

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 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 with Wi-Fi expansion card.

Contents

1	Overview.....	1
2	Hardware Requirements.....	1
3	Working with the i.MX 8M Mini EVK Board.....	1
4	Working with the i.MX 8M Nano EVK Board.....	9
5	Working with i.MX 8M Plus EVK Board.....	16
6	Working with the i.MX 8M Quad EVK Board.....	25
7	Working with the i.MX 8QuadMax MEK Board.....	34
8	Working with the i.MX 8QuadXPlus MEK Board.....	41
9	Revision History.....	47



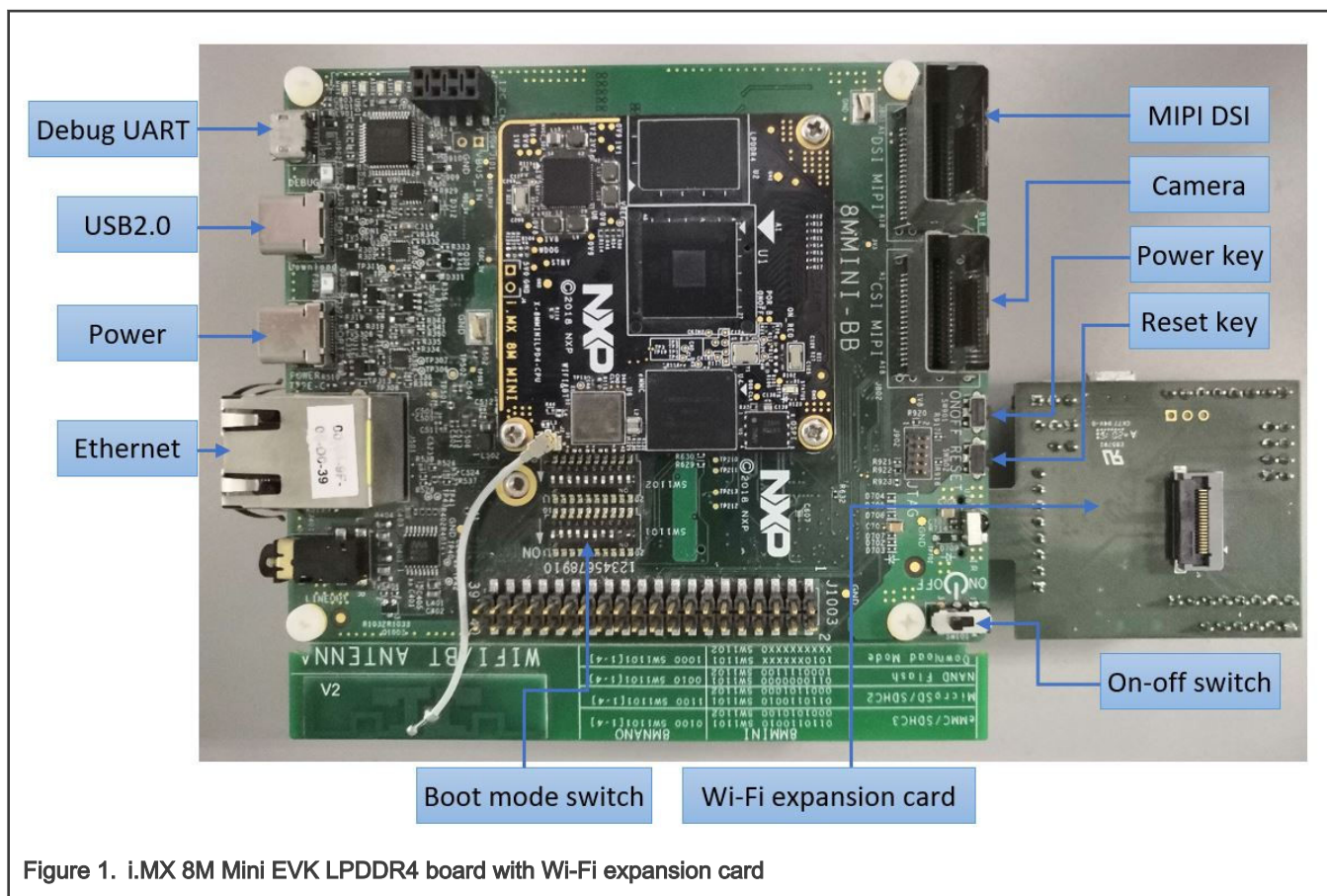


Figure 1. i.MX 8M Mini EVK LPDDR4 board with Wi-Fi expansion card

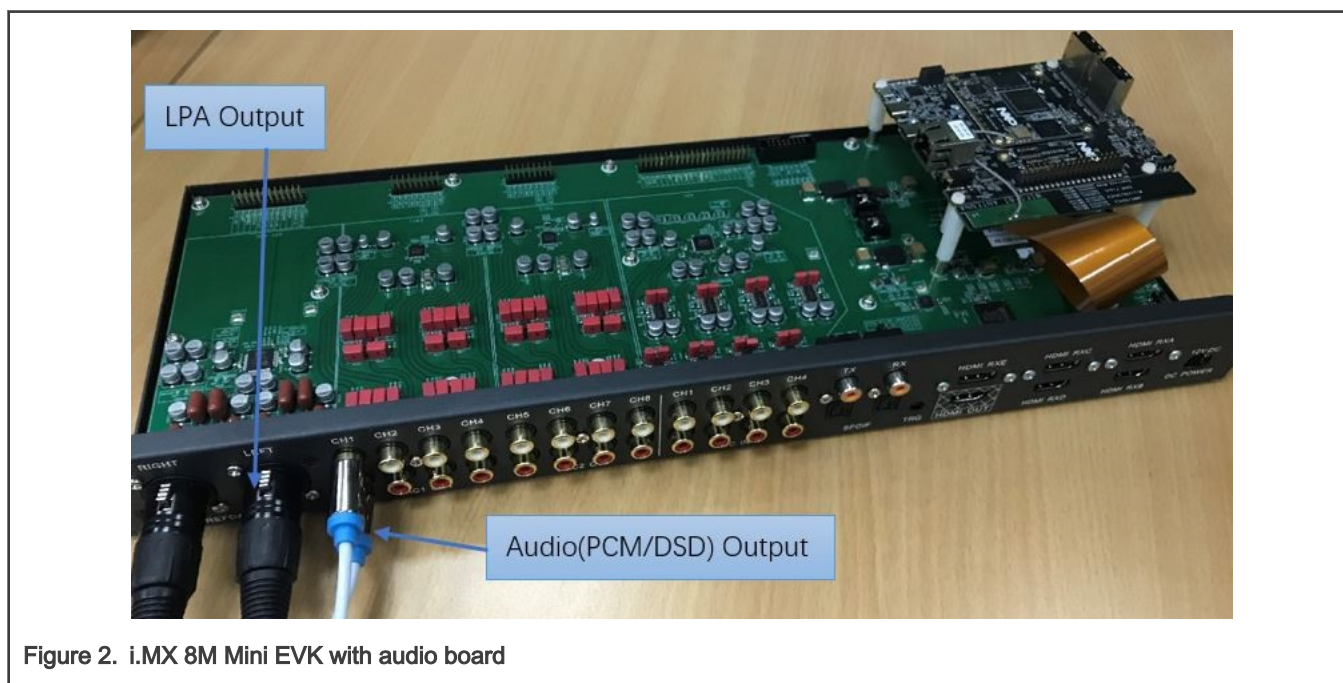


Figure 2. i.MX 8M Mini EVK with audio board



Figure 3. i.MX 8M Mini SAS cable with DSI-to-HDMI adapter



Figure 4. i.MX MIPI panel



Figure 5. i.MX 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 i.MX CSI MIPI Camera to the "Camera" port.
- For i.MX 8M Mini EVK LPDDR4 board, BT function is supported. Wi-Fi function is supported with the Wi-Fi expansion card plugged to the SD card slot.
- For i.MX 8M Mini EVK DDR4 board, BT function is not maintained. This board cannot be plugged with a Wi-Fi expansion card, and the Wi-Fi function is not maintained.
- "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-10.0.0_2.3.0_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.
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.
u-boot-imx8mm.imx	33 KB offset of MMC for a board with LPDDR4 on it.
u-boot-imx8mm-trusty.imx	33 KB offset of MMC for a board with LPDDR4 on it.
u-boot-imx8mm-trusty-secure-unlock.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 will not be 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.
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.

Table continues on the next page...

Table 1. Board images (continued)

Image name	Download target
boot.img	boot_a and boot_b partitions.
boot-debug.img	boot_a and 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 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.
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.
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 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.
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 1.3.124 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. B boards, change the first two bits of board's sw1101 to 10 (from 1-2 bit) to 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 OTG port to connect your PC with the board.
4. Decompress release_package/android-10.0.0_2.3.0_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 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, a and b slot 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 Mini EVK, it can be followed with "28". If this option is not used, default "partition-table.img"/"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", "trusty-secure-unlock", and "trusty". If this option is not used, the default "u-boot-imx8mm.img" 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". 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. If this option is not used, images in the current working directory will be 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 will be 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 images will be flashed. The script just load U-Boot to RAM and execute to fastboot mode. This option is used for development.
-dryrun	Only generate a UUU script but not execute UUU with this script.
-super	"super.img" is not generated when flashing images. The one already exists together with other images is used.

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
```

- 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
```

When the command above is executed, u-boot-imx8mm-trusty.imx with other default images will be flashed into eMMC slot a for i.MX 8M Mini 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 Mini EVK LPDDR4 board:
 - To test Trusty OS, execute the tool with "-u trusty".
 - 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 demonstration implementation of secure unlock, execute the tool with "-u trusty-secure-unlock". 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.
 - To test MIPI panel output, execute the tool with "-d mipi-panel".
 - 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".
- uuu_imx_android_flash.bat generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- 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.

- Ipmake is used to generate super.img to be flashed to the super partition when flashing images. Do not delete or rename this tool.
- The tool used to generate super.img on the Windows system needs "libwinpthread-1.dll". If this file cannot be found, "cmd" will tell you, but "powershell" will just terminate the the execution of the script and do nothing. This "libwinpthread-1.dll" is included in Android SDK platform tools. Download the platform tools and set the "PATH" environment variable to contain the directory in which there is this dll file.

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.

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.

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 androidboot.console=ttyMXC1 androidboot.hardware=freescale
cma=800M@0x400M-0xb80M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 androidboot.console=ttyMXC1 androidboot.hardware=freescale
cma=800M@0x400M-0xb80M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wificountrycode=CN androidboot.lcd_density=240
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, "androidboot.selinux=permissive" needs to be appended 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.

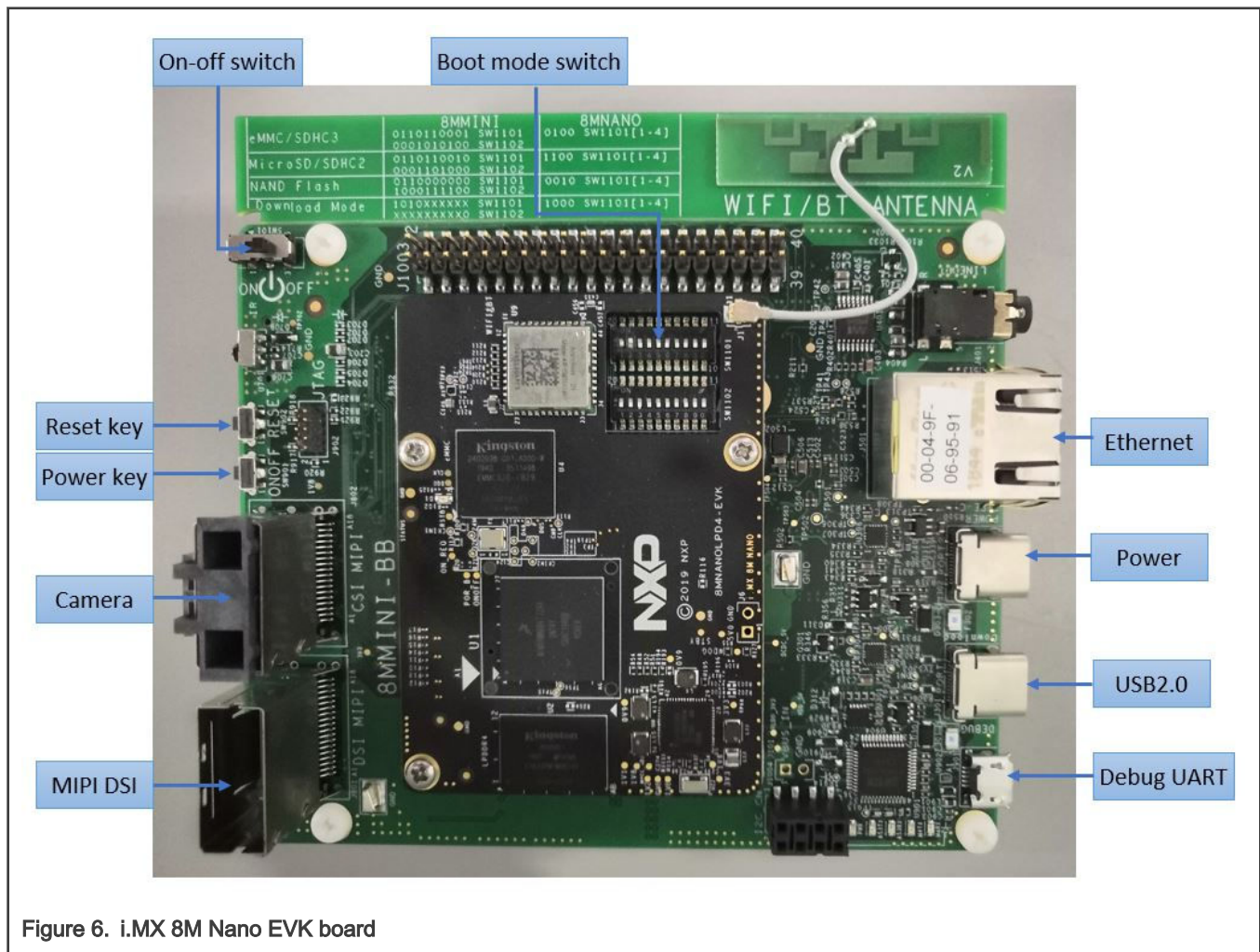


Figure 6. i.MX 8M Nano EVK board



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



Figure 8. i.MX MIPI panel



Figure 9. i.MX 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 i.MX CSI MIPI Camera to the "Camera" 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-10.0.0_2.3.0_image_8mnev.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.
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.
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-trusty.imx	32 KB offset of MMC for i.MX 8M Nano EVK LPDDR4 board.
u-boot-imx8mn-trusty-secure-unlock.imx	32 KB offset of MMC for i.MX 8M Nano EVK LPDDR4 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 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 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.
boot-debug.img	boot_a and 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.

Table continues on the next page...

Table 3. Board images (continued)

Image name	Download target
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 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.
vbmeta-imx8mn-ddr4-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support 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.
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.
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 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 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

For this release, these two scripts are validated on UUU 1.3.124 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-10.0.0_2.3.0_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 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, a and b slot 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"/"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", "trusty-secure-unlock", or "trusty". For i.MX 8M Nano EVK DDR4, it should be "ddr4". If this option is not used, the default "u-boot-imx8mn.img" 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 "rpsmsg" or "mipi-panel". For i.MX 8M Nano EVK DDR4 board, it can be "ddr4", "ddr4-rpsmsg", or "ddr4-mipi-panel". If this option is not used, the default "dtbo-imx8mn.img" and "vbmeta-imx8mn.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. If this option is not used, images in the current working directory will be 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 will be 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 images will be flashed. The script just load U-Boot to RAM and execute to fastboot mode. This option is used for development.
-dryrun	Only generate a UUU script but not execute UUU with this script.

Table continues on the next page...

Table 4. Options for uuu_imx_android_flash tool (continued)

-super	"super.img" is not generated when flashing images. The one already exists together with other images is used.
--------	---------------------------------------------------------------------------------------------------------------

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
```

- 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
```

When the command above is executed, u-boot-imx8mn-trusty.imx with other default images will be 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 Trusty OS, execute the tool with "-u trusty".
 - 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 demonstration implementation of secure unlock, execute the tool with "-u trusty-secure-unlock". 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.
 - To test MIPI-DSI to HDMI output and MCU image, execute the tool with "-d rpmsg".
 - To test MIPI panel output, execute the tool with "-d mipi-panel".
- For i.MX 8M Nano EVK DDR4 board:
 - To test MIPI-DSI to HDMI output, execute the tool with "-d ddr4".
 - To test MIPI-DSI to HDMI output and MCU image, execute the tool with "-d ddr4-rpmsg".
 - To test MIPI panel output, execute the tool with "-d mipi-panel".
- 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.

- Ipmake is used to generate super.img to be flashed to the super partition when flashing images. Do not delete or rename this tool.
- The tool used to generate super.img on the Windows system needs "libwinpthread-1.dll". If this file cannot be found, "cmd" will tell you, but "powershell" will just terminate the the execution of the script and do nothing. This "libwinpthread-1.dll" is included in Android SDK platform tools. Download the platform tools and set the "PATH" environment variable to contain the directory in which there is this dll file.

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.

For Rev. C2 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=ttyMX1,115200
earlycon=ec_imx6q,0x30890000,115200 init=/init androidboot.console=ttyMX1
androidboot.hardware=freescale cma=800M@0x400M-0xb80M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never loop.max_part=7
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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=ttyMX1,115200
earlycon=ec_imx6q,0x30890000,115200 init=/init androidboot.console=ttyMX1
androidboot.hardware=freescale cma=800M@0x400M-0xb80M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never loop.max_part=7
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > setenv bootcmd "bootmcu && boota mmc0" # for SD boot
U-Boot > setenv bootcmd "bootmcu && boota mmc1" # for eMMC boot
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 mmc0" # for SD boot
U-Boot > setenv bootcmd "boota mmc1" # for eMMC boot
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 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.

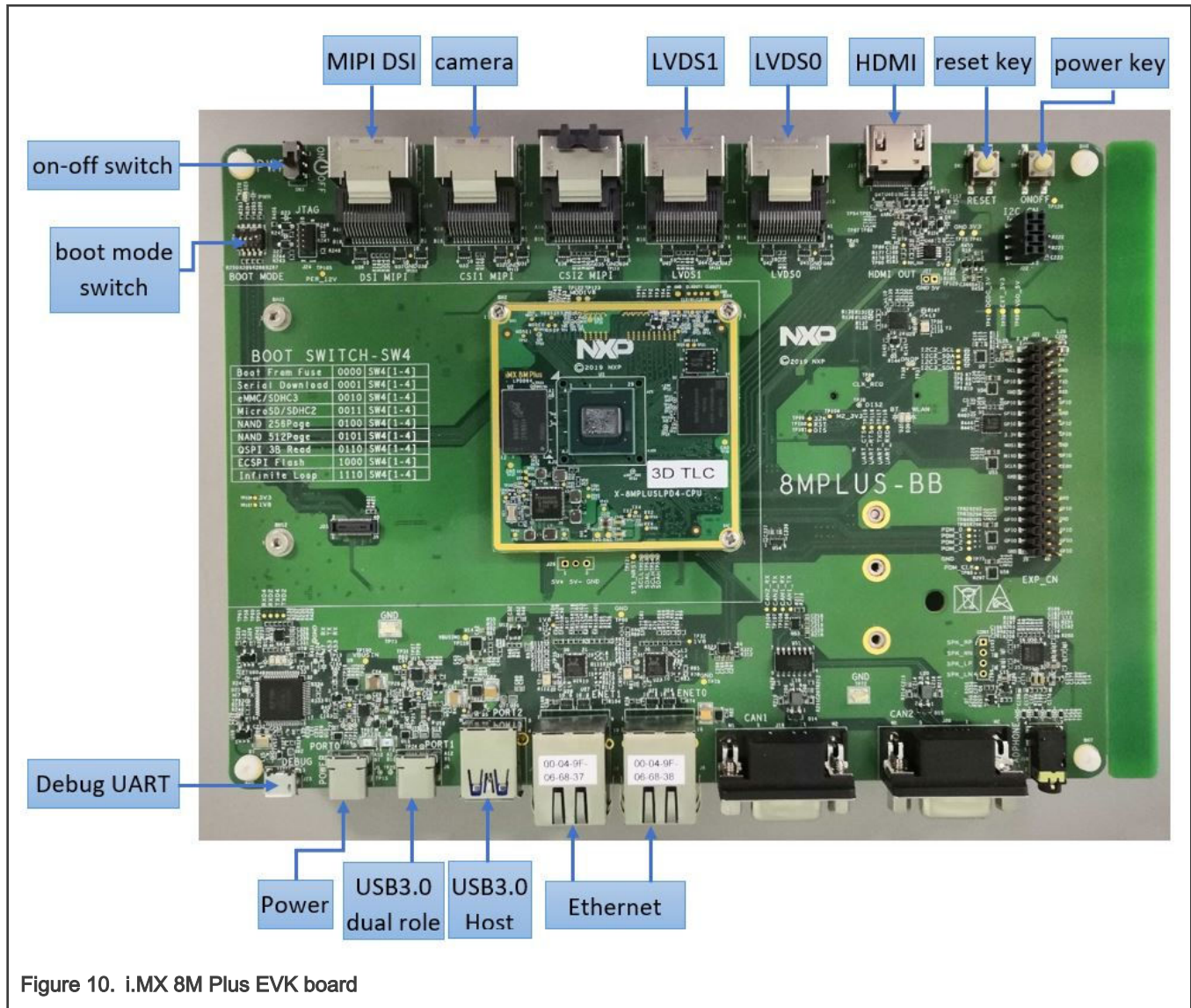


Figure 10. i.MX 8M Plus EVK board



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



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



Figure 13. i.MX MIPI panel

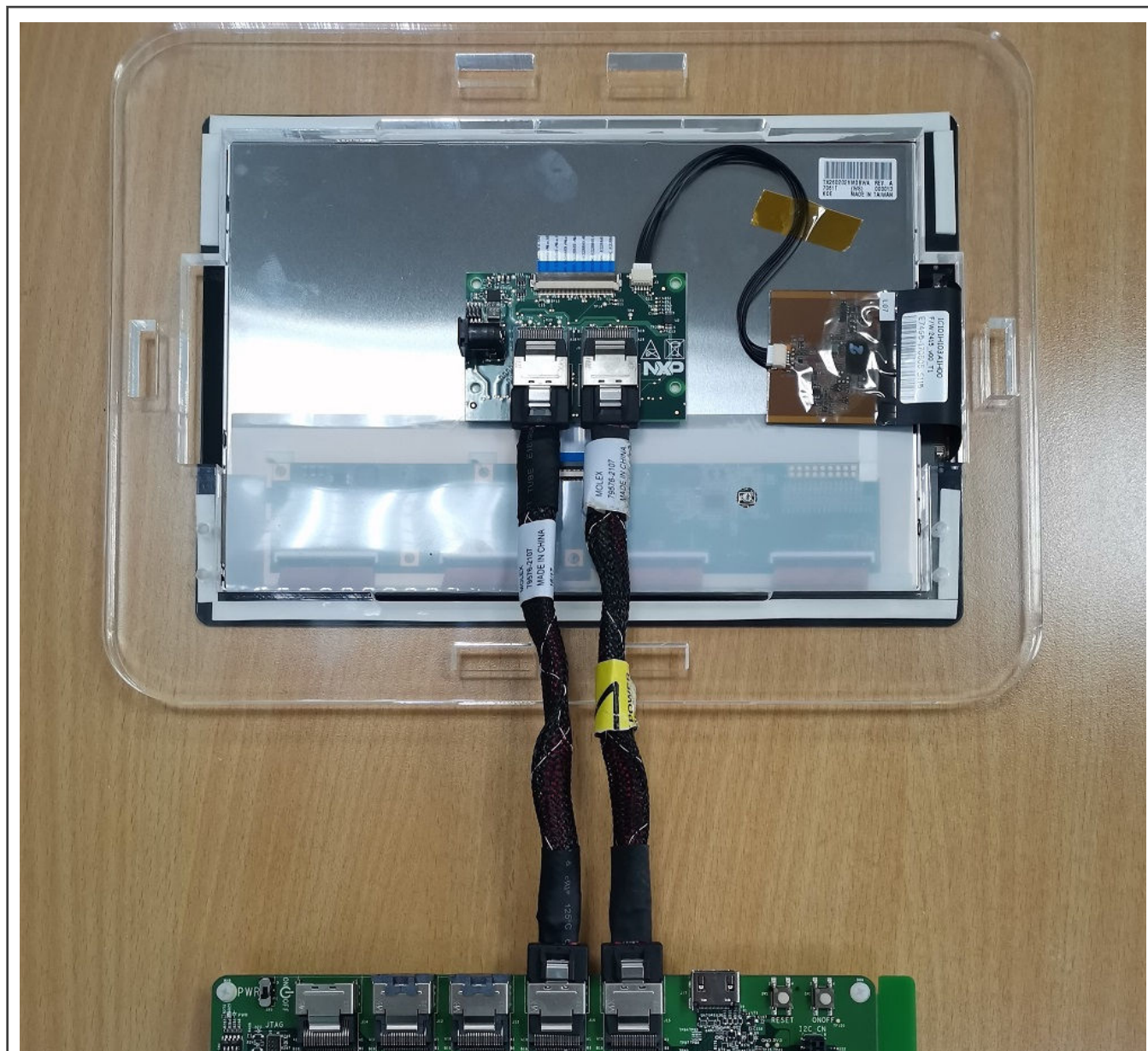


Figure 14. i.MX LVDS panel



Figure 15. i.MX 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 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 the camera, connect the i.MX CSI MIPI Camera to the "Camera" port.

5.2 Board images

The table below describes the location in the board partitions of the software images in android-10.0.0_2.3.0_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.
bootloader-imx8mp-dual.img	bootloader_a and bootloader_b partitions.
bootloader-imx8mp-trusty-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.
u-boot-imx8mp-trusty.imx	0 KB offset of eMMC boot0 partition.
u-boot-imx8mp-trusty-secure-unlock.imx	0 KB offset of eMMC boot0 partition.
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.
boot-debug.img	boot_a and 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.
vbmeta-imx8mp-lvds-panel.img	vbmeta_a and vbmeta_b partitions to support LVDS panel output.
vbmeta-imx8mp-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support MIPI panel output.

Table continues on the next page...

Table 5. Board images (continued)

Image name	Download target
vbmeta-imx8mp-rpmsg.img	vbmeta_a and vbmeta_b partitions to support MIPI-to-HDMI output and MCU image.
system.img	Logical partition system_a and logical partition system_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.
dtbo-imx8mp.img	dtbo_a and dtbo_b partitions to support MIPI-to-HDMI output.
dtbo-imx8mp-lvds-panel.img	dtbo_a and dtbo_b partitions to support LVDS panel output.
dtbo-imx8mp-mipi-panel.img	dtbo_a and dtbo_b partitions to support MIPI panel output.
dtbo-imx8mp-rpmsg.img	dtbo_a and dtbo_b partitions to support MIPI-to-HDMI output and MCU image.
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.

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 1.3.124 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 on the board USB 3.0 port to connect your PC with the board.
4. Decompress release_package/android-10.0.0_2.3.0_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, a and b slot 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 Plus EVK, it can be followed with "28". If this option is not used, default "partition-table.img"/"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 Plus EVK board, it can be "dual", "trusty-dual", "trusty-secure-unlock", or "trusty". 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 "lvds-panel", "lvds", "mipi-panel", "rpmsg". 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. If this option is not used, images in the current working directory will be 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 will be 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 images will be flashed. The script just load U-Boot to RAM and execute to fastboot mode. This option is used for development.
-dryrun	Only generate a UUU script but not execute UUU with this script.
-super	"super.img" is not generated when flashing images. The one already exists together with other images is used.

- 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
```

- 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
```

When the command above is executed, u-boot-imx8mp-trusty.imx with other default images will be flashed into eMMC slot a for i.MX 8M Plus 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 Trusty OS, execute the tool with "-u trusty".
- 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 demonstration implementation of secure unlock, execute the tool with "-u trusty-secure-unlock". 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.
- To test MIPI-DSI to HDMI output and MCU image, execute the tool with "-d rpmsg".
- To test MIPI panel output, execute the tool with "-d mipi-panel".
- 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 the MIPI panel display, execute the tool with "-d mipi-panel". 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.
- uuu_imx_android_flash.bat generates temporary file under the current working directory. Make sure you have write permission under the current working directory.
- 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.
- Ipmake is used to generate super.img to be flashed to the super partition when flashing images. Do not delete or rename this tool.
- The tool used to generate super.img on the Windows system needs "libwinpthread-1.dll". If this file cannot be found, "cmd" will tell you, but "powershell" will just terminate the the execution of the script and do nothing. This "libwinpthread-1.dll" is included in Android SDK platform tools. Download the platform tools and set the "PATH" environment variable to contain the directory in which there is this dll file.

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.
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 display

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

```
U-Boot > setenv bootargs console=ttyMX1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMX1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240 swiotlb=65536
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 display: MIPI-to-HDMI

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

```
U-Boot > setenv bootargs console=ttyMX1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMX1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240 swiotlb=65536
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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: MIPI panel

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

```
U-Boot > setenv bootargs console=ttyMX1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMX1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240 swiotlb=65536
U-Boot > saveenv
```

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

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

5.4.4 Booting with single display: LVDS panel

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

```
U-Boot > setenv bootargs console=ttyMX1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMX1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
```

```
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240 swiotlb=65536
U-Boot > saveenv
```

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

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

5.4.5 Booting with multiple displays

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

```
U-Boot > setenv bootargs console=ttyMX1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMX1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240 swiotlb=65536
U-Boot > saveenv
```

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

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

5.4.6 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=ttyMX1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMX1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240 swiotlb=65536
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, "androidboot.selinux=permissive" needs to be appended to the U-Boot's bootargs. Boot environment variables are as follows:

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

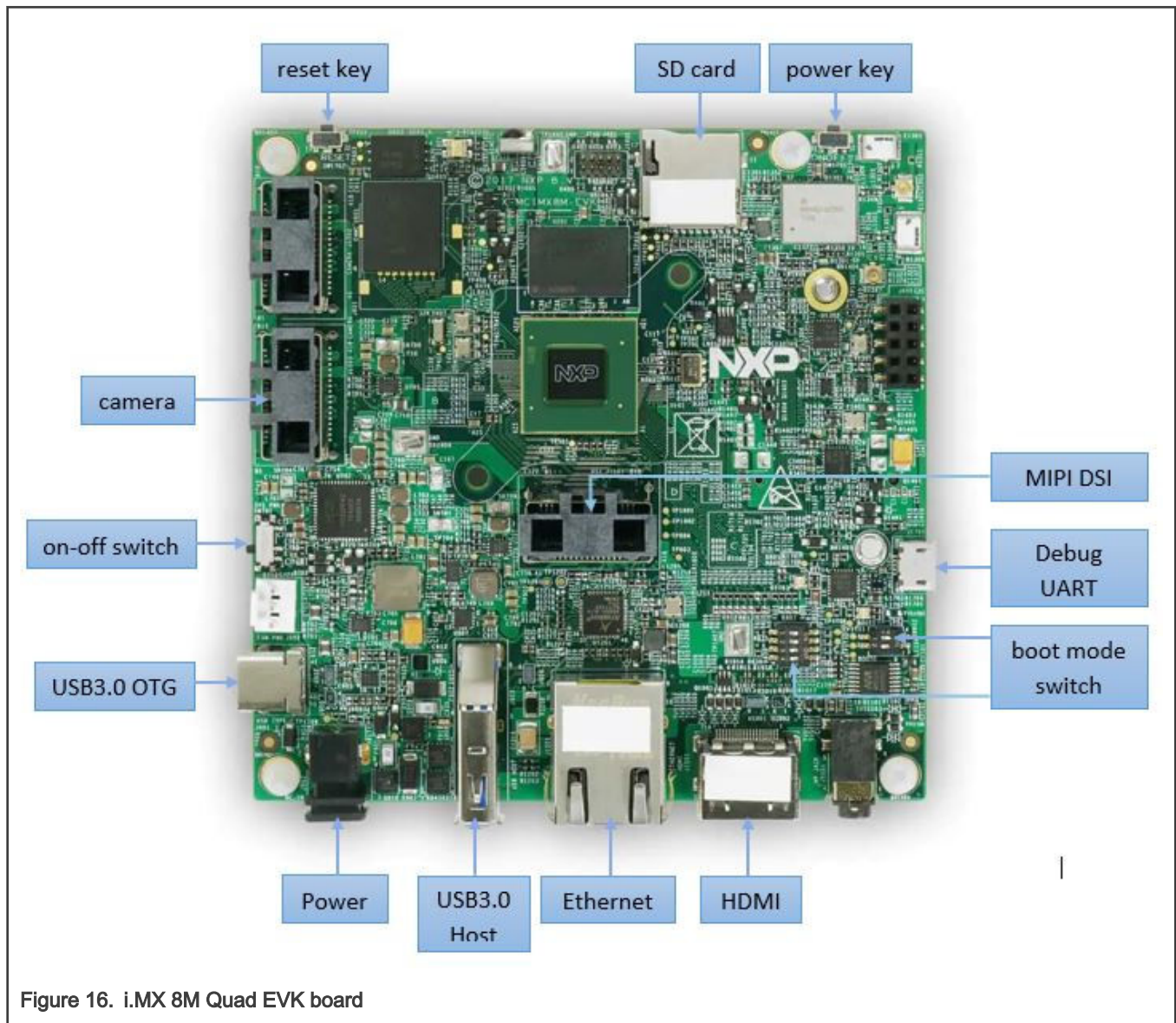
5.5 Board reboot

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.



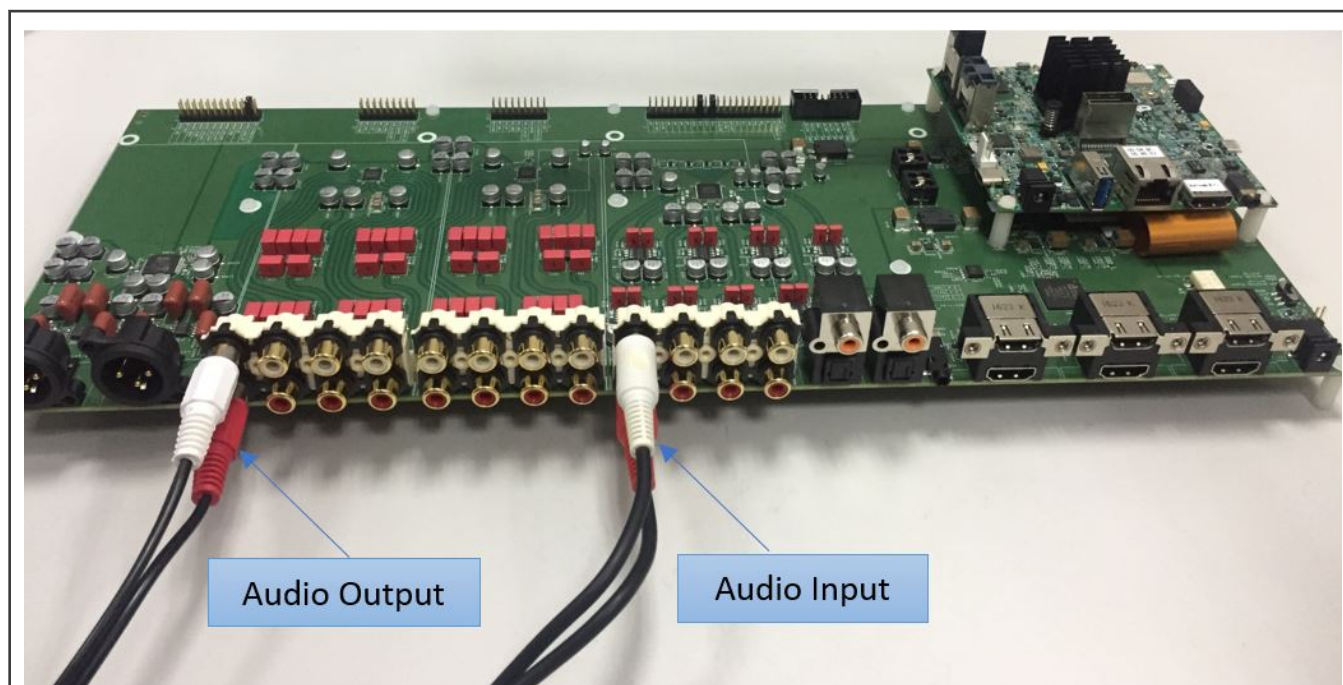


Figure 17. i.MX 8M Quad EVK with audio board



Figure 18. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 19. i.MX MIPI panel



Figure 20. i.MX 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 i.MX CSI MIPI Camera to the "Camera" port.
- Connect the NXP 88W8997 Wi-Fi&BT M.2 expansion card to the J1401 connector to enable Wi-Fi and Bluetooth to work.

6.2 Board images

The table below describes the location in the board partitions of the software images in `android-10.0.0_2.3.0_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.
bootloader-imx8mq-dual.img	bootloader_a and bootloader_b partitions.
bootloader-imx8mq-trusty-dual.img	bootloader_a and bootloader_b partitions.
u-boot-imx8mq.imx	33 KB offset of MMC.
u-boot-imx8mq-trusty.imx	33 KB offset of MMC.
u-boot-imx8mq-trusty-secure-unlock.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 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 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.
boot-debug.img	boot_a and 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 MIPI panel output.
system.img	Logical partition system_a and logical partition system_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.
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 MIPI panel output.

Table continues on the next page...

Table 7. Board images (continued)

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 1.3.124 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.

4. Decompress release_package/android-10.0.0_2.3.0_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 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.

Table continues on the next page...

Table 8. Options for uuu_imx_android_flash tool (continued)

-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"/"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", and "trusty". If this option is not used, default "u-boot-imx8mq.img" 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". 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. 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.
-super	"super.img" is not generated when flashing images. The one already exists together with other images is used.

- 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
```

- 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
```

When the command above is executed, u-boot-imx8mq-trusty.img with other default images are flashed into the eMMC slot a for i.MX 8M Quad 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 Trusty OS, execute the tool with "-u trusty".
- 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 demonstration implementation of secure unlock, execute the tool with "-u trusty-secure-unlock". 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 MIPI panel output, execute the tool with "-d mipi-panel".
 - To test HDMI and MIPI-DSI-to-HDMI dual output, execute the tool with "-d dual".
- uuu_imx_android_flash.bat generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- 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.

- lpmake is used to generate super.img to be flashed to the super partition when flashing images. Do not delete or rename this tool.
- The tool used to generate super.img on the Windows system needs "libwinpthread-1.dll". If this file cannot be found, "cmd" will tell you, but "powershell" will just terminate the the execution of the script and do nothing. This "libwinpthread-1.dll" is included in Android SDK platform tools. Download the platform tools and set the "PATH" environment variable to contain the directory in which there is this dll file.

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 androidboot.console=ttyMXC0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=imx-dcss
firmware_class.path=/vendor/firmware transparent_hugepage=never androidboot.fbTileSupport=enable
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 androidboot.console=ttyMXC0 androidboot.hardware=freescale
cma=1280M androidboot.primary_display=mxsfb-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.displaymode=720p androidboot.wificountrycode=CN
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 androidboot.console=ttyMXC0 androidboot.hardware=freescale
cma=1280M androidboot.primary_display=imx-dcss firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 androidboot.console=ttyMXC0 androidboot.hardware=freescale
cma=1280M androidboot.primary_display=imx-dcss firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 8QuadMax MEK Board

7.1 Board hardware

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

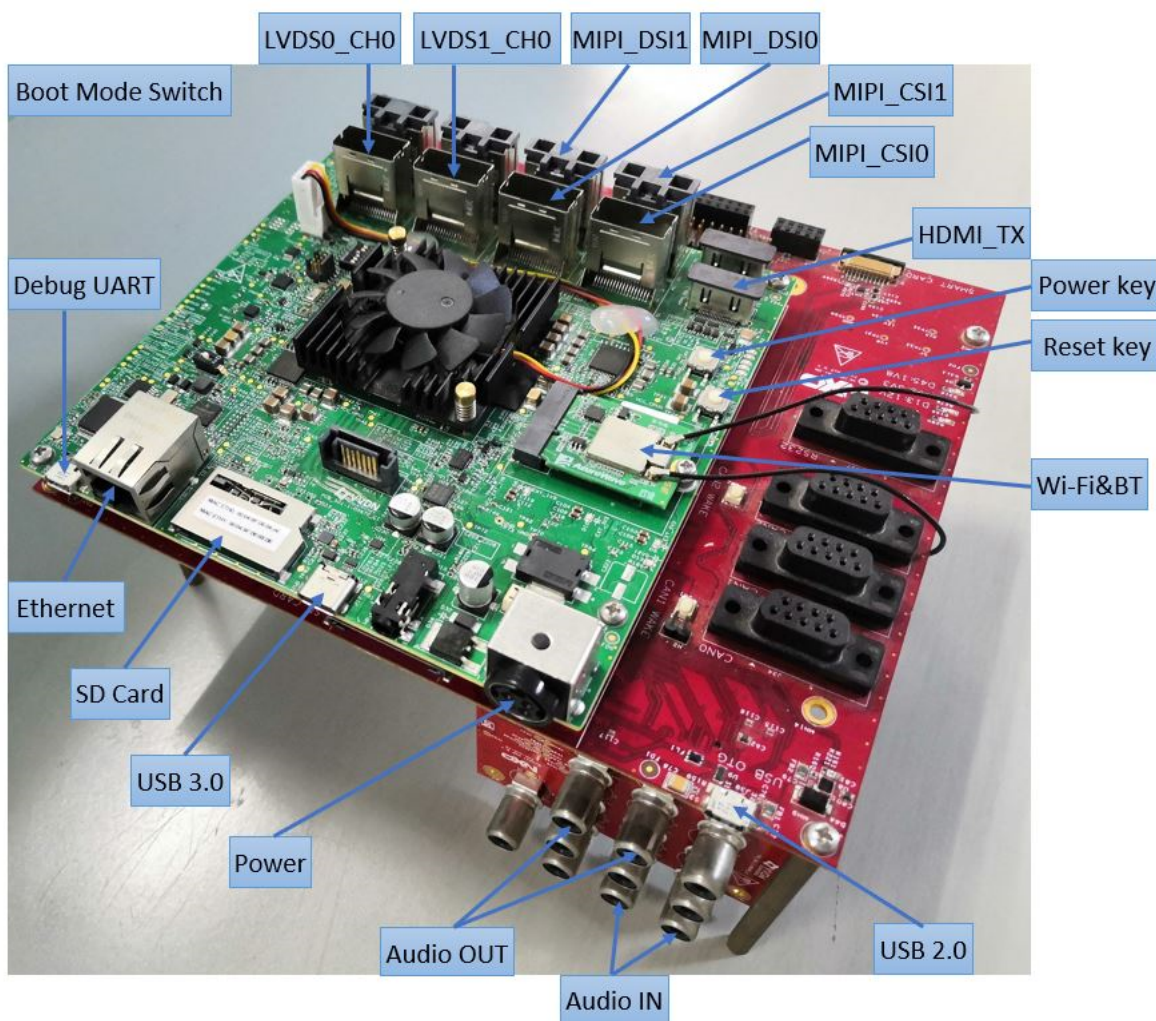


Figure 21. i.MX 8QuadMax MEK board



Figure 22. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 23. i.MX mini SAS cable with LVDS-to-HDMI adapter



Figure 24. i.MX MIPI panel



Figure 25. 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 i.MX 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.
- To test single camera, connect i.MX 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.

7.2 Board images

The table below describes the location in the board partitions of the software images in android-10.0.0_2.3.0_image_8qmek.tar.gz.

Table 9. Board images

Image name	Download target
u-boot-imx8qm.imx	0 KB offset of eMMC and 32 KB offset of SD card.
u-boot-imx8qm-trusty.imx	0 KB offset of eMMC.
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-trusty-secure-unlock.imx	0 KB offset of eMMC.
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 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 boot storage larger than 28 GB.
boot.img	boot_a and boot_b partitions.
boot-debug.img	boot_a and 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 partitions 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 MIPI panel display.
system.img	Logical partition system_a and logical partition system_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.
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 MIPI panel display.
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.

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: [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 1.3.124 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 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-10.0.0_2.3.0_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 10. 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 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 8QuadMax, 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", "trusty", or "trusty-secure-unlock". If this option is not used, the default "u-boot-imx8qm.img" 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" and "mipi-panel". 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. If this option is not used, images in current working directory will be flashed.

Table continues on the next page...

Table 10. Options for uuu_imx_android_flash tool (continued)

Option	Description
-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.
-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.
-super	"super.img" is not generated when flashing images. The one already exists together with other images is used.

- 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
```

- 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
```

When the command above is executed, u-boot-imx8qm-trusty.imx 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 Trusty OS, execute the tool with "-u trusty".
- To test the demonstration implementation of secure unlock, execute the tool with "-u trusty-secure-unlock". 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 -d option. Four display ports of LVDS0_CH0, LVDS1_CH0, MIPI_DSI0, and MIPI_DSI1 can be used for display individually or together.
- To test MIPI panel output, execute the tool with "-d mipi-panel". MIPI_DSI0 port is used for display.
- To test physical HDMI display, execute the tool with "-d hdm". HDMI_RX port is used for display.
- To test multiple displays, execute the tool with "-u md -d md". Four display ports can be used for display:
 - If HDMI_RX is used, the other three ports are LVDS0_CH0, LVDS1_CH0, and MIPI_DSI1.
 - If HDMI_RX is not used, the four ports are LVDS0_CH0, LVDS1_CH0, MIPI_DSI0, and MIPI_DSI1.
- uuu_imx_android_flash.bat generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- 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.

- lpmake is used to generate super.img to be flashed to the super partition when flashing images. Do not delete or rename this tool.
- The tool used to generate super.img on the Windows system needs "libwinpthread-1.dll". If this file cannot be found, "cmd" will tell you, but "powershell" will just terminate the the execution of the script and do nothing. This "libwinpthread-1.dll" is included in Android SDK platform tools. Download the platform tools and set the "PATH" environment variable to contain the directory in which there is this dll file.

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 000100 (from 1-6 bit) to enter eMMC boot mode.
 - Change SW2 to switch the board back to 001100 (from 1-6 bit) to enter SD boot mode.

7.4 Booting

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

7.4.1 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 androidboot.console=ttyLP0
androidboot.hardware=freescale androidboot.fbTileSupport=enable cma=1184M@0x960M-0xe00M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

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

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

7.4.2 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.console=ttyLP0
androidboot.hardware=freescale androidboot.fbTileSupport=enable cma=1184M@0x960M-0xe00M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

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

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

7.4.3 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 androidboot.console=ttyLP0 androidboot.hardware=freescale cma=1184M@0x960M-0xe00M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 8QuadXPlus MEK Board

8.1 Board hardware

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

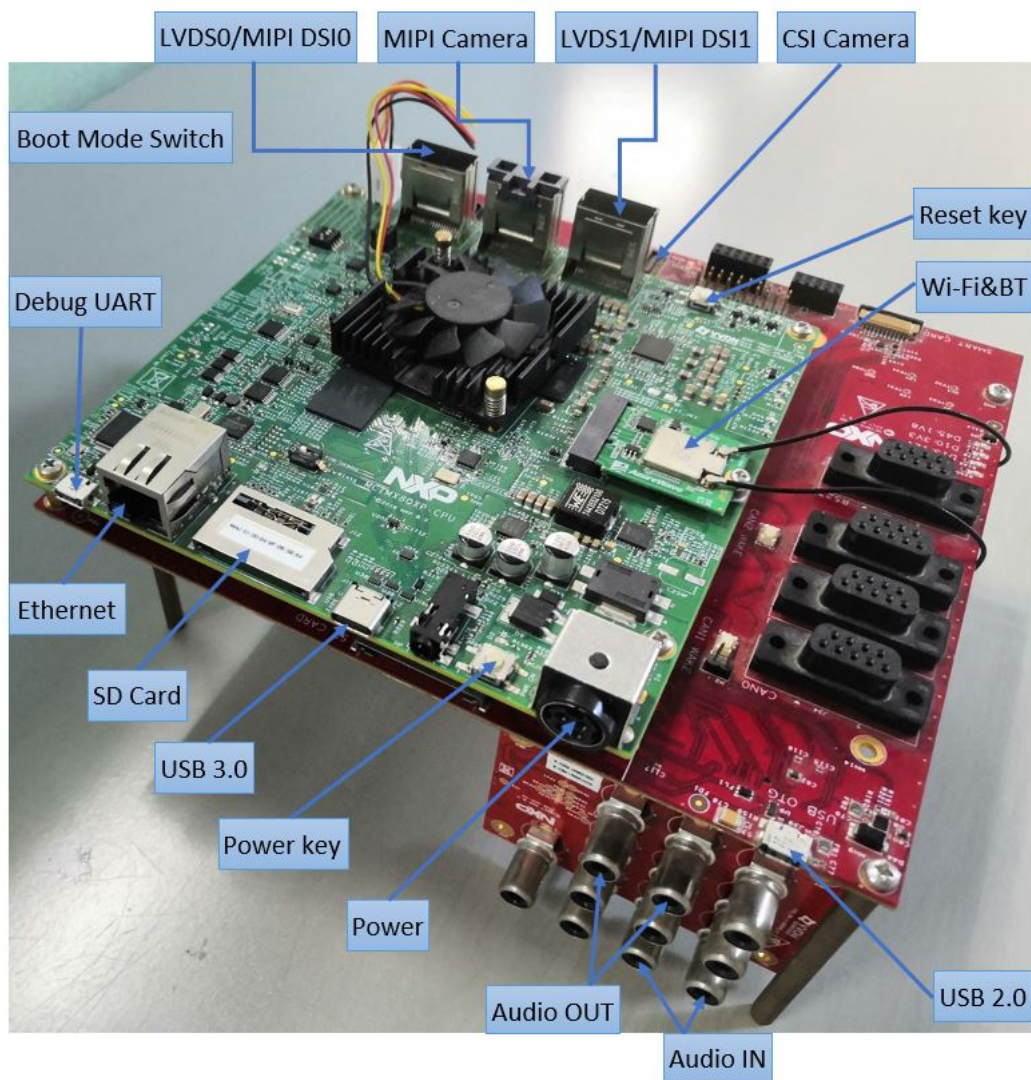


Figure 26. i.MX 8QuadXPlus MEK board



Figure 27. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 28. i.MX mini SAS cable with LVDS-to-HDMI adapter



Figure 29. 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 "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 i.MX MIPI Camera to the "MIPI Camera" port or connect OV5640 Camera to the "CSI Camera" port.

8.2 Board images

The table below describes the location in the board partitions of the software images in `android-10.0.0_2.3.0_image_8qmek.tar.gz`.

Table 11. Board images

Image name	Download target
u-boot-imx8qxp.imx	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
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-trusty.imx	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
u-boot-imx8qxp-trusty-c0.imx	0 KB offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision C0 chip.
u-boot-imx8qxp-trusty-secure-unlock.imx	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision B0 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 boot storage larger than 13 GB.

Table continues on the next page...

Table 11. Board images (continued)

Image name	Download target
partition-table-28GB.img	Program to first 17 KB, and then back up to last 17 KB of the boot storage. GPT image for boot storage larger than 32 GB.
boot.img	boot_a and boot_b partitions.
boot-debug.img	boot_a and boot_b partitions when doing VTS test with GSI system image.
vbmata-imx8qxp.img	vbmata_a and vbmata_b partitions to support single LVDS-to-HDMI/MIPI-DSI-to-HDMI or dual LVDS-to-HDMI display with dual cameras support.
system.img	Logical partition system_a and logical partition system_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.
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.
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 1.3.124 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-10.0.0_2.3.0_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 12. 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 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 8QuadXPlus 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 "trusty", "trusty-secure-unlock", "c0", or "trusty-c0". If this option is not used, the default "u-boot-imx8qxp.imx" is flashed.
-e	Erases user data after images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. If this option is not used, images in current working directory will be 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 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.
-super	"super.img" is not generated when flashing images. The one already exists together with other images is used.

- 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
```

- 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
```

When the command above is executed, u-boot-imx8qxp-trusty.imx 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 Trusty OS, execute the tool with "-u trusty".
- To test the demonstration implementation of secure unlock, execute the tool with "-u trusty-secure-unlock". For secure unlock details, see the *i.MX Android Security User's Guide (ASUG)*.
- `uuu_imx_android_flash.bat` generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- 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.

- `lpmake` is used to generate `super.img` to be flashed to the super partition when flashing images. Do not delete or rename this tool.
- The tool used to generate `super.img` on the Windows system needs "libwinpthread-1.dll". If this file cannot be found, "cmd" will tell you, but "powershell" will just terminate the the execution of the script and do nothing. This "libwinpthread-1.dll" is included in Android SDK platform tools. Download the platform tools and set the "PATH" environment variable to contain the directory in which there is this dll file.

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..

8.4 Booting

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

8.4.1 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.console=ttyLP0
androidboot.hardware=freescale androidboot.fbTileSupport=enable cma=1184M@0x960M-0xe00M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

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

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

8.4.2 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 androidboot.console=ttyLP0 androidboot.hardware=freescale cma=1184M@0x960M-0xe00M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN androidboot.lcd_density=240
U-Boot > saveenv
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended 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 Revision History

Table 13. 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.
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.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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

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

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2018-2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 15 July 2020

Document identifier: AQSUG

