera

AQSUG

User guide Rev. android-12.1.0_1.0.0 — 24 October 2022

22 / 65

Note:

- *To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "MIPI DSI" port.*
- *To test the MIPI panel display, connect the i.MX MIPI panel to the "MIPI DSI" port.*
- *To test physical HDMI display, connect the HDMI cable to the "HDMI" port.*
- *To test the LVDS-to-HDMI display, use the i.MX mini SAS cable to connect the LVDS to HDMI adapter to the "LVDS0" port.*
- *To test the LVDS panel display, use two i.MX mini SAS cables to connect the LVDS panel to the "LVDS0" and "LVDS1" ports. Provide 5V power through the DC jack on the LVDS panel or connect pin3 with pin4 of pin header j7 beside the DC jack.*
- *To test dual channel LVDS to HDMI display, use two i.MX mini SAS cables to connect the dual LVDS to HDMI adapter to "LVDS0" and "LVDS1" port.*
- *To test camera, follow below rules:*
  - *Basler(CSI1) + Basler(CSI2) : Use* `dtbo-imx8mp.img` + `vbmeta-imx8mp.img`*.*
  - *Basler(CSI1) + Ov5640(CSI2) : Use* `dtbo-imx8mp-basler-ov5640.img` + `vbmeta-imx8mp-basler-ov5640.img`*.*
  - *Only Basler(CSI1) : Use* `dtbo-imx8mp-basler.img` + `vbmeta-imx8mp-basler.img`*.*
  - *Only Ov5640(CSI1) : Use* `dtbo-imx8mp-ov5640.img` + `vbmeta-imx8mp-ov5640.img`*.*
  - *Os08a20(CSI1) + Os08a20(CSI2) : Use* `dtbo-imx8mp-dual-os08a20.img` + `vbmeta-imx8mp-dual-os08a20.img`*.*
  - *Os08a20(CSI1) + Ov5640(CSI2) : Use* `dtbo-imx8mp-os08a20-ov5640.img` + `vbmeta-imx8mp-os08a20-ov5640.img`*.*
  - *Only os08a20(CSI1) : Use* `dtbo-imx8mp-os08a20.img` + `vbmeta-imx8mp-os08a20.img`*.*

5.2 Board images

The table below describes the location in the board partitions of the software images in `android-12.1.0_1.0.0_image_8mpevk.tar.gz`.

Table 5. Board images

Image name	Download target
`spl-imx8mp-dual.bin`	0 KB offset of eMMC `boot0` partition or 32 KB offset of SD card.
`spl-imx8mp-trusty-dual.bin`	0 KB offset of eMMC `boot0` partition.
`spl-imx8mp-trusty-secure-unlock-dual.bin`	0K offset of eMMC boot0 partition.
`bootloader-imx8mp-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`bootloader-imx8mp-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`bootloader-imx8mp-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`u-boot-imx8mp.imx`	0 KB offset of eMMC `boot0` partition or 32 KB offset of SD card.

Table 5. Board images...*continued*

Image name	Download target
`u-boot-imx8mp-evk-uuu.imx`	Bootloader used by UUU for i.MX 8M Plus board. It is not flashed to MMC.
`imx8mp_mcu_demo.img`	5120 KB offset of eMMC user partition or SD card.
`partition-table.img`	0 KB offset of eMMC user partition or SD card. It is used for single bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 KB offset of eMMC user partition or SD card. It is used for dual-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 KB offset of eMMC user partition or SD card. It is used for single bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 KB offset of eMMC user partition or SD card. It is used for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions. This is the AOSP GKI boot image.
`boot-imx.img`	`boot_a` and `boot_b` partitions. This is the boot image built with i.MX kernel tree for debugging.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions when doing VTS test with GSI system image.
`vbmeta-imx8mp.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and supoort two basler cameras plug in CSI1 and CSI2 port.
`vbmeta-imx8mp-basler-ov5640.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and supoort basler camera plug in CSI1 port and OV5640 camera plug in CSI2 port.
`vbmeta-imx8mp-dual-os08a20.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and supoort OS08A20 + OS08A20 camera.
`vbmeta-imx8mp-os08a20-ov5640.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and supoort os08a20(CSI1) + ov5640(CSI2) camera.
`vbmeta-imx8mp-os08a20.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and support only OS08A20 camera plug in CSI1 slot.
`vbmeta-imx8mp-basler.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and support only basler camera plug in CSI1 slot.
`vbmeta-imx8mp-ov5640.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and support only OV5640 CSI MIPI camera plug in CSI1 slot.
`vbmeta-imx8mp-lvds-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support LVDS panel output.
`vbmeta-imx8mp-lvds.img`	`vbmeta_a` and `vbmeta_b` partitions to support dual-channel LVDS-to-HDMI output.
`vbmeta-imx8mp-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support RM67199 MIPI panel output.

Table 5. Board images...*continued*

Image name	Download target
`vbmeta-imx8mp-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions to support RM67191 MIPI panel output.
`vbmeta-imx8mp-rpmsg.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output and MCU image.
`vbmeta-imx8mp-sof.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI-to-HDMI output, and support the Sound Open Firmware audio output.
`system.img`	Logical partition `system_a` and logical partition `system_b` in super partition.
`system_ext.img`	Logical partition `system_ext_a` and logical partition `system_ext_b` in super partition.
`vendor.img`	Logical partition `vendor_a` and logical partition `vendor_b` in super partition.
`product.img`	Logical partition `product_a` and logical partition `product_b` in super partition.
`super.img`	Super partition.
`dtbo-imx8mp.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output, and two basler cameras plug in CSI1 and CSI2 port.
`dtbo-imx8mp-basler-ov5640.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output, and supoort basler camera plug in CSI1 port and OV5640 camera plug in CSI2 port.
`dtbo-imx8mp-basler.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output, and support only basler camera plug in CSI1 slot.
`dtbo-imx8mp-ov5640.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output, and support only OV5640 CSI MIPI camera plug in CSI1 slot.
`dtbo-imx8mp-dual-os08a20.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output, and supoort OS08A20 + OS08A20 camera.
`dtbo-imx8mp-os08a20-ov5640.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output, and supoort os08a20(CSI1) + ov5640(CSI2) camera.
`dtbo-imx8mp-os08a20.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output, and support only OS08A20 camera plug in CSI1 slot.
`dtbo-imx8mp-lvds-panel.img`	`dtbo_a` and `dtbo_b` partitions to support LVDS panel output.
`dtbo-imx8mp-lvds.img`	`dtbo_a` and `dtbo_b` partitions to support dual-channel LVDS to HDMI output.
`dtbo-imx8mp-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions to support RM67199 MIPI panel output.
`dtbo-imx8mp-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions to support RM67191 MIPI panel output.
`dtbo-imx8mp-rpmsg.img`	`dtbo_a` and `dtbo_b` partitions to support MIPI-to-HDMI output and MCU image.
`dtbo-imx8mp-sof.img`	`dtbo_a` and `dtbo_b` partitions to support the Sound Open Firmware audio output.
`rpmb_key_test.bin`	Prebuilt test RPMB key, which can be used to set the RPMB key as fixed 32 bytes 0x00.

Table 5. Board images...*continued*

Image name	Download target
`testkey_public_ rsa4096.bin`	Prebuilt AVB public key, which is extracted from the default AVB private key.

5.3 Flashing board images

The board image files can be flashed into the target board using Universal Update Utility (UUU).

For the UUU binary file, download it from GitHub: uuu release page on GitHub.

To achieve more flexibility, two script files are provided to invoke UUU to automatically flash all Android images.

- `uuu_imx_android_flash.sh` for Linux OS
- `uuu_imx_android_flash.bat` for Windows OS

For this release, these two scripts are validated on UUU 1.4.182 version. Download corresponding version from GitHub:

- For Linux OS, download the file named `uuu`.
- For Windows OS, download the file named `uuu.exe`.

Because the two script files will directly invoke UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from GitHub as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   Change the board's SW4 (boot mode) to 0001 (from 1-4 bit) to enter serial download mode.
3. Power on the board.
   Use the USB cable to connect the USB 3.0 dual-role port (with silkprint "PORT1") on the board to your host PC.
4. Decompress `release_package/android-12.1.0_1.0.0_ image_8mpevk.tar.gz`. The package contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Plus board, related options are described as follows.

Table 6. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8M Plus EVK, it should be `imx8mp`. This option is **mandatory**.
`-a`	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b. If this option and `-a` option are not used, slots a and b are both flashed.

Table 6. Options for uuu_imx_android_flash tool...*continued*

Option	Description
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8M Plus EVK, it can be followed with "28". If this option is not used, default `partition-table.img` or `partition-table-dual.img` is flashed.
-m	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader image with `uboot_feature` in their names. For i.MX 8M Plus EVK board, it can be `dual`, `trusty-dual`, `trusty-secure-unlock-dual`. If this option is not used, the default `u-boot-imx8mp.imx` is flashed.
`-d dtb_feature`	Flashes dtbo and vbmeta images with `dtb_feature` in their names. For i.MX 8M Plus EVK board, it can be `basler-ov5640`, `basler`, `ov5640`, `lvds-panel`, `lvds`, `mipi-panel`, `rpmsg`, `dual-os08a20`, `os08a20-ov5640`, `os08a20`, `mipi-panel-rm67191`, `sof`. If this option is not used, the default `dtbo-imx8mp.img` and `vbmeta-imx8mp.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8M Plus EVK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and execute to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not execute UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

Please notice that, `-m` should be used together with `-d rpmsg`.

- On Linux system, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx8mp -a -e -u
  trusty-dual
```

- On Windows system, open the command-line interface in administrator mode. The corresponding command is as follows:

```
> uuu_imx_android_flash.bat -f imx8mp -a -e -u trusty-dual
```

When the command above is executed, `spl-imx8mp-trusty-dual.bin` is flashed, `bootloader-imx8mp-trusty-dual.img` with other default images are flashed into eMMC slot a for i.MX 8MPlus EVK.

Note:

- *`-u` followed with a parameter and containing `trusty` cannot be used together with `-t sd`, because Trusty OS cannot boot from SD card.*

- *To flash SD card, execute the tool with `-t sd`. To flash eMMC, it does not need to use `-t` option.*
- *If your SD card is 16 GB or uses onboard eMMC as the boot device, it does not need to use `-c` option.*
- *If your SD card is 32 GB, execute the tool with `-c 28`.*
- *To test dual bootloaders, execute the tool with `-u dual`.*
- *To test Trusty OS and dual-bootloader both enabled condition, execute the tool with `-u trusty-dual`.*
- *To test the demostration implementation of secure unlock, execute the tool with `-u trusty-secure-unlock-dual`. For secure unlock details, see the i.MX Android Security User's Guide (ASUG).*
- *To test features on the i.MX 8M Plus EVK board:*
  - *To test the MIPI-to-HDMI display, do not use the `-d` option. Physical HDMI and LVDS-to-HDMI are also supported in this condition.*
  - *To test the LVDS panel display, execute the tool with `-d lvds-panel`. MIPI-to-HDMI and physical HDMI are also supported in this condition.*
  - *To test dual channel LVDS to HDMI display, execute the tool with `-d lvds`.*
  - *To test the RM67199 MIPI panel display, execute the tool with `-d mipi-panel`. Physical HDMI and LVDS-to-HDMI are also supported in this condition.*
  - *To test RM67191 MIPI panel display, execute the tool with `-d mipi-panel-rm67191`. Physical HDMI and LVDS-to-HDMI are also supported in this condition.*
  - *To test the MIPI-to-HDMI display and MCU image, execute the tool with `-d rpmsg`. Physical HDMI and LVDS-to-HDMI are also supported in this condition.*
  - *To test two basler cameras, it does not need to use the `-d` option. The cameras can work by default.*
  - *To test the combination of one Basler camera and one OV5640 camera, execute the tool with `-d basler-ov5640`. Check the Android User's Guide (AUG) for more steps to make cameras work.*
  - *To test single Basler camera, execute the tool with `-d basler`. Check Android User's Guide (AUG) for more steps to make the camera work.*
  - *To test single OV5640 camera, execute the tool with `-d ov5640`. Check Android User's Guide (AUG) for more steps to make the camera work.*
  - *To test the combination of dual OS08A20 cameras, execute the tool with `-d dual-os08a20`. Check Android User's Guide (AUG) for more steps to make the camera work.*
  - *To test the combination of one Os08a20 camera and one Ov5640 camera, execute the tool with `-d os08a20-ov5640`. Check Android User's Guide (AUG) for more steps to make the camera work.*
  - *To test single OS08A20 camera, execute the tool with `-d os08a20`. Check Android User's Guide (AUG) for more steps to make the camera work.*
  - *To test the Sound Open Firmware audio output, execute the tool with `-d sof`.*
- *If `uuu_imx_android_flash.bat` is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:*

```
> net use z: \\192.168.1.1\daily_image
```

"z" in the command represents an available drive letter. It can be other available drive letter.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

28 / 65

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, the command window displays information indicating that images are already flashed.
7. Power off the board.
8. Change boot device as eMMC or SD card.
   • Change SW4 to switch the board back to 0011 (SD boot mode).
   • Change SW4 to switch the board back to 0010 (eMMC boot mode).

5.4 Booting

After downloading the images, you can boot the board by connecting it to the power supply.

5.4.1 Booting with single display: HDMI /MIPI-to-HDMI/MIPI panel/LVDS panel/ dual channel LVDS to HDMI or multiple displays

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

```
U-Boot > setenv bootargs console=ttymxc1,115200
 earlycon=ec_imx6q,0x30890000,115200 init=/init
 cma=1184M@0x400M-0x1000M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware loop.max_part=7
 transparent_hugepage=never moal.mod_para=wifi_mod_para.conf
 androidboot.lcd_density=240 swiotlb=65536 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

5.4.2 Booting with single MIPI-to-HDMI display and audio playback based on Cortex-M7 FreeRTOS

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

```
U-Boot > setenv bootargs console=ttymxc1,115200
 earlycon=ec_imx6q,0x30890000,115200 init=/init
 cma=1184M@0x400M-0x1000M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware loop.max_part=7
 transparent_hugepage=never moal.mod_para=wifi_mod_para.conf
 androidboot.lcd_density=240 swiotlb=65536 bootconfig
U-Boot > setenv bootcmd "bootmcu && boota"
U-Boot > saveenv
```

Note:

To use other dtbo images, do not add `bootmcu` to `bootcmd`. The following command can recover bootcmd:

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

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

29 / 65

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

5.4.3 Booting with single display: HDMI 4K display

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

```
U-Boot > setenv bootargs console=ttymxc1,115200
 earlycon=ec_imx6q,0x30890000,115200 init=/init
 cma=1184M@0x400M-0x1000M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware loop.max_part=7
 transparent_hugepage=never moal.mod_para=wifi_mod_para.conf
 androidboot.displaymode=4kp30 androidboot.lcd_density=480
 swiotlb=65536 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs.

Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
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.

6 Working with the i.MX 8M Quad EVK Board

6.1 Board hardware

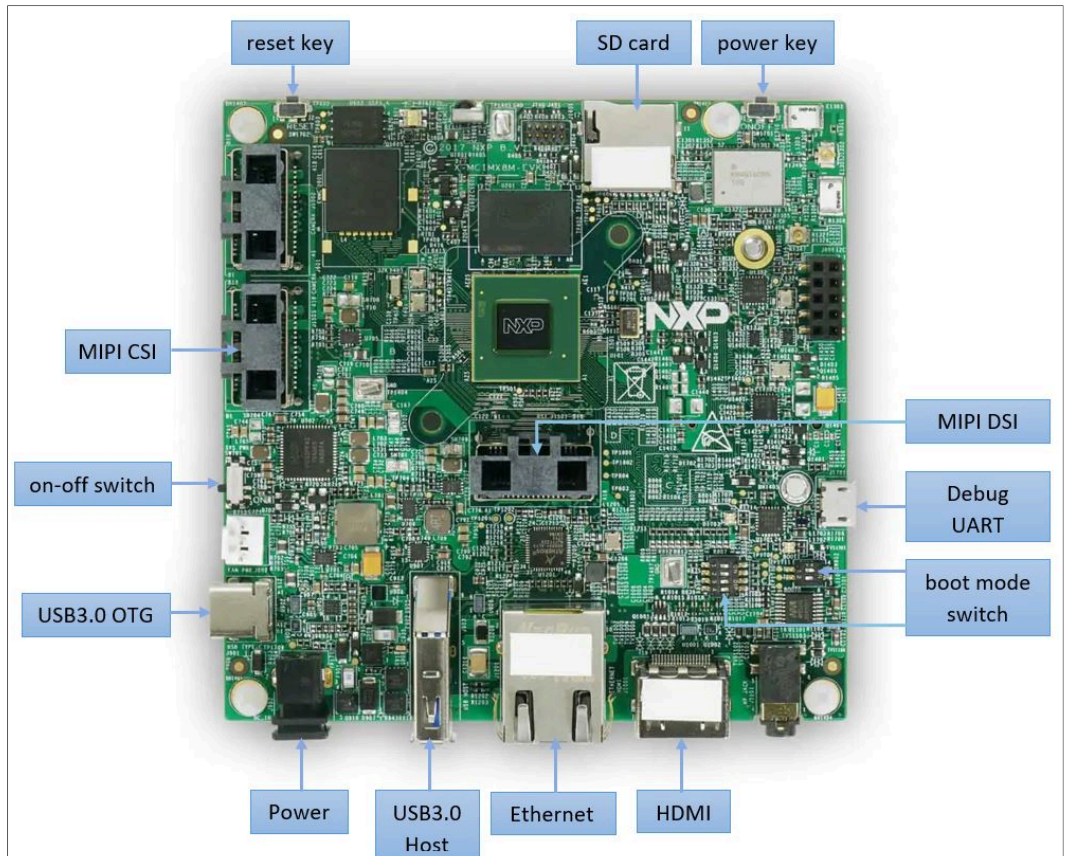The figures below show the different components of the i.MX 8M Quad EVK board.
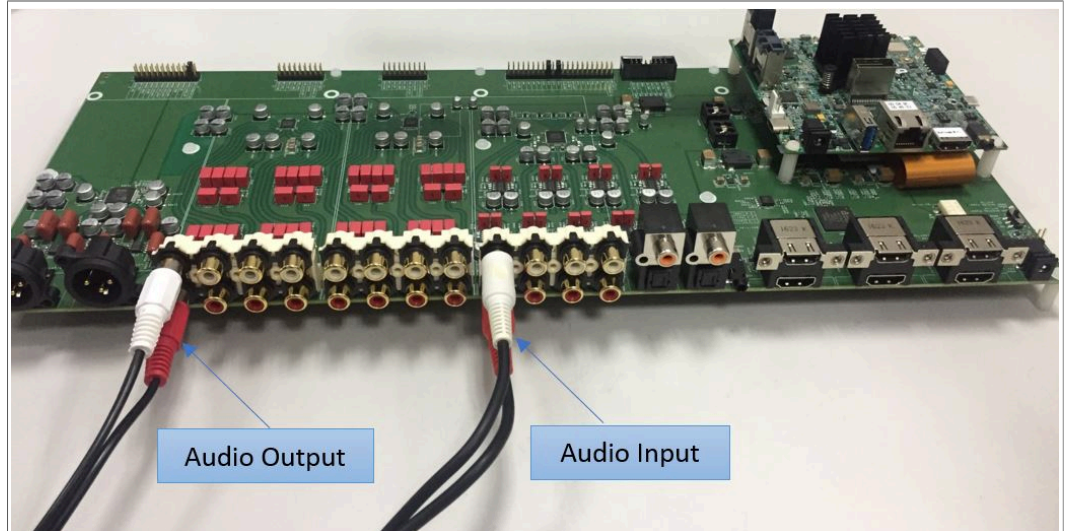
Figure 19. i.MX 8M Quad EVK board

Figure 20. i.MX 8M Quad EVK with audio board

Figure 21. i.MX mini SAS cable with DSI-to-HDMI adapter

Figure 22. MX8-DSI-OLED1 MIPI panel

Figure 23. OV5640 CSI MIPI camera

Note:

- *i.MX 8M Quad Rev. A board is supported in this release.*
- *To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "MIPI DSI" port.*
- *To test the MIPI panel display, connect the i.MX MIPI panel to the "MIPI DSI" port.*
- *To test the camera, connect the OV5640 CSI MIPI camera to the "MIPI CSI" port.*
- *Connect the PCIE8997(muRata LBEE5XV1YM) Wi-Fi&BT M.2 expansion card to the J1401 connector to have Wi-Fi&BT to work.*

6.2 Board images

The table below describes the location in the board partitions of the software images in `android-12.1.0_1.0.0_image_8mqevk.tar.gz`.

Table 7. Board images

Image name	Download target
`spl-imx8mq-dual.bin`	33 KB offset of MMC.
`spl-imx8mq-trusty-dual.bin`	33 KB offset of MMC.
`spl-imx8mq-trusty-secure-unlock-dual.bin`	33 KB offset of MMC.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

32 / 65

Table 7. Board images...*continued*

Image name	Download target
`bootloader-imx8mq-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`bootloader-imx8mq-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`bootloader-imx8mq-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`u-boot-imx8mq.imx`	33 KB offset of MMC.
`u-boot-imx8mq-evk-uuu.imx`	Bootloader used by UUU for i.MX 8M Quad board, which is not flashed to MMC.
`partition-table.img`	0 KB offset of MMC. It is used for single bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 KB offset of MMC. It is used for dual-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 KB offset of MMC. It is used for single bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 KB offset of MMC. It is used for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions. This is the AOSP GKI boot image.
`boot-imx.img`	`boot_a` and `boot_b` partitions. This is the boot image built with i.MX kernel tree for debugging.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions when doing VTS test with GSI system image.
`vbmeta-imx8mq.img`	`vbmeta_a` and `vbmeta_b` partitions to support i.MX 8M Quad Rev. A board HDMI output.
`vbmeta-imx8mq-mipi.img`	`vbmeta_a` and `vbmeta_b` partitions to support i.MX 8M Quad Rev. A board MIPI-DSI-to-HDMI output.
`vbmeta-imx8mq-dual.img`	`vbmeta_a` and `vbmeta_b` partitions to support i.MX 8M Quad Rev. A board HDMI and MIPI-DSI-to-HDMI dual output.
`vbmeta-imx8mq-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support i.MX 8M Quad Rev. A board RM67199 MIPI panel output.
`vbmeta-imx8mq-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions to support i.MX 8M Quad Rev. A board RM67191 MIPI panel output.
`system.img`	Logical partition `system_a` and logical partition `system_b` in super partition.
`system_ext.img`	Logical partition `system_ext_a` and logical partition `system_ext_b` in super partition.
`vendor.img`	Logical partition `vendor_a` and logical partition `vendor_b` in super partition.
`product.img`	Logical partition `product_a` and logical partition `product_b` in super partition.
`super.img`	Super partition.

Table 7. Board images...*continued*

Image name	Download target
`dtbo-imx8mq.img`	`dtbo_a` and `dtbo_b` partitions to support i.MX 8M Quad Rev. A board HDMI output.
`dtbo-imx8mq-mipi.img`	`dtbo_a` and `dtbo_b` partitions to support i.MX 8M Quad Rev. A board MIPI-DSI-to-HDMI output.
`dtbo-imx8mq-dual.img`	`dtbo_a` and `dtbo_b` partitions to support i.MX 8M Quad Rev. A board HDMI and MIPI-DSI-to-HDMI dual output.
`dtbo-imx8mq-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions to support i.MX 8M Quad Rev. A board RM67199 MIPI panel output.
`dtbo-imx8mq-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions to support i.MX 8M Quad Rev. A board RM67191 MIPI panel output.
`rpmb_key_test.bin`	Prebuilt test RPMB key, which can be used to set the RPMB key to fixed 32 bytes 0x00.
`testkey_public_rsa4096.bin`	Prebuilt AVB public key, which is extracted from the default AVB private key.

6.3 Flashing board images

The board image files can be flashed into the target board using Universal Update Utility (UUU).

For the UUU binary file, download it from GitHub: [uuu release page on GitHub](#).

To achieve more flexibility, two script files are provided to invoke UUU to automatically flash all Android images.

- `uuu_imx_android_flash.sh` for Linux OS
- `uuu_imx_android_flash.bat` for Windows OS

For this release, these two scripts are validated on UUU 1.4.182 version. Download corresponding version from GitHub:

- For Linux OS, download the file named `uuu`.
- For Windows OS, download the file named `uuu.exe`.

Because the two script files will directly invoke UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from GitHub as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   Change the board's SW802 (boot mode) to 01 (from 1-2 bit) to enter serial download mode.
3. Power on the board. Use the USB cable on the board USB 3.0 port to connect your PC with the board.
   Note:
   - *There are three USB ports on the i.MX 8M Quad EVK board: USB-to-UART, USB 3.0 host, and USB 3.0 OTG. The USB-to-UART can be referenced as the debug UART in the hardware image above. The debug UART can be used to watch the log of the hardware boot processing.*
   - *The SD card must be plugged in after the board is powered on.*

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

34 / 65

4. Decompress `release_package/android-12.1.0_1.0.0_image_8mqevk.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.

5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Quad board, related options are described as follows.

Table 8. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8M Quad EVK, it should be `imx8mq`. This option is **mandatory**.
`-a`	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b. If this option and `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8M Quad EVK, it can be followed with "28". If this option is not used, default `partition-table.img` or `partition-table-dual.img` is flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader image with `uboot_feature` in their names. For i.MX 8M Quad EVK, it can be `dual`, `trusty-dual`, `trusty-secure-unlock-dual`. If this option is not used, default `u-boot-imx8mq.imx` is flashed.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8M Quad EVK, it can be `dual`, `mipi-panel`, `mipi-panel-rm67191`, `mipi`. If this option is not used, default `dtbo-imx8mq.img` and `vbmeta-imx8mq.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in current working directory will be flashed.
`-t target_dev`	Specifies the target device. For i.MX 8M Quad EVK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no images are flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not executes UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

- On Linux system, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx8mq -a -e -u
trusty-dual
```

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

35 / 65

- On Windows system, open the command-line interface in administrator mode. The corresponding command is as follows:

```
> uuu_imx_android_flash.bat -f imx8mq -a -e -u trusty-dual
```

When the command above is executed, `spl-imx8mq-trusty-dual.bin` is flashed, `bootloader-imx8mq-trusty-dual.img` with other default images are flashed into eMMC slot a for i.MX 8MQuad EVK.

Note:

- *`-u` followed with a parameter and containing `trusty` cannot be used together with `-t sd`, because Trusty OS cannot boot from SD card.*
- *To flash the SD card, execute the tool with `-t sd`. To flash eMMC, it does not need to use `-t` option.*
- *If your SD card is 16 GB or the on-board eMMC is used as the boot device, it does not need to use `-c` option.*
- *If your SD card is 32 GB, execute the tool with `-c 28`.*
- *To test dual-bootloader, execute the tool with `-u dual`.*
- *To test Trusty OS and dual-bootloader both in enabled condition, execute the tool with `-u trusty-dual`.*
- *To test the demostration implementation of secure unlock, execute the tool with `-u trusty-secure-unlock-dual`. For secure unlock details, see the i.MX Android Security User's Guide (ASUG).*
- *To test feature on i.MX 8M Quad EVK Rev. A board:*
  - *To test HDMI output, it does not need to use `-d` option.*
  - *To test MIPI-DSI-to-HDMI output, execute the tool with `-d mipi`.*
  - *To test RM67199 MIPI panel output, execute the tool with `-d mipi-panel`.*
  - *To test RM67191 MIPI panel output, execute the tool with `-d mipi-panel-rm67191`.*
  - *To test HDMI and MIPI-DSI-to-HDMI dual-output, execute the tool with `-d dual`.*
- *If `uuu_imx_android_flash.bat` is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:*

```
> net use z: \\192.168.1.1\daily_images
```

"z" in the command represents an available drive letter. It can be other available drive letter.

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, you will get information on the command window indicating that images are already flashed.
   Note:
   If the target device has a DOS partition table on it, the flash process fails for the first time. Push the reset key on the board and execute the flash script again.
7. Power off the board.
8. Change boot device as eMMC or SD card. Change the board's SW802 (boot mode) to 10 (from 1-2 bit) to leave serial download mode.
   - Change SW801 to switch the board back to 1100 (SD boot mode).
   - Change SW801 to switch the board back to 0010 (eMMC boot mode).

6.4 Booting

After downloading the images, boot the board by connecting it to the power supply.

6.4.1 Booting with single display: HDMI display

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

• i.MX 8M Quad EVK Rev. A Board:

```
U-Boot > setenv bootargs console=ttymxc0,115200
 earlycon=ec_imx6q,0x30860000,115200 init=/init
 androidboot.gui_resolution=1080p cma=1280M
 androidboot.primary_display=imx-dcss
 firmware_class.path=/vendor/firmware loop.max_part=7
 transparent_hugepage=never androidboot.fbTileSupport=enable
 moal.mod_para=wifi_mod_para.conf androidboot.lcd_density=240
 cpuidle.off=1 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append
`androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

6.4.2 Booting with single display: MIPI-DSI-to-HDMI display

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

• i.MX 8M Quad EVK Rev. A Board:

```
U-Boot > setenv bootargs console=ttymxc0,115200
 earlycon=ec_imx6q,0x30860000,115200 init=/
init androidboot.lcd_density=160 cma=1280M
 androidboot.primary_display=mxsfb-drm firmware_class.path=/
vendor/firmware loop.max_part=7 transparent_hugepage=never
 androidboot.displaymode=720p moal.mod_para=wifi_mod_para.conf
 cpuidle.off=1 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append
`androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

6.4.3 Booting with dual displays: HDMI and MIPI-DSI-to-HDMI displays

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

• i.MX 8M Quad EVK Rev. A Board:

```
U-Boot > setenv bootargs console=ttymxc0,115200
 earlycon=ec_imx6q,0x30860000,115200 init=/init
 androidboot.gui_resolution=1080p cma=1280M
 androidboot.primary_display=imx-dcss firmware_class.path=/
vendor/firmware loop.max_part=7 transparent_hugepage=never
 moal.mod_para=wifi_mod_para.conf androidboot.lcd_density=240
 cpuidle.off=1 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```
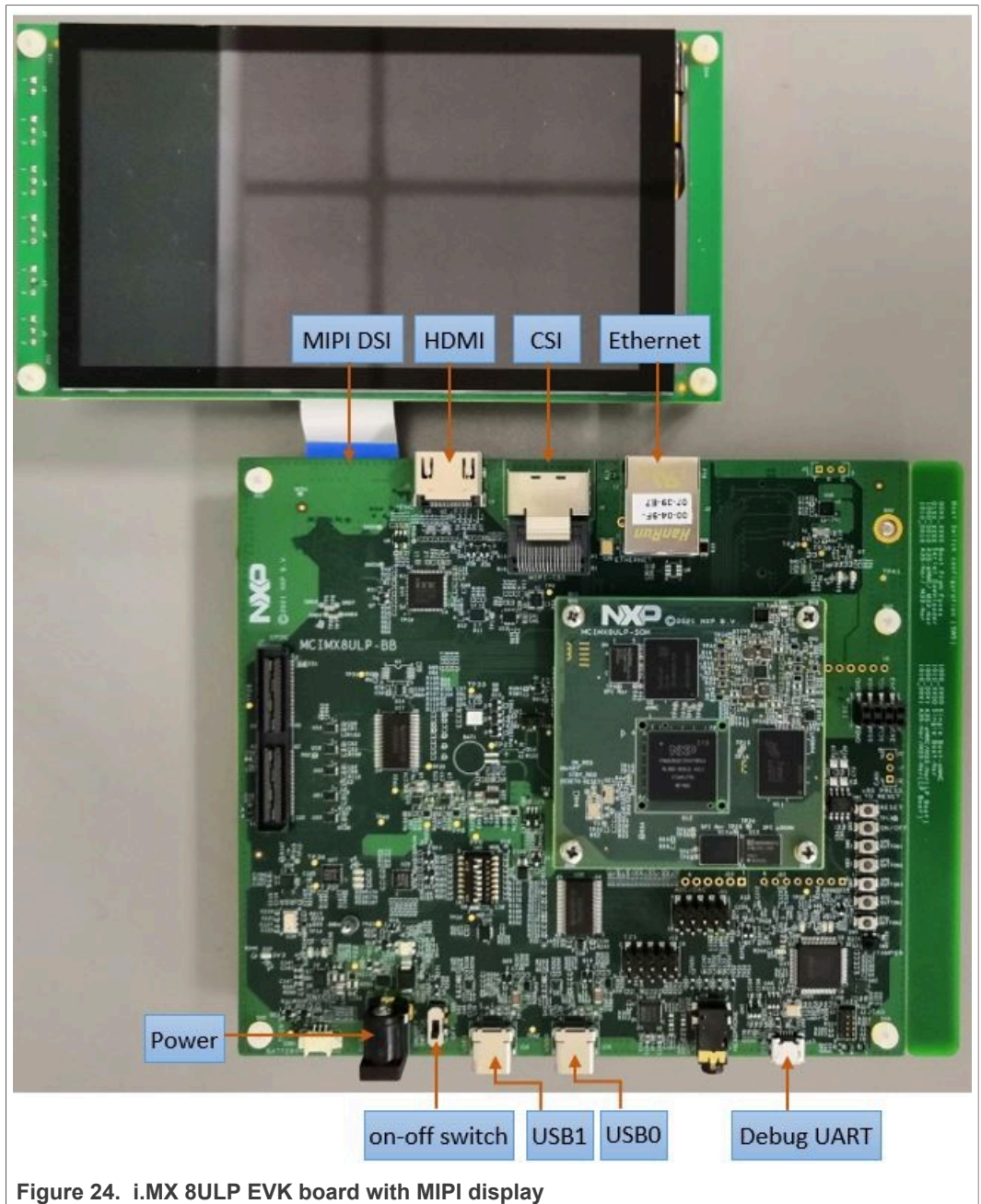
6.4.4 Booting with single display: MIPI panel

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

- i.MX 8M Quad EVK Rev. A Board:

```
U-Boot > setenv bootargs console=ttymxc0,115200
  earlycon=ec_imx6q,0x30860000,115200 init=/init cma=1280M
  androidboot.primary_display=imx-dcss firmware_class.path=/
vendor/firmware loop.max_part=7 transparent_hugepage=never
  moal.mod_para=wifi_mod_para.conf androidboot.lcd_density=240
  cpuidle.off=1 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
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.

7 Working with the i.MX 8ULP EVK Board

7.1 Board hardware

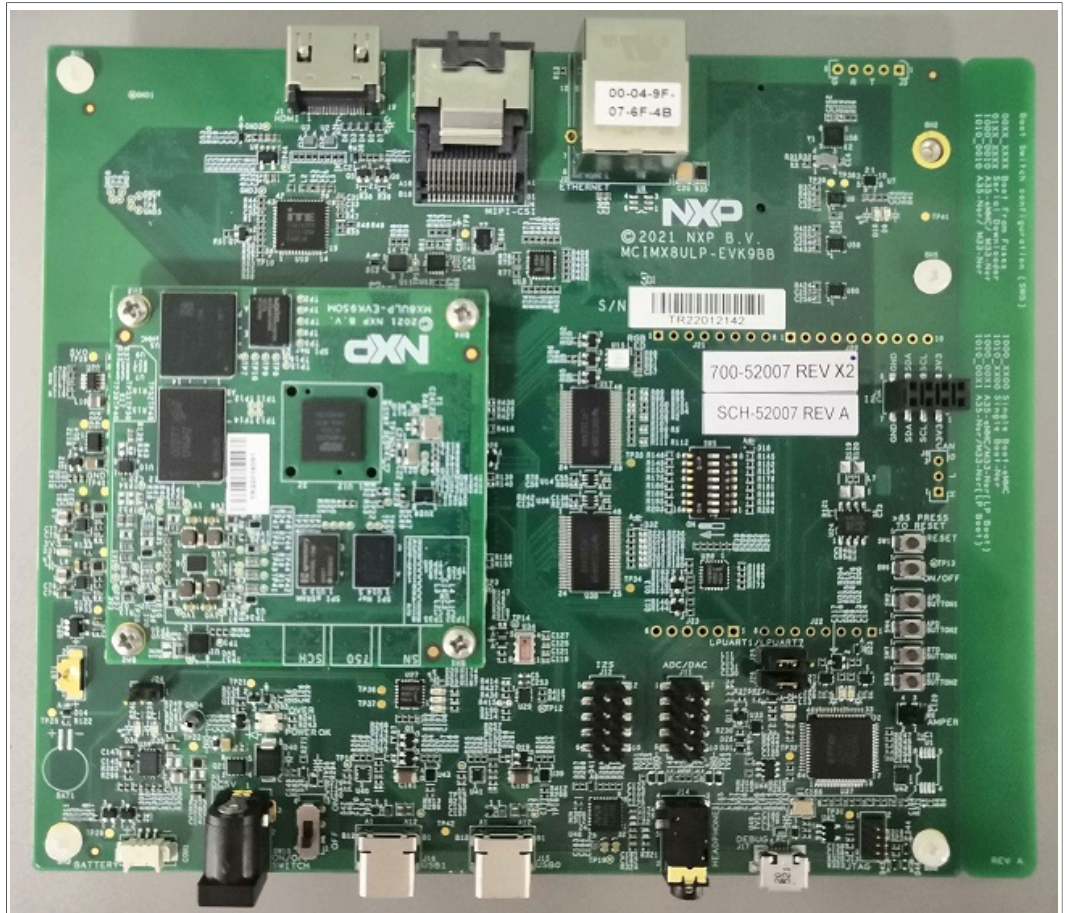The figure below shows the different components of the i.MX 8ULP EVK board.

Figure 24. i.MX 8ULP EVK board with MIPI display

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

39 / 65

Figure 25. i.MX 8ULP EVK 9x9 board

Figure 26. i.MX CSI MIPI camera

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

40 / 65

Figure 27. EPDC display

Note:

- *To test the camera, connect the i.MX CSI MIPI camera to the "CSI" port.*
- *To test EPDC display, connect the EPDC display to the base board of i.MX 8ULP EVK, the connector on the base board can be found near the silkprint of "NXP".*
- *The EPDC display board has a power switch, beside which is a DC jack. This DC jack should be connected to the 5V power to make the EPDC display board work.*
- *i.MX 8ULP EVK 9x9 board is named so because the i.MX 8ULP SoC used on the board is of size 9 mm x 9 mm. Also note that the base board does not have an EPDC interface.*

7.2 Board images

The table below describes the location in the board partitions of the software images in `android-12.1.0_1.0.0_image_8ulp-evk.tar.gz`.

Table 9. Board images

Image name	Download target
`spl-imx8ulp-dual.bin`	0 offset of eMMC boot partition for i.MX 8ULP EVK.
`spl-imx8ulp-trusty-dual.bin`	0 offset of eMMC boot partition for i.MX 8ULP EVK.

Table 9. Board images...*continued*

Image name	Download target
`spl-imx8ulp-trusty-9x9-dual.bin`	0 offset of eMMC boot partition for i.MX 8ULP EVK 9x9 board.
`spl-imx8ulp-trusty-secure-unlock-dual.bin`	0 offset of eMMC boot partition for i.MX 8ULP EVK.
`spl-imx8ulp-trusty-lpa-dual.bin`	0 offset of eMMC boot partition for i.MX 8ULP EVK with Low Power Audio(LPA) enabled.
`bootloader-imx8ulp-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8ULP EVK.
`bootloader-imx8ulp-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8ULP EVK.
`bootloader-imx8ulp-trusty-9x9-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8ULP EVK 9x9 board.
`bootloader-imx8ulp-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8ULP EVK.
`bootloader-imx8ulp-trusty-lpa-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8ULP EVK with Low Power Audio (LPA) enabled.
`u-boot-imx8ulp-9x9.imx`	0 offset of eMMC boot partition for i.MX 8ULP EVK 9x9 board.
`u-boot-imx8ulp-trusty-9x9.imx`	0 offset of eMMC boot partition for i.MX 8ULP EVK 9x9 board.
`u-boot-imx8ulp-9x9-evk-uuu.imx`	The bootloader used by UUU for i.MX 8ULP EVK 9x9 board, it is not flashed to MMC.
`u-boot-imx8ulp.imx`	0 offset of eMMC boot partition for i.MX 8ULP EVK.
`u-boot-imx8ulp-trusty.imx`	0 offset of eMMC boot partition for i.MX 8ULP EVK.
`u-boot-imx8ulp-trusty-secure-unlock.imx`	0 offset of eMMC boot partition for i.MX 8ULP EVK.
`u-boot-imx8ulp-evk-uuu.imx`	The bootloader used by uuu for i.MX 8ULP EVK, it will not be flashed to mmc.
`partition-table.img`	0 offset of eMMC user data area. It is used for single bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 offset of eMMC user data area. It is used for dual-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 offset of eMMC user data area. It is used for single-bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 offset of eMMC user data area. It is used for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions. This is the AOSP GKI boot image.
`boot-imx.img`	`boot_a` and `boot_b` partitions.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

42 / 65

Table 9. Board images...*continued*

Image name	Download target
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions. This is the boot image built with i.MX kernel tree for debugging.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions when doing VTS test with GSI system image.
`vbmeta-imx8ulp-9x9-hdmi.img`	`vbmeta_a` and `vbmeta_b` partitions to support HDMI output on i.MX 8ULP EVK 9x9 board.
`vbmeta-imx8ulp-9x9.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI panel output on i.MX 8ULP EVK 9x9 board.
`vbmeta-imx8ulp.img`	`vbmeta_a` and `vbmeta_b` partitions to support MIPI panel output on i.MX 8ULP EVK.
`vbmeta-imx8ulp-hdmi.img`	`vbmeta_a` and `vbmeta_b` partitions to support HDMI output on i.MX 8ULP EVK.
`vbmeta-imx8ulp-epdc.img`	`vbmeta_a` and `vbmeta_b` partitions to support EPDC output on i.MX 8ULP EVK.
`vbmeta-imx8ulp-sof.img`	`vbmeta_a` and `vbmeta_b` partitions to support the Sound Open Firmware audio output on i.MX 8ULP EVK.
`vbmeta-imx8ulp-lpa.img`	`vbmeta_a` and `vbmeta_b` partitions to support Low Power Audio on i.MX 8ULP EVK.
`system.img`	Logical partition `system_a` and logical partition `system_b` in super partition.
`system_ext.img`	Logical partition `system_ext_a` and logical partition `system_ext_b` in super partition.
`vendor.img`	Logical partition `vendor_a` and logical partition `vendor_b` in super partition.
`product.img`	Logical partition `product_a` and logical partition `product_b` in super partition.
`super.img`	Super partition.
`dtbo-imx8ulp-9x9-hdmi.img`	`dtbo_a` and `dtbo_b` paritions to support HDMI output on i.MX 8ULP EVK 9x9 board.
`dtbo-imx8ulp-9x9.img`	`dtbo_a` and `dtbo_b` paritions to support MIPI panel output on i.MX 8ULP EVK 9x9 board.
`dtbo-imx8ulp.img`	`dtbo_a` and `dtbo_b` paritions to support MIPI panel output on i.MX 8ULP EVK.
`dtbo-imx8ulp-hdmi.img`	`dtbo_a` and `dtbo_b` paritions to support HDMI output on i.MX 8ULP EVK.
`dtbo-imx8ulp-epdc.img`	`dtbo_a` and `dtbo_b` paritions to support EPDC output on i.MX 8ULP EVK.
`dtbo-imx8ulp-sof.img`	`dtbo_a` and `dtbo_b` partitions to support the Sound Open Firmware audio output on i.MX 8ULP EVK.
`dtbo-imx8ulp-lpa.img`	`dtbo_a` and `dtbo_b` partitions to support Low Power Audio on i.MX 8ULP EVK.
`rpmb_key_test.bin`	Prebuilt test RPMB key. It can be used to set the RPMB key as fixed 32 bytes 0x00.

Table 9. Board images...*continued*

Image name	Download target
`testkey_public_rsa4096.bin`	Prebuilt AVB public key. It is extracted from the default AVB private key.

7.3 Flashing board images

The board image files can be flashed into the target board using Universal Update Utility (UUU).

For the UUU binary file, download it from GitHub: <u>uuu release page on GitHub</u>.

To achieve more flexibility, two script files are provided to invoke UUU to automatically flash all Android images.

- `uuu_imx_android_flash.sh` for Linux OS
- `uuu_imx_android_flash.bat` for Windows OS

For this release, these two scripts are validated on UUU 1.4.182 version. Download corresponding version from GitHub:

- For Linux OS, download the file named `uuu`.
- For Windows OS, download the file named `uuu.exe`.

Because the two script files will directly invoke UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from GitHub as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   Change the boot switch SW5 to 00000010 (from 1-8 bit) to enter serial download mode.
3. Power on the board. Use the USB cable to connect the PC with the board through the USB0 port on the board.
4. Decompress `release_package/android-12.1.0_1.0.0_image_8ulpevk.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX the 8ULP EVK board, related options are described as follows.

Table 10. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8ULP EVK, it should be `imx8ulp`. This option is **mandatory**.
`-a`	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b. If this option and `-a` option are not used, slots a and b are both flashed.

Table 10. Options for uuu_imx_android_flash tool*...continued*

Option	Description
`-c card_size`	Specifies which partition table image file to flash. i.MX 8ULP EVK, it can be followed with "28". If this option is not used, default `partition-table.img` is flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader image with `uboot_feature` in their names. For i.MX 8ULP EVK, it can be `9x9`, `dual`, `trusty-dual`, `trusty-9x9-dual`, `trusty-secure-unlock-dual`, `trusty-lpa-dual`. If this option is not used, the default `u-boot-imx8ulp.imx` is flashed.
`-d dtb_feature`	Flashes dtbo and vbmeta images with `dtb_feature` in their names. For i.MX 8ULP EVK, it can be `9x9`, `9x9-hdmi`, `hdmi`, `epdc` and `lpa`. If this option is not used, default `dtbo-imx8ulp.img` and `vbmeta-imx8ulp.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in current working directory are flashed.
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not executes UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

- On Linux system, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx8ulp -a -e -u
  trusty-dual
```

- On Windows system, open the command-line interface in administrator mode. The corresponding command is as follows:

```
> uuu_imx_android_flash.bat -f imx8ulp -a -e -u trusty-dual
```

When the command above is executed, `spl-imx8ulp-trusty-dual.bin` is flashed, `bootloader-imx8ulp-trusty-dual.img` with other default images will be flashed into eMMC slot a for i.MX 8ULP EVK.

Note:

- *For i.MX 8ULP EVK:*
  - *To test dual bootloader, execute the tool with* `-u dual`.
  - *To test Trusty OS and dual bootloader both enabled condition, execute the tool with* `-u trusty-dual`.
  - *To test the demostration implementation of secure unlock, execute the tool with* `-u trusty-secure-unlock-dual`. *For secure unlock details, see the i.MX Android Security User's Guide (ASUG).*
  - *To test MIPI panel display, no need to use* `-d` *option.*
  - *To test HDMI display, execute the tool with* `-d hdmi`.

– *To test EPDC display, execute the tool with* `-d epdc`*.*
- *For i.MX 8ULP EVK 9x9 board:*
  – *To test MIPI panel display, execute the tool with* `-d 9x9`*.*
  – *To test HDMI display, execute the tool with* `-d 9x9-hdmi`*.*
- *If* `uuu_imx_android_flash.bat` *is used to flash images on a remote server through Samba, you need to map the remote resource to the local environment first. Take the following command as an example:*

```
> net use z: \\192.168.1.1\daily_images
```

"z" in the command represents an available drive letter. It can be other available drive letter.

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, the command window displays the information indicating that images are already flashed.
7. Power off the board.
8. Change boot device as eMMC.
   - Change SW5 to 00000001 to boot from eMMC.

7.4 Booting

After downloading the images, boot the board by connecting it to the power supply.

7.4.1 Booting with MIPI panel display

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

```
U-Boot > setenv bootargs console=ttyLP1,115200 earlycon
 init=/init firmware_class.path=/vendor/firmware
 loop.max_part=7 cma=800M transparent_hugepage=never
 androidboot.lcd_density=240 androidboot.primary_display=imx-
dcnano moal.mod_para=wifi_mod_para_sd416.conf bootconfig
U-Boot > saveenv
```

To disable selinux for `userdebug`/`eng` build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs.

Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

7.4.2 Booting with HDMI display

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

```
U-Boot > setenv bootargs console=ttyLP1,115200 earlycon
 init=/init firmware_class.path=/vendor/firmware
 loop.max_part=7 cma=800M transparent_hugepage=never
 androidboot.lcd_density=120 androidboot.displaymode=720x480p60
 androidboot.primary_display=imx-dcnano
 moal.mod_para=wifi_mod_para_sd416.conf bootconfig
U-Boot > saveenv
```

To disable selinux for `userdebug`/`eng` build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs.

Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

As the camera orientation is set to 90 for the default image (MIPI Panel, 720x1280), for HDMI display (720x480), the orientation needs to be set to **0**. To try the camera with the prebuilt image without modifying the code and rebuilding the images, follow the instructions below on the host after the system boots up:

```
adb reboot
adb remount
adb pull /vendor/etc/configs/camera_config_imx8ulp.json
# set "orientation" to 0 in the json file.
adb push camera_config_imx8ulp.json /vendor/etc/configs
adb reboot
```

7.4.3 Booting with EPDC display

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

```
U-Boot > setenv bootargs console=ttyLP1,115200
 earlycon init=/init firmware_class.path=/
vendor/firmware loop.max_part=7 cma=800M
 transparent_hugepage=never androidboot.lcd_density=240
 moal.mod_para=wifi_mod_para_sd416.conf bootconfig
U-Boot > saveenv
```

To disable selinux for `userdebug`/`eng` build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs.

Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs 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.

8   Working with the i.MX 8QuadMax MEK Board

8.1 Board hardware

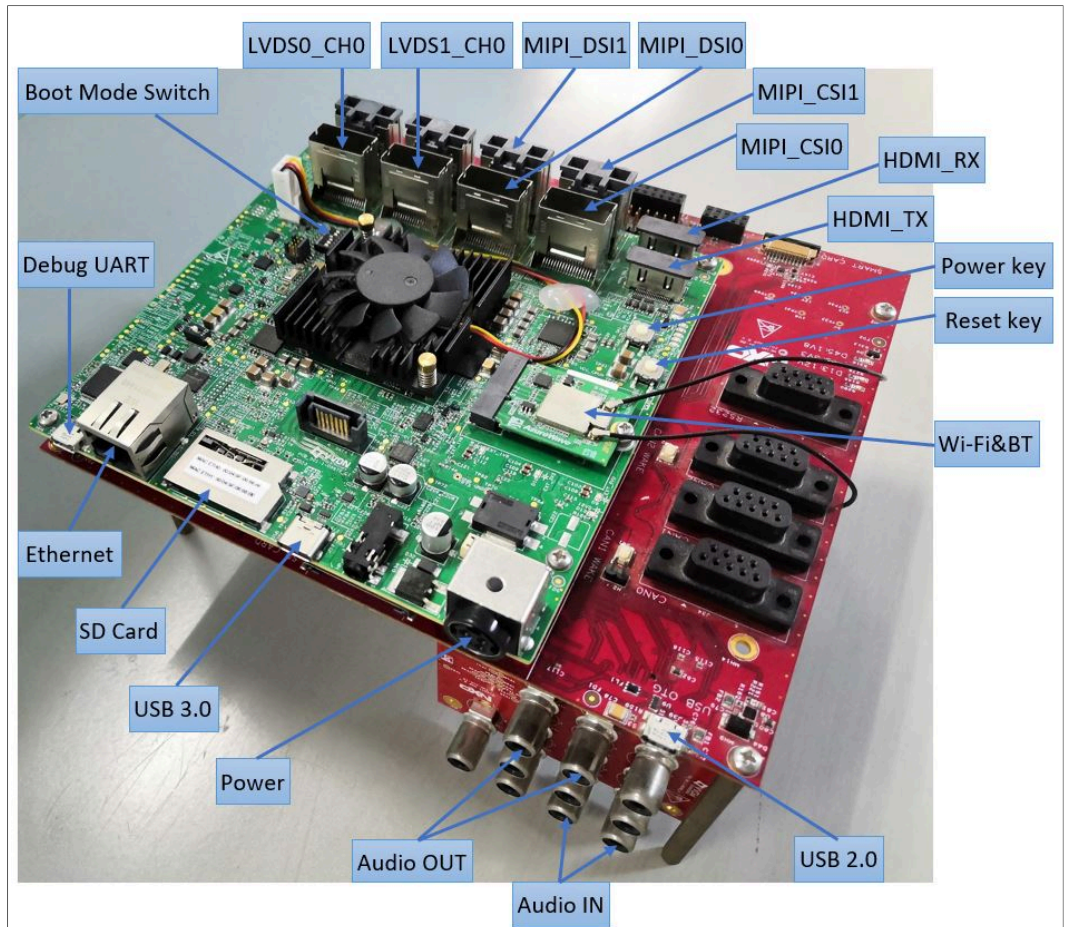The figures below show the different components of the i.MX 8QuadMax MEK board.

Figure 28. i.MX 8QuadMax MEK board

Figure 29. i.MX mini SAS cable with DSI-to-HDMI adapter

Figure 30. i.MX mini SAS cable with LVDS-to-HDMI adapter

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

48 / 65

Figure 31. MX8-DSI-OLED1 MIPI panel

Figure 32. i.MX MIPI camera

Note:

- *To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "HDMI DSI" port.*
- *To test the LVDS-to-HDMI display, use the i.MX mini SAS cable to connect the LVDS-to-HDMI adapter to the "LVDS0/LVDS1" port.*
- *To test the camera, connect two OV5640 CSI MIPI cameras to the "MIPI Camera0" and "MIPI Camera1" ports.*
- *To test the MIPI panel display, connect the i.MX MIPI panel to the "MIPI DSI" port. OLED1 (RM67191) MIPI panel is supported, while OLED1A (RM67199) is not supported.*
- *To test single camera, connect OV5640 CSI MIPI camera to the "MIPI Camera0" or "MIPI Camera1" port.*
- *To test dual displays, use the the i.MX mini SAS cable to connect both "LVDS0" and "LVDS1" ports with the LVDS-to-HDMI adapter.*

8.2 Board images

The table below describes the location in the board partitions of the software images in `android-12.1.0_1.0.0_image_8qmek.tar.gz`.

Table 11. Board images

Image name	Download target
`spl-imx8qm-dual.bin`	0K offset of eMMC or 32K offset of SD card.
`bootloader-imx8qm-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`spl-imx8qm-trusty-dual.bin`	0K offset of eMMC.
`bootloader-imx8qm-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`spl-imx8qm-trusty-secure-unlock-dual.bin`	0K offset of eMMC.
`bootloader-imx8qm-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions.
`u-boot-imx8qm.imx`	0 KB offset of eMMC and 32 KB offset of SD card.
`u-boot-imx8qm-hdmi.imx`	0 KB offset of eMMC and 32 KB offset of SD card.
`u-boot-imx8qm-md.imx`	0 KB offset of eMMC and 32 KB offset of SD card.
`u-boot-imx8qm-mek-uuu.imx`	Bootloader used by UUU for i.MX 8QuadMax MEK board. It is not flashed to MMC.
`partition-table.img`	Programs to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for single bootloader condition and for boot storage larger than 13 GB.
`partition-table-dual.img`	Programs to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for dual-bootloader condition and for boot storage larger than 13 GB.
`partition-table-28GB.img`	Programs to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for single bootloader condition and for boot storage larger than 28 GB.
`partition-table-28GB-dual.img`	Programs to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for dual-bootloader condition and for boot storage larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions when doing VTS test with GSI system image.
`vbmeta-imx8qm.img`	`vbmeta_a` and `vbmeta_b` partitions to support LVDS-to-HDMI/MIPI-DSI-to-HDMI display.
`vbmeta-imx8qm-md.img`	`vbmeta_a` and `vbmeta_b` partitons to support multiple displays.
`vbmeta-imx8qm-hdmi.img`	`vbmeta_a` and `vbmeta_b` partitions to support physical HDMI display.
`vbmeta-imx8qm-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support OLED1 (RM67191) MIPI panel display.
`vbmeta-imx8qm-hdmi-rx.img`	`vbmeta_a` and `vbmeta_b` partitions to support HDMI input.
`vbmeta-imx8qm-lvds1-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support lvds panel display.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

50 / 65

Table 11. Board images...*continued*

Image name	Download target
vbmeta-imx8qm-sof.img	vbmeta_a and vbmeta_b partitions to support the Sound Open Firmware audio output.
system.img	Logical partition system_a and logical partition system_b in super partition.
system_ext.img	Logical partition system_ext_a and logical partition system_ext_b in super partition.
vendor.img	Logical partition vendor_a and logical partition vendor_b in super partition.
product.img	Logical partition product_a and logical partition product_b in super partition.
super.img	Super partition.
dtbo-imx8qm.img	dtbo_a and dtbo_b partitions to support LVDS-to-HDMI/MIPI-DSI-to-HDMI display.
dtbo-imx8qm-md.img	dtbo_a and dtbo_b partitions to support multiple displays.
dtbo-imx8qm-hdmi.img	dtbo_a and dtbo_b partitions to support physical HDMI display.
dtbo-imx8qm-mipi-panel.img	dtbo_a and dtbo_b partitions to support OLED1 (RM67191) MIPI panel display.
dtbo-imx8qm-hdmi-rx.img	dtbo_a and dtbo_b partitions to support HDMI input.
dtbo-imx8qm-lvds1-panel.img	dtbo_a and dtbo_b partitions to support LVDS panel display.
dtbo-imx8qm-sof.img	dtbo_a and dtbo_b partitions to support the Sound Open Firmware audio output.
rpmb_key_test.bin	Prebuilt test RPMB key. It can be used to set the RPMB key to fixed 32 bytes 0x00.
testkey_public_rsa4096.bin	Prebuilt AVB public key, which is extracted from the default AVB private key.

8.3 Flashing board images

The board image files can be flashed into the target board using Universal Update Utility (UUU).

For the UUU binary file, download it from GitHub: uuu release page on GitHub.

To achieve more flexibility, two script files are provided to invoke UUU to automatically flash all Android images.

- uuu_imx_android_flash.sh for Linux OS
- uuu_imx_android_flash.bat for Windows OS

For this release, these two scripts are validated on UUU 1.4.182 version. Download corresponding version from GitHub:

- For Linux OS, download the file named uuu.
- For Windows OS, download the file named uuu.exe.

Because the two script files will directly invoke UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

51 / 65

Perform the following steps to download the board images:

1. Download the UUU binary file from GitHub as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   Change the board's SW2 (boot mode) to 001000 (from 1-6 bit) to enter serial download mode.
3. Power on the board. Use the USB cable on the board USB 3.0 port to connect your PC with the board.
   Note:
   • *There are three USB ports on the i.MX 8QuadMax MEK board: USB-to-UART, USB 2.0, and USB 3.0.*
   • *The USB-to-UART port can be referenced as debug UART, which can be used to watch the log of the hardware boot processing.*
   • *USB 2.0 is USB Host and USB 3.0 is USB OTG.*
4. Decompress `release_package/android-12.1.0_1.0.0_image_ 8qmek.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8QuadMax board, related options are described as follows.

Table 12. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8QuadMax MEK, it should be `imx8qm`. This option is **mandatory**.
`-a`	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b. If this option and `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8Quad Max, it can be followed with "28". If this option is not used, default `partition-table.img` is flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader image with `uboot_feature` in their names. For i.MX 8QuadMax MEK, it can be `hdmi`, `md`, `dual`, `trusty-dual`, or `trusty-secure-unlock-dual`. If this option is not used, the default `u-boot-imx8qm.imx` is flashed.
`-d dtb_feature`	Flashes dtbo and vbmeta images with `dtb_feature` in their names. For i.MX 8QuadMax MEK, it can be `hdmi-rx`, `hdmi`, `md`, `mipi-panel`, or `sof`. If this option is not used, default `dtbo-imx8qm.img` and `vbmeta-imx8qm.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8QuadMax, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

52 / 65

Table 12. Options for uuu_imx_android_flash tool...*continued*

Option	Description
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not executes UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

- On Linux system, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx8qm -a -e -u
  trusty-dual
```

- On Windows system, open the command-line interface in administrator mode. The corresponding command is as follows:

```
> uuu_imx_android_flash.bat -f imx8qm -a -e -u trusty-dual
```

When the command above is executed, `spl-imx8qm-trusty-dual.bin` is flashed, `bootloader-imx8qm-trusty-dual.img` with other default images are flashed into eMMC slot a for i.MX 8QuadMax MEK.

Note:

- *`-u` followed with a parameter and containing `trusty` cannot be used together with `-t sd`, because Trusty OS cannot boot from SD card.*

- *To flash the SD card, execute the tool with `-t sd`. To flash eMMC, it does not need to use `-t` option.*

- *If your SD card is 16 GB or the on-board eMMC is used as the boot device, it does not need to use `-c` option.*

- *If your SD card is 32 GB, execute the tool with `-c 28`.*

- *To test dual bootloader, execute the tool with `-u dual`.*

- *To test Trusty OS and dual bootloader both enabled condition, execute the tool with `-u trusty-dual`.*

- *To test the demostration implementation of secure unlock, execute the tool with `-u trusty-secure-unlock-dual`. For secure unlock details, see the i.MX Android Security User's Guide (ASUG).*

- *To test LVDS-to-HDMI/MIPI-DSI-to-HDMI display, it does not need to use the `-d` option. Four display ports of LVDS0_CH0, LVDS1_CH0, MIPI_DSI0, and MIPI_DSI1 can be used for display individualy or together.*

- *To test RM67191 MIPI panel display, execute the tool with `-d mipi-panel`. MIPI_DSI0 port is used for display.*

- *To test physical HDMI display, execute the tool with `-u hdmi -d hdmi`. HDMI_TX port is used for display.*

- *To test physical HDMI RX as external camera input device, execute the tool with `-d hdmi-rx`. HDMI_RX port is used for HDMI RX. In addition, OV5640 on CSI1 could work well at the same time.*

AQSUG

User guide

All information provided in this document is subject to legal disclaimers.

Rev. android-12.1.0_1.0.0 — 24 October 2022

© 2022 NXP B.V. All rights reserved.

53 / 65

- *To test the Sound Open Firmware audio output, execute the tool with* `-d sof`.
- *To test multiple displays, execute the tool with* `-u md -d md`. *Four display ports can be used for display:*
  - *If HDMI_TX is used, the other three ports are LVDS0_CH0, LVDS1_CH0, and MIPI_DSI1.*
  - *If HDMI_TX is not used, the four ports are LVDS0_CH0, LVDS1_CH0, MIPI_DSI0, and MIPI_DSI1.*
- *If* `uuu_imx_android_flash.bat` *is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:*

```
> net use z: \\192.168.1.1\daily_images
```

"z" in the command represents an available drive letter. It can be other available drive letter.

6. Wait for the `uuu_imx_android_flash execution` to complete. If there is not any error, the command window displays the information indicating that images are already flashed.
   Note:
   If the target device has a DOS partition table on it, the flash process fails for the first time. Push the reset key on the board and execute the flash script again.
7. Power off the board.
8. Change boot device as eMMC or SD card.
   - Change SW2 to switch the board back to 000100 (from 1-6 bit) to enther eMMC boot mode.
   - Change SW2 to sithch the board back to 001100 (from 1-6 bit) to enther SD boot mode.

8.4 Booting

After downloading the images, boot the board by connecting it to the power supply.

8.4.1 Booting with LVDS-to-HDMI/MIPI-DSI-to-HDMI display

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon init=/
init cma=928M@0x960M-0xfc0M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware loop.max_part=7
 transparent_hugepage=never moal.mod_para=wifi_mod_para.conf
 androidboot.lcd_density=240 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

8.4.2 Booting with physical HDMI display

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon init=/
init androidboot.fbTileSupport=enable cma=928M@0x960M-0xfc0M
 androidboot.primary_display=imx-drm firmware_class.path=/
vendor/firmware loop.max_part=7 transparent_hugepage=never
 moal.mod_para=wifi_mod_para.conf androidboot.lcd_density=240
 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append
`androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

8.4.3 Booting with dual LVDS-to-HDMI displays

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon init=/
init cma=928M@0x960M-0xfc0M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware loop.max_part=7
 transparent_hugepage=never moal.mod_para=wifi_mod_para.conf
 androidboot.lcd_density=240 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append
`androidboot.selinux=permissive` to the U-Boot's bootargs. Boot environment
variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

8.5 Board reboot

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

9 Working with the i.MX 8QuadXPlus MEK Board

9.1 Board hardware

The figures below show the different components of the i.MX 8QuadXPlus MEK board.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

55 / 65
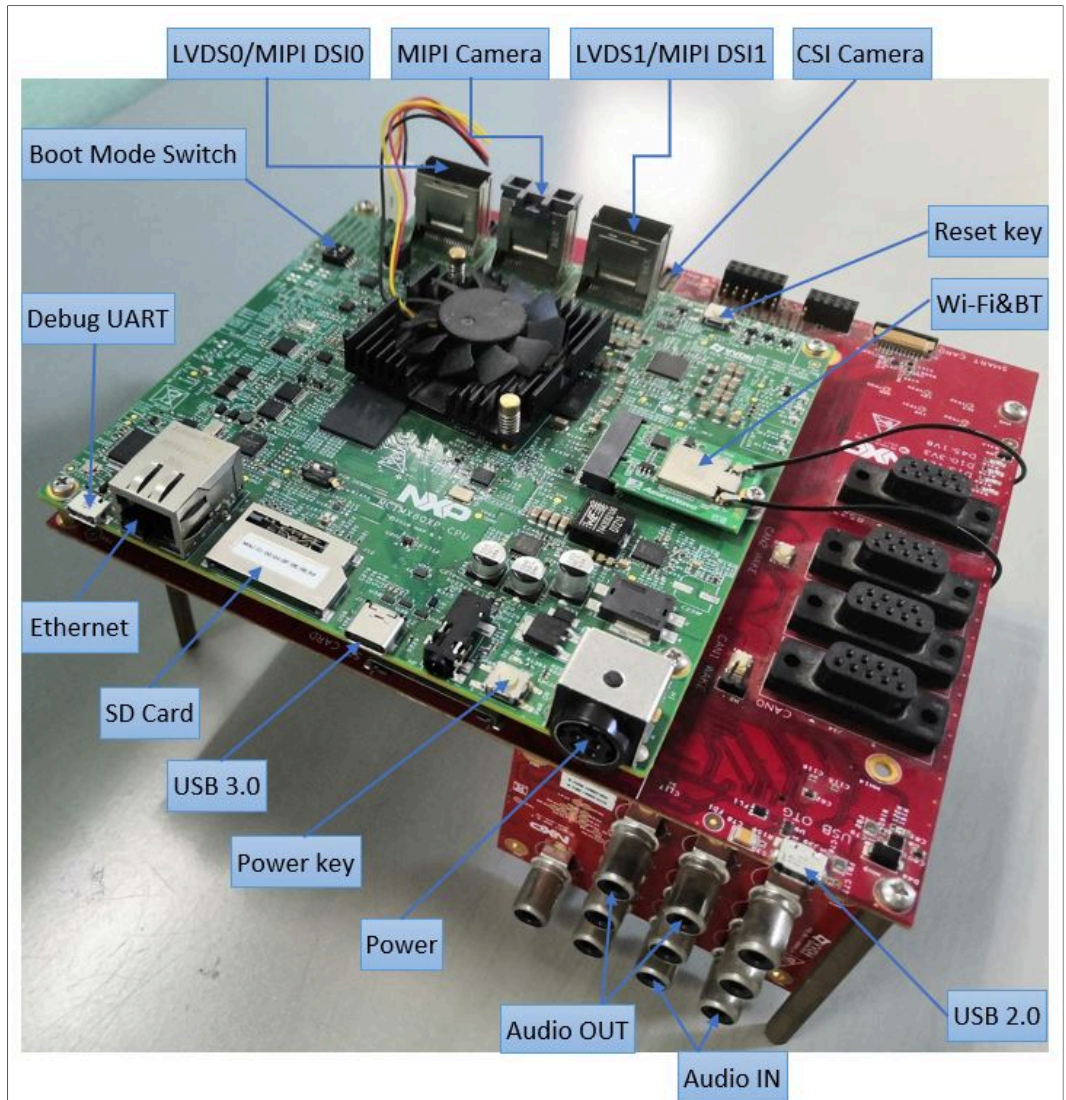
Figure 33. i.MX 8QuadXPlus MEK board

Figure 34. i.MX mini SAS cable with DSI-to-HDMI adapter

Figure 35. i.MX mini SAS cable with LVDS-to-HDMI adapter

Figure 36. OV5640 CSI MIPI camera

Note:

- *To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI to HDMI adapter to the "MIPI DSI" port.*
- *To test the LVDS-to-HDMI display, use the i.MX mini SAS cable to connect the LVDS-to-HDMI adapter to the "LVDS0/LVDS1" port.*
- *To test a single camera, connect the OV5640 CSI MIPI camera to the "MIPI Camera" port or connect OV5640 Camera to the "CSI Camera" port.*

9.2 Board images

The table below describes the location in the board partitions of the software images in `android-12.1.0_1.0.0_image_8qmek.tar.gz`.

Table 13. Board images

Image name	Download target
`spl-imx8qxp-dual.bin`	32K offset of mmc for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
`bootloader-imx8qxp-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
`spl-imx8qxp-c0-dual.bin`	0K offset of eMMC or 32K offset of SD card for i.MX 8QuadXPlus MEK board with silicon revision C0 chip.
`bootloader-imx8qxp-c0-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8QuadXPlus MEK board with silicon revision C0 chip.
`spl-imx8qxp-trusty-dual.bin`	32k offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
`bootloader-imx8qxp-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
`spl-imx8qxp-trusty-secure-unlock-dual.bin`	32K offset of mmc for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
`bootloader-imx8qxp-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
`spl-imx8qxp-trusty-c0-dual.bin`	0K offset of emmc for i.MX 8QuadXPlus MEK board with silicon revision C0 chip.
`bootloader-imx8qxp-trusty-c0-dual.img`	`bootloader_a` and `bootloader_b` partitions for i.MX 8QuadXPlus MEK board with silicon revision C0 chip.
`u-boot-imx8qxp.imx`	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.

AQSUG

User guide | **Rev. android-12.1.0_1.0.0 — 24 October 2022**

57 / 65

Table 13. Board images...*continued*

Image name	Download target
`u-boot-imx8qxp-c0.imx`	0 KB offset of MMC or 32 KB offset of SD card for i.MX 8QuadXPlus MEK board with silicon revision C0 chip.
`u-boot-imx8qxp-mek-uuu.imx`	Bootloader used by UUU for i.MX 8QuadXPlus MEK board with silicon revision B0 chip. It is not flashed to MMC.
`u-boot-imx8qxp-mek-c0-uuu.imx`	Bootloader used by UUU for i.MX 8QuadXPlus MEK board with silicon revision C0 chip. It is not flashed to MMC.
`partition-table.img`	Programed to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for single bootloader condition and for boot storage larger than 13GB.
`partition-table-dual.img`	Programed to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for dual-bootloader condition and for boot storage larger than 13 GB.
`partition-table-28GB.img`	Program to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for single bootloader condition and for boot storage larger than 28 GB.
`partition-table-28GB-dual.img`	Program to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for dual-bootloader condition and for boot storage larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions when doing VTS test with GSI system image.
`vbmeta-imx8qxp.img`	`vbmeta_a` and `vbmeta_b` partitions to support single LVDS-to-HDMI/MIPI-DSI-to-HDMI or dual LVDS-to-HDMI display with dual cameras support.
`vbmeta-imx8qxp-lvds0-panel.img`	`vbmeta_a` and `vbmeta_b` partitions to support lvds panel display.
`vbmeta-imx8qxp-sof.img`	`vbmeta_a` and `vbmeta_b` partitions to support the Sound Open Firmware audio output.
`system.img`	Logical partition `system_a` and logical partition `system_b` in super partition.
`system_ext.img`	Logical partition `system_ext_a` and logical partition `system_ext_b` in super partition.
`vendor.img`	Logical partition `vendor_a` and logical partition `vendor_b` in super partition.
`product.img`	Logical partition `product_a` and logical partition `product_b` in super partition.
`super.img`	Super partition.
`dtbo-imx8qxp.img`	`dtbo_a` and `dtbo_b` partitions to support single LVDS-to-HDMI/MIPI-DSI-to-HDMI or dual LVDS-to-HDMI displays with dual-cameras support.
`dtbo-imx8qxp-lvds0-panel.img`	`dtbo_a` and `dtbo_b` partitions to support LVDS panel display.
`dtbo-imx8qxp-sof.img`	`dtbo_a` and `dtbo_b` partitions to support the Sound Open Firmware audio output.

Table 13. Board images...*continued*

Image name	Download target
`rpmb_key_test.bin`	Prebuilt test RPMB key. It can be used to set the RPMB key to fixed 32 bytes 0x00.
`testkey_public_rsa4096.bin`	Prebuilt AVB public key, which is extracted from the default AVB private key.

9.3 Flashing board images

The board image files can be flashed into the target board using Universal Update Utility (UUU).

For the UUU binary file, download it from GitHub: [uuu release page on GitHub](#).

To achieve more flexibility, two script files are provided to invoke UUU to automatically flash all Android images.

- `uuu_imx_android_flash.sh` for Linux OS
- `uuu_imx_android_flash.bat` for Windows OS

For this release, these two scripts are validated on UUU 1.4.182 version. Download corresponding version from GitHub:

- For Linux OS, download the file named `uuu`.
- For Windows OS, download the file named `uuu.exe`.

Because the two script files will directly invoke UUU, make sure that UUU is in a path contained by the system environment variable of "PATH".

Perform the following steps to download the board images:

1. Download the UUU binary file from GitHub as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   Change the board's SW2 (boot mode) to 1000 (from 1-4 bit) to enter serial download mode.
3. Power on the board. Use the USB cable on the board USB 3.0 Type-C port to connect your PC with the board.
   Note:
   - *There are three USB ports on the 8QuadXPlus MEK board: USB-to-UART, USB 2.0, and USB 3.0.*
   - *The USB-to-UART port can be referenced as debug UART, which can be used to watch the log of the hardware boot processing.*
   - *USB 2.0 is USB Host and USB 3.0 is USB OTG.*
4. Decompress `release_package/android-12.1.0_1.0.0_image_8qmek.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For 8QuadXPlus board, related options are described as follows

Table 14. Options for uuu_imx_android_flash tool

Option	Description
-h	Displays the help information of this tool.
-f soc_name	Specifies SoC information. For i.MX 8QuadXPlus MEK, it should be `imx8qxp`. This option is **mandatory**.
-a	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.
-b	Only flashes slot b. If this option and `-a` option are not used, slots a and b are both flashed.
-c card_size	Specifies which partition table image file to flash. For i.MX 8Quad XPlus MEK, it can be followed with "28". If this option is not used, default `partition-table.img` is flashed.
-u uboot_feature	Flashes U-Boot or SPL&bootloader image with `uboot_feature` in their names. For i.MX 8QuadXPlus MEK, it can be `dual`, `trusty-dual`, `trusty-secure-unlock-dual`, `c0`, `c0-dual` or `trusty-c0-dual`. If this option is not used, the default `u-boot-imx8qxp.imx` is flashed.
-d dtb_feature	Flash dtbo and vbmeta images with `dtb_feature` in their names, for i.MX 8QuadXPlus MEK, it can be `sof`. if it is not used, the default `dtbo-imx8qxp.img` and `vbmeta-imx8qxp.img` will be flashed.
-e	Erases user data after images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in current working directory are flashed.
-t target_dev	Specifies the target device. For i.MX 8QuadXPlus MEK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
-daemon	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
-dryrun	Only generates a UUU script but not executes UUU with this script.
-usb usb_path	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

- On Linux system, open the shell terminal. For example, you can execute a command as follows:

```
> sudo ./uuu_imx_android_flash.sh -f imx8qxp -a -e -u
  trusty-dual
```

- On Windows system, open the command-line interface in administrator mode. The corresponding command is as follows:

```
> uuu_imx_android_flash.bat -f imx8qxp -a -e -u trusty-dual
```

When the command above is executed, `spl-imx8qxp-trusty-dual.bin` is flashed, `bootloader-imx8qxp-trusty-dual.img` with other default images are

flashed into eMMC slot a for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.

Note:

- *`-u` followed with a parameter and containing `trusty` cannot be used together with `-t sd`, because Trusty OS cannot boot from SD card.*
- *To flash the SD card, execute the tool with `-t sd`. To flash eMMC, it does not need to use `-t` option.*
- *If your SD card is 16 GB or the on-board eMMC is used as the boot device, it does not need to use `-c` option.*
- *If your SD card is 32 GB, execute the tool with `-c 28`.*
- *To test dual bootloader, execute the tool with `-u dual`.*
- *To test Trusty OS and dual bootloader both enabled condition, execute the tool with `-u trusty-dual`.*
- *To test the demostration implementation of secure unlock, execute the tool with `-u trusty-secure-unlock-dual`. For secure unlock details, see the i.MX Android Security User's Guide (ASUG).*
- *To test the Sound Open Firmware audio output, execute the tool with `-d sof`.*
- *If `uuu_imx_android_flash.bat` is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:*

```
> net use z: \\192.168.1.1\daily_images
```

"z" in the command represents an available drive letter. It can be other available drive letter.

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, you will get information on the command window indicating that images are already flashed.
   Note:
   If the target device has a DOS partition table on it, the flash process fails for the first time. Push the reset key on the board and execute the flash script again.
7. Power off the board.
8. Change boot device as eMMC or SD card.
   - Change SW2 to switch the board back to 0100 (from 1-4 bit) to enter eMMC boot mode.
   - Change SW2 to switch the board back to 1100 (from 1-4 bit) to enter SD boot mode.

9.4 Booting

After downloading the images, boot the board by connecting it to the power supply.

9.4.1 Booting with single LVDS-to-HDMI/MIPI-DSI-to-HDMI or dual LVDS-to-HDMI displays

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon init=/
init androidboot.fbTileSupport=enable cma=928M@0x960M-0xfc0M
 androidboot.primary_display=imx-drm firmware_class.path=/
vendor/firmware loop.max_part=7 transparent_hugepage=never
 moal.mod_para=wifi_mod_para.conf androidboot.lcd_density=240
 bootconfig
```

AQSUG

User guide

All information provided in this document is subject to legal disclaimers.

Rev. android-12.1.0_1.0.0 — 24 October 2022

© 2022 NXP B.V. All rights reserved.

61 / 65

```
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

9.4.2 Booting with dual LVDS-to-HDMI displays

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon init=/
init cma=928M@0x960M-0xfc0M androidboot.primary_display=imx-
drm firmware_class.path=/vendor/firmware loop.max_part=7
 transparent_hugepage=never moal.mod_para=wifi_mod_para.conf
 androidboot.lcd_density=240 bootconfig
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, append `androidboot.selinux=permissive` to the U-Boot's bootargs. Boot environment variables are as follows:

```
U-Boot > setenv append_bootargs androidboot.selinux=permissive
U-Boot > saveenv
```

9.5 Board reboot

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

10 Revision History

Revision history

Revision number	Date	Substantive changes
P9.0.0_1.0.0-beta	11/2018	Initial release
P9.0.0_1.0.0-ga	01/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.
P9.0.0_2.0.0-ga	04/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.
P9.0.0_2.0.0-ga	08/2019	Updated the location of the SCFW porting kit.
android-10.0.0_1.0.0	02/2020	i.MX 8M Mini, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.
android-10.0.0_1.0.0	03/2020	Deleted the Android 10 image.
android-10.0.0_2.1.0	04/2020	i.MX 8M Plus Alpha and i.MX 8QuadXPlus Beta release.
android-10.0.0_2.0.0	05/2020	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.

Revision history...*continued*

Revision number	Date	Substantive changes
android-10.0.0_2.3.0	07/2020	i.MX 8M Plus EVK Beta1 release, and all the other i. MX 8 GA release.
android-11.0.0_1.0.0	12/2020	i.MX 8M Plus EVK Beta release, and all the other i. MX 8 GA release.
android-11.0.0_2.0.0	04/2021	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, and i.MX 8M Quad GA release.
android-11.0.0_2.2.0	07/2021	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, and i.MX 8M Quad GA release.
android-11.0.0_2.4.0	10/2021	i.MX 8ULP EVK Alpha release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, and i.MX 8M Quad GA release.
android-11.0.0_2.6.0	01/2022	i.MX 8ULP EVK Beta release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, and i.MX 8M Quad GA release.
android-12.0.0_1.0.0	03/2022	i.MX 8ULP EVK Beta release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, and i.MX 8M Quad GA release.
android-12.0.0_2.0.0	07/2022	i.MX 8ULP EVK Beta release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, and i.MX 8M Quad GA release.
android-12.1.0_1.0.0	10/2022	i.MX 8ULP EVK Beta release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8Quad Max, and i.MX 8QuadXPlus GA release.

AQSUG

User guide

All information provided in this document is subject to legal disclaimers.

Rev. android-12.1.0_1.0.0 — 24 October 2022

© 2022 NXP B.V. All rights reserved.

63 / 65

11 Legal information

11.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

11.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

11.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AQSUG

All information provided in this document is subject to legal disclaimers.

© 2022 NXP B.V. All rights reserved.

User guide

Rev. android-12.1.0_1.0.0 — 24 October 2022

64 / 65

Contents