

AQSUG

Android Quick Start Guide

Rev. android-14.0.0_1.0.0 — 6 February 2024

User guide

Document information

Information	Content
Keywords	Android, i.MX, android-14.0.0_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 the 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 the 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
- WEVK 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.

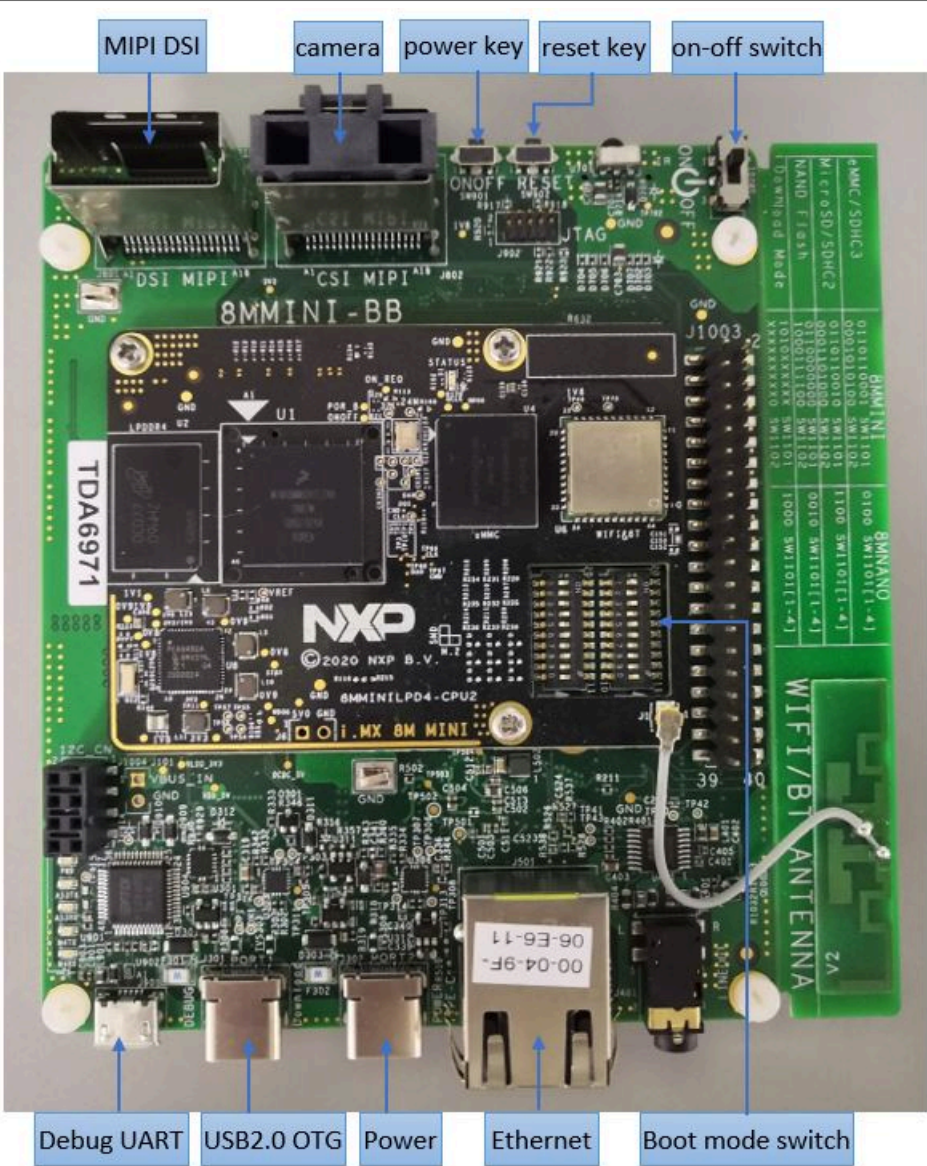


Figure 1. i.MX 8M Mini EVK LPDDR4 board

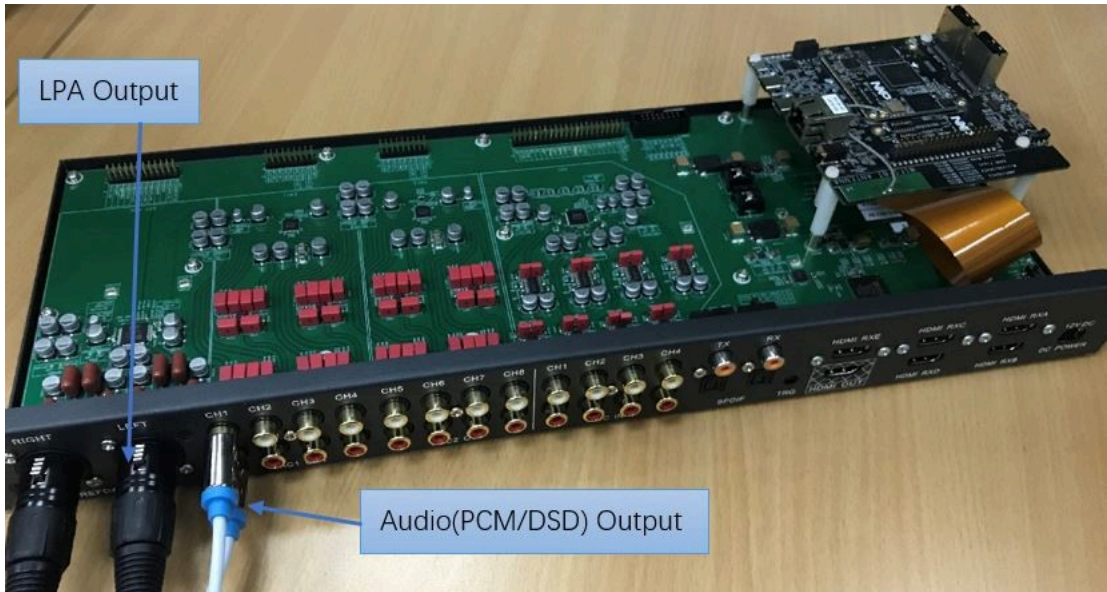


Figure 2. i.MX 8M Mini EVK with audio board



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



Figure 4. MX8-DSI-OLED1 MIPI panel



Figure 5. OV5640 CSI MIPI camera



Figure 6. i.MX 8MIC PDM Microphone board

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.
- To test the i.MX 8MIC PDM microphone, connect the i.MX 8MIC PDM Microphone board to the J1003 connector.
- For i.MX 8M Mini EVK LPDDR4 board, Wi-Fi and Bluetooth functions are supported.
- For i.MX 8M Mini EVK DDR4 board, Wi-Fi and Bluetooth 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-14.0.0_1.0.0_image_8mmevk.tar.gz`.

Table 1. Board images

Image name	Download target
<code>spl-imx8mm-dual.bin</code>	33 KB offset of MMC for i.MX 8M Mini EVK LPDDR4 board.
<code>spl-imx8mm-trusty-dual.bin</code>	33 KB offset of MMC for i.MX 8M Mini EVK LPDDR4 board.
<code>spl-imx8mm-trusty-secure-unlock-dual.bin</code>	33 KB offset of MMC for i.MX 8M Mini EVK LPDDR4 board.
<code>bootloader-imx8mm-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions on the i.MX 8M Mini EVK LPDDR4 board.
<code>bootloader-imx8mm-trusty-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions on the i.MX 8M Mini EVK LPDDR4 board.
<code>bootloader-imx8mm-trusty-secure-unlock-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions on i.MX 8M Mini EVK LPDDR4 board.
<code>u-boot-imx8mm.imx</code>	33 KB offset of MMC for a board with LPDDR4 on it.
<code>u-boot-imx8mm-evk-uuu.imx</code>	The bootloader used by UUU for i.MX 8M Mini board with LPDDR4 on it. It is not flashed to MMC.
<code>u-boot-imx8mm-ddr4.imx</code>	33 KB offset of SD card for a board with DDR4 on it.
<code>u-boot-imx8mm-ddr4-evk-uuu.imx</code>	Bootloader used by UUU for i.MX 8M Mini board with DDR4 on it. It is not flashed to the SD card.
<code>imx8mm_mcu_demo.img</code>	5120 KB offset of MMC.
<code>partition-table.img</code>	0 offset of MMC. It is used for single bootloader condition and the target storage device should be larger than 13 GB.
<code>partition-table-dual.img</code>	0 offset of MMC. It is used for dual bootloader condition and the target storage device should be larger than 13 GB.
<code>partition-table-28GB.img</code>	0 offset of MMC. It is used for single bootloader condition and the target storage device should be larger than 28 GB.
<code>partition-table-28GB-dual.img</code>	0 offset of MMC. It is used for dual bootloader condition and the target storage device should be larger than 28 GB.
<code>boot.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions. This is the AOSP GKI boot image.
<code>boot-imx.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions. This is the boot image built with i.MX kernel tree for debugging.
<code>init_boot.img</code>	<code>init_boot_a</code> and <code>init_boot_b</code> partitions to contain the generic ramdisk on GKI enabled system.
<code>vendor_boot.img</code>	<code>vendor_boot_a</code> and <code>vendor_boot_b</code> partitions.
<code>vendor_boot-debug.img</code>	<code>vendor_boot_a</code> and <code>vendor_boot_b</code> partitions when doing VTS test with GSI system image.
<code>vbmeta-imx8mm.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support MIPI-to-HDMI output on i.MX 8M Mini EVK LPDDR4 board.
<code>vbmeta-imx8mm-m4.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support MIPI-to-HDMI output and audio playback based on Cortex-M4 FreeRTOS on i.MX 8M Mini EVK LPDDR4 board.

Table 1. Board images...continued

Image name	Download target
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-8mic.img	vbmeta_a and vbmeta_b partitions to support 8mic PDM Microphone audio input 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.
vendor_dlkm.img	Logical partition vendor_dlkm_a and logical partition vendor_dlkm_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-8mic.img	dtbo_a and dtbo_b partitions to support 8mic PDM Microphone audio input 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 1.4.182 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-14.0.0_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
<code>-h</code>	Displays the help information of this tool.
<code>-f soc_name</code>	Specifies SoC information. For i.MX 8M Mini EVK, it should be <code>imx8mm</code> . This option is mandatory .
<code>-a</code>	Only flashes slot a. If this option and <code>-b</code> option are not used, slots a and b are both flashed.
<code>-b</code>	Only flashes slot b. If this option and <code>-a</code> option are not used, slots a and b are both flashed.
<code>-c card_size</code>	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 <code>partition-table.img</code> or <code>partition-table-dual.img</code> is flashed.
<code>-m</code>	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&bootloader image with <code>uboot_feature</code> in their names. For i.MX 8M Mini EVK LPDDR4, it can be <code>dual</code> , <code>trusty-dual</code> and <code>"trusty-secure-unlock-dual"</code> . If this option is not used, the default <code>u-boot-imx8mm.imx</code> is flashed. For i.MX 8M Mini EVK DDR4, it should be <code>ddr4</code> .
<code>-d dtb_feature</code>	Flashes DTBO and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8M Mini EVK LPDDR4, it can be <code>m4</code> , <code>mipi-panel</code> , <code>mipi-panel-rm67191</code> and <code>8mic</code> . If this option is not used, default <code>dtbo-imx8mm.img</code> and <code>vbmeta-imx8mm.img</code> are flashed. For i.MX 8M Mini EVK DDR4, it should be <code>ddr4</code> .
<code>-e</code>	Erases user data after images are flashed.
<code>-D directory</code>	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash.bat</code> , it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
<code>-t target_dev</code>	Specifies the target device. For i.MX 8M Mini EVK, it can be <code>emmc</code> and <code>sd</code> . If this option is not used, images are flashed to eMMC.
<code>-daemon</code>	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.

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

Option	Description
<code>-i</code>	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.
<code>-dryrun</code>	Only generates a UUU script but not execute UUU with this script.
<code>-usb usb_path</code>	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 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 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 demonstration 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`.
 - To test support i.MX 8MIC PDM Microphone audio input, execute the tool with `-d 8mic`.
- For i.MX 8M Mini EVK DDR4 board: execute the command with `-u ddr4 -d ddr4 -t sd`.
- For i.MX 8M Mini EVK with audio board: To test low power audio, execute the tool with `-d m4 -m`. See the Android User's Guide (AUG) for more steps to make the low-power audio work.
- 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.

- 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 to eMMC or SD card.
For Rev. C boards:
 - Change SW1101 to 0110110010 and SW1102 to 0001101000 to boot from SD card.
 - Change SW1101 to 0110110001 and SW1102 to 0001010100 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=ttyMX1,115200
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=ttyMX1,115200
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.

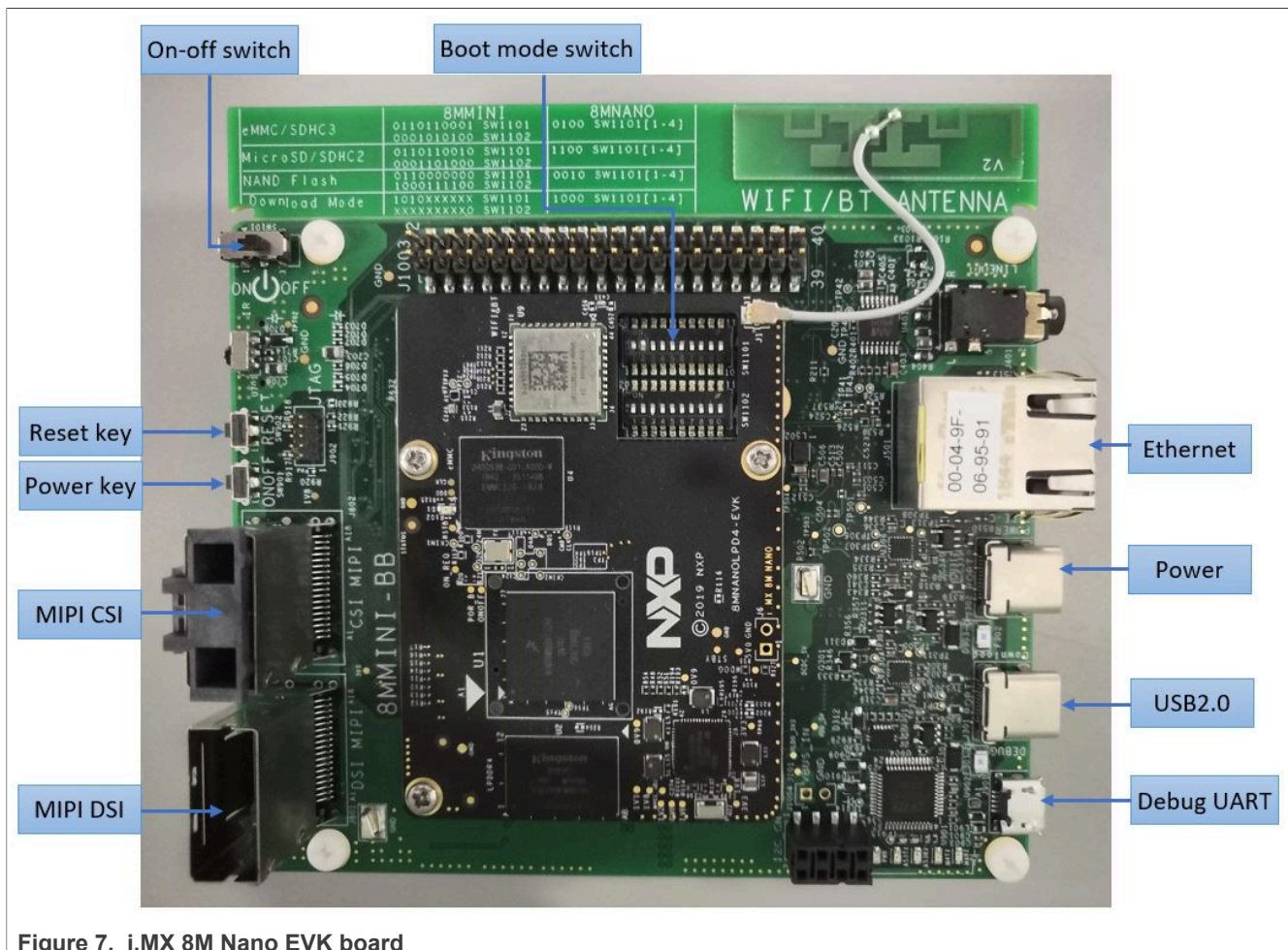


Figure 7. i.MX 8M Nano EVK board



Figure 8. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 9. MX8-DSI-OLED1 MIPI panel



Figure 10. OV5640 CSI MIPI camera



Figure 11. i.MX 8MIC PDM Microphone board

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.
- To test i.MX 8MIC PDM microphone, connect the i.MX 8MIC PDM Microphone board to the J1003 connector.
- 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-14.0.0_1.0.0_image_8mnevkc.tar.gz`.

Table 3. Board images

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

Table 3. Board images...continued

Image name	Download target
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-8mic.img	vbmeta_a and vbmeta_b partitions to support 8mic PDM Microphone audio input 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 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.
vendor_dlkm.img	Logical partition vendor_dlkm_a and logical partition vendor_dlkm_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-8mic.img	dtbo_a and dtbo_b partitions to support 8mic PDM Microphone audio input 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

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-14.0.0_1.0.0_image_8mnevkc.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
<code>-h</code>	Displays the help information of this tool.
<code>-f soc_name</code>	Specifies SoC information. For i.MX 8M Nano EVK, it should be <code>imx8mn</code> . This option is mandatory .
<code>-a</code>	Only flashes slot a. If this option and <code>-b</code> option are not used, slots a and b are both flashed.
<code>-b</code>	Only flashes slot b. If this option and <code>-a</code> option are not used, slots a and b are both flashed.
<code>-c card_size</code>	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 <code>partition-table.img</code> or <code>partition-table-dual.img</code> is flashed.
<code>-m</code>	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&bootloader image with <code>uboot_feature</code> in their names. For i.MX 8M Nano EVK LPDDR4 board, it can be <code>dual</code> , <code>trusty-dual</code> and <code>trusty-secure-unlock-dual</code> . For i.MX 8M Nano EVK DDR4, it should be <code>ddr4</code> . If this option is not used, the default <code>u-boot-imx8mn.img</code> is flashed.
<code>-d dtb_feature</code>	Flashes DTBO and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8M Nano EVK LPDDR4 board, it can be <code>rpmsg</code> , <code>mipi-panel</code> , <code>mipi-panel-rm67191</code> or <code>8mic</code> . For i.MX 8M Nano EVK DDR4 board, it can be <code>ddr4</code> , <code>ddr4-rpmsg</code> , <code>ddr4-</code>

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

Option	Description
	mipi-panel or ddr4-mipi-panel-rm67191. 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. For <code>uuu_imx_android_flash.bat</code> , 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 <code>emmc</code> and <code>sd</code> . 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 demonstration 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`.
 - To test i.MX 8MIC PDM Microphone audio input, execute the tool with `-d 8mic`.
- 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 tool 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.

- Wait for the `uuu_imx_android_flash` execution to complete. If there is no 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.

- Power off the board.
- Change boot device to 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:

```
U-Boot > setenv bootargs console=ttyMXCL,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:

```
U-Boot > setenv bootargs console=ttyMXCL,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.

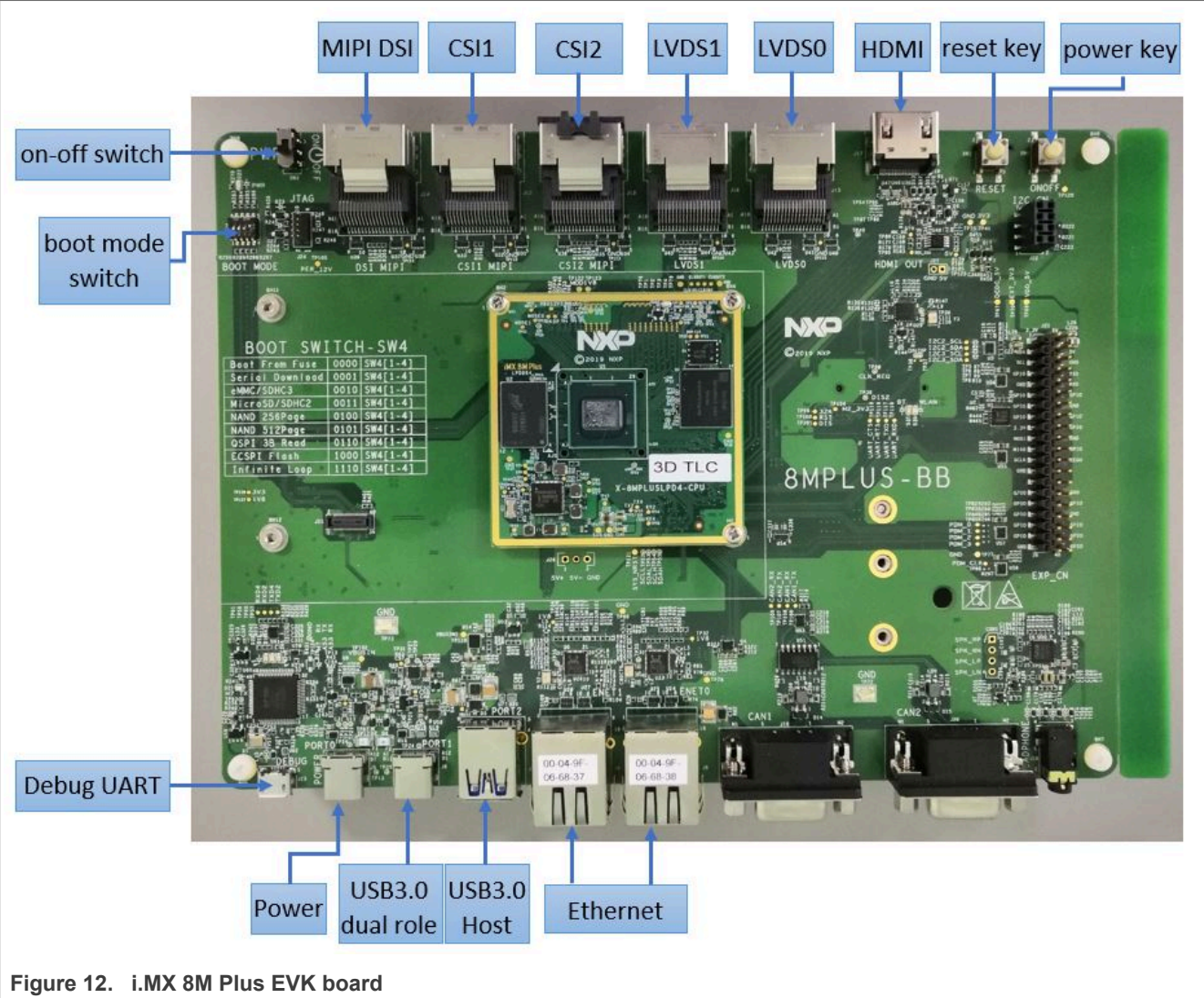


Figure 12. i.MX 8M Plus EVK board

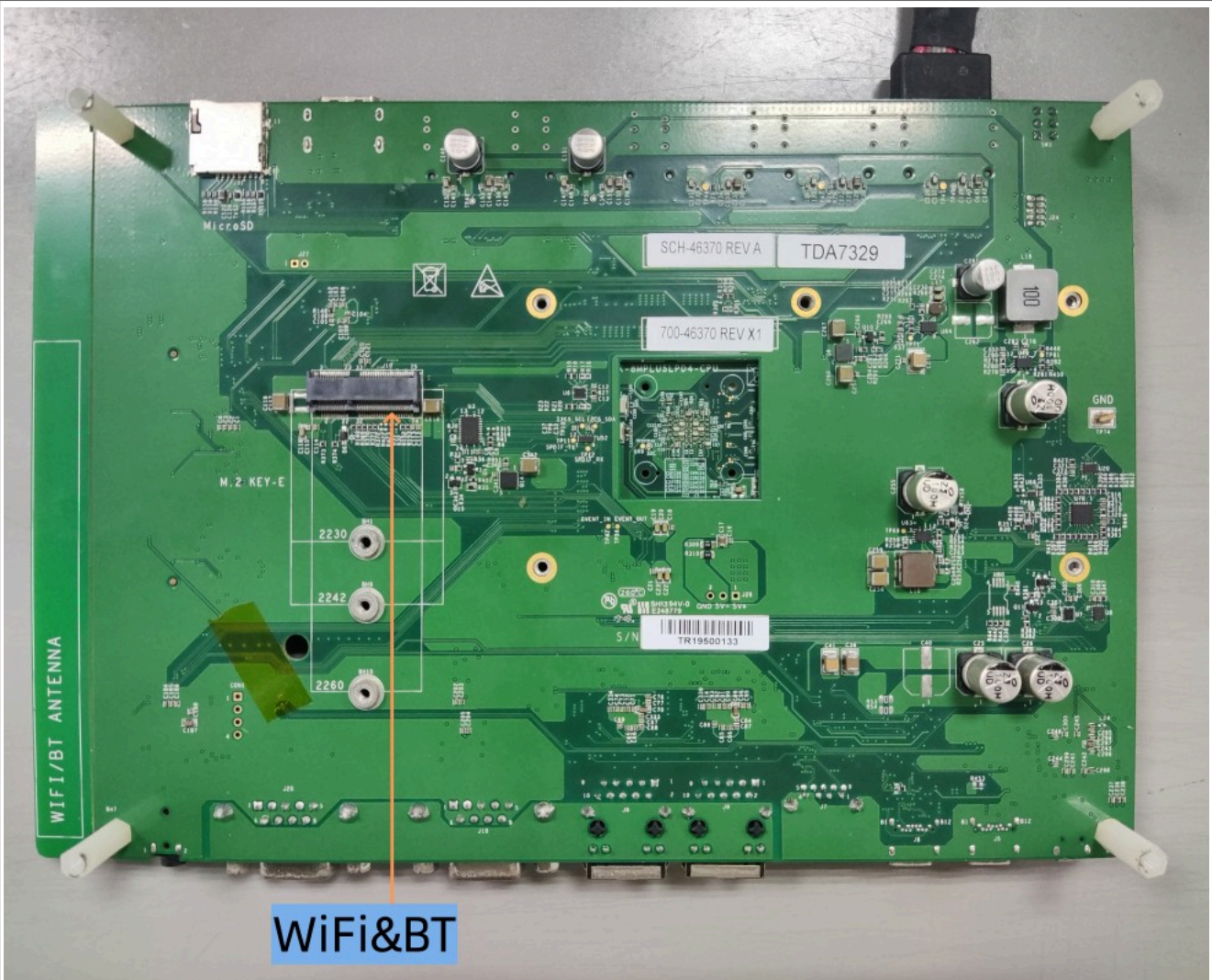


Figure 13. i.MX 8M Plus EVK board back view



Figure 14. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 15. i.MX mini SAS cable with LVDS-to-HDMI adapter



Figure 16. MX8-DSI-OLED1 MIPI panel

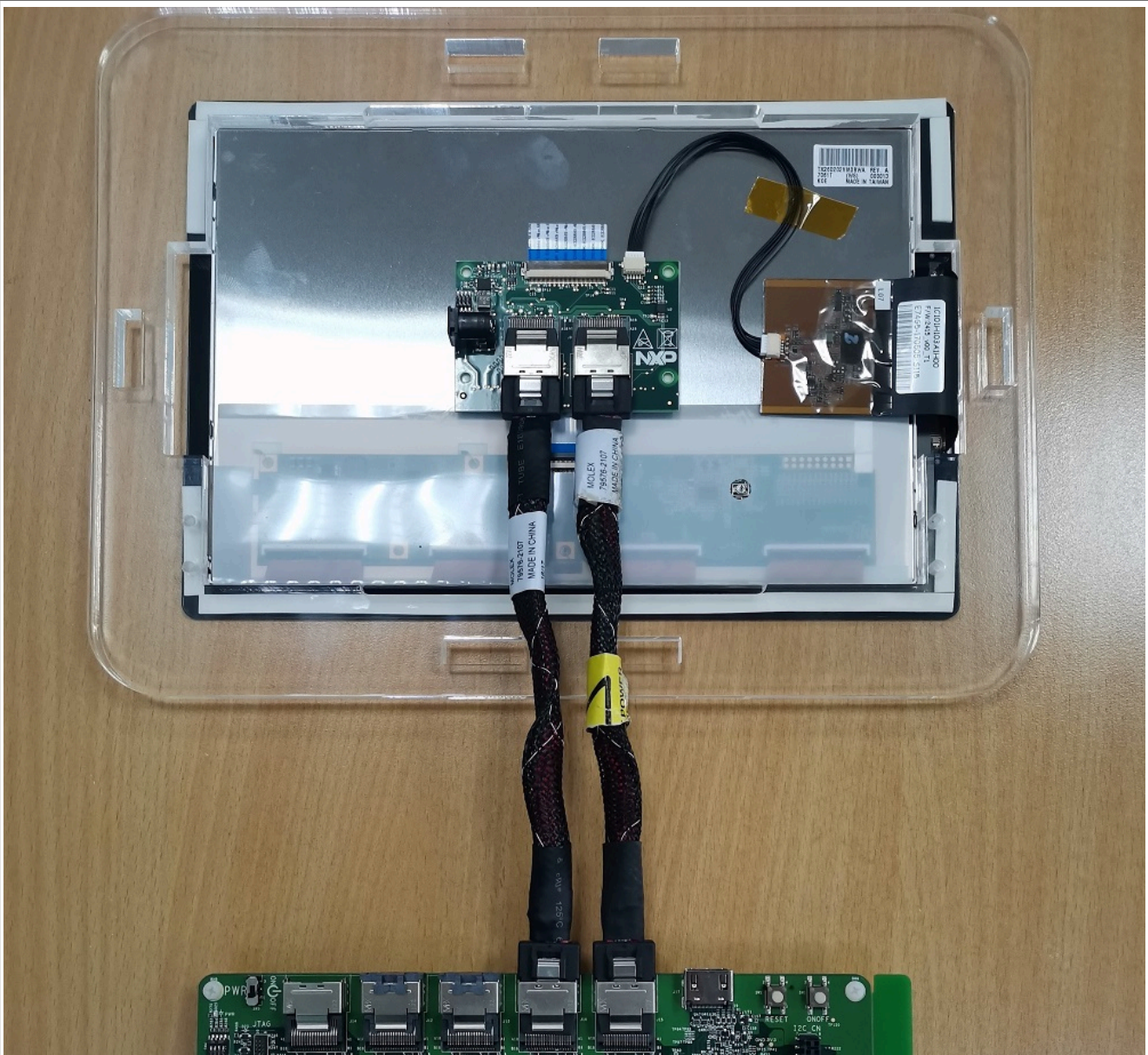


Figure 17. i.MX LVDS panel



Figure 18. i.MX dual-channel LVDS-to-HDMI adapter



Figure 19. OV5640 CSI MIPI camera



Figure 20. Basler CSI MIPI camera



Figure 21. OS08A20 CSI MIPI camera



Figure 22. PCIE8997 (AW-CM276)

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-channel LVDS-to-HDMI adapter to the "LVDS0" and "LVDS1" ports.
- To test the camera, follow the rules below:
 - OS08A20 (CSI1) + OS08A20 (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`.
 - Basler (CSI1) + Basler (CSI2) : Use `dtbo-imx8mp-dual-basler.img + vbmeta-imx8mp-dual-basler.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`.
- Connect the PCIE8997 (AW-CM276) Wi-Fi&BT M.2 expansion card to the J10 connector to enable Wi-Fi and Bluetooth.

5.2 Board images

The table below describes the location in the board partitions of the software images in `android-14.0.0_1.0.0_image_8mpcvk.tar.gz`.

Table 5. Board images

Image name	Download target
<code>spl-imx8mp-dual.bin</code>	0 KB offset of eMMC <code>boot0</code> partition or 32 KB offset of SD card.
<code>spl-imx8mp-trusty-dual.bin</code>	0 KB offset of eMMC <code>boot0</code> partition.
<code>spl-imx8mp-trusty-secure-unlock-dual.bin</code>	0 KB offset of eMMC <code>boot0</code> partition.
<code>bootloader-imx8mp-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions.
<code>bootloader-imx8mp-trusty-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions.
<code>bootloader-imx8mp-trusty-secure-unlock-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions.
<code>u-boot-imx8mp.imx</code>	0 KB offset of eMMC <code>boot0</code> partition or 32 KB offset of SD card.
<code>u-boot-imx8mp-evk-uuu.imx</code>	Bootloader used by UUU for i.MX 8M Plus board. It is not flashed to MMC.
<code>imx8mp_mcu_demo.img</code>	5120 KB offset of eMMC user partition or SD card.
<code>partition-table.img</code>	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.
<code>partition-table-dual.img</code>	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.
<code>partition-table-28GB.img</code>	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.
<code>partition-table-28GB-dual.img</code>	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.
<code>boot.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions. This is the AOSP GKI boot image.
<code>boot-imx.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions. This is the boot image built with i.MX kernel tree for debugging.
<code>init_boot.img</code>	<code>init_boot_a</code> and <code>init_boot_b</code> partitions to contain the generic ramdisk on GKI enabled system.
<code>vendor_boot.img</code>	<code>vendor_boot_a</code> and <code>vendor_boot_b</code> partitions.
<code>vendor_boot-debug.img</code>	<code>vendor_boot_a</code> and <code>vendor_boot_b</code> partitions when doing VTS test with GSI system image.
<code>vbmeta-imx8mp.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support MIPI-to-HDMI output, and support two OS08A20 cameras plug-in CSI1 and CSI2 ports.
<code>vbmeta-imx8mp-basler-ov5640.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support MIPI-to-HDMI output, and support Basler camera plug-in CSI1 port and OV5640 camera plug-in CSI2 port.
<code>vbmeta-imx8mp-dual-basler.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support MIPI-to-HDMI output, and support Basler + Basler camera.
<code>vbmeta-imx8mp-os08a20-ov5640.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support MIPI-to-HDMI output, and support OS08A20 (CSI1) + OV5640 (CSI2) camera.
<code>vbmeta-imx8mp-os08a20.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support MIPI-to-HDMI output, and support only OS08A20 camera plug-in CSI1 slot.

Table 5. Board images...continued

Image name	Download target
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.
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.
vendor_dlkm.img	Logical partition vendor_dlkm_a and logical partition vendor_dlkm_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 OS08A20 cameras plug-in CSI1 and CSI2 ports.
dtbo-imx8mp-basler-ov5640.img	dtbo_a and dtbo_b partitions to support MIPI-to-HDMI output, and support 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-basler.img	dtbo_a and dtbo_b partitions to support MIPI-to-HDMI output, and support Basler + Basler camera.
dtbo-imx8mp-os08a20-ov5640.img	dtbo_a and dtbo_b partitions to support MIPI-to-HDMI output, and support 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.

Table 5. Board images...continued

Image name	Download target
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.
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-14.0.0_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 <code>imx8mp</code> . 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
<code>-c card_size</code>	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 <code>partition-table.img</code> or <code>partition-table-dual.img</code> is flashed.
<code>-m</code>	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&Bootloader image with <code>uboot_feature</code> in their names. For i.MX 8M Plus EVK board, it can be <code>dual</code> , <code>trusty-dual</code> , <code>trusty-secure-unlock-dual</code> . If this option is not used, the default <code>u-boot-imx8mp.img</code> is flashed.
<code>-d dtb_feature</code>	Flashes DTBO and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8M Plus EVK with Non-Rev. B4 BB, it can be <code>basler-ov5640</code> , <code>basler</code> , <code>ov5640</code> , <code>lvds-panel</code> , <code>lvds</code> , <code>mipi-panel</code> , <code>rpmsg</code> , <code>dual-basler</code> , <code>os08a20-ov5640</code> , <code>os08a20</code> , <code>mipi-panel-rm67191</code> , or <code>sof</code> . If this option is not used, the default <code>dtbo-imx8mp.img</code> and <code>vbmeta-imx8mp.img</code> are flashed.
<code>-e</code>	Erases the user data after images are flashed.
<code>-D directory</code>	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash.bat</code> , it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
<code>-t target_dev</code>	Specifies the target device. For i.MX 8M Plus EVK, it can be <code>emmc</code> and <code>sd</code> . If this option is not used, images are flashed to eMMC.
<code>-daemon</code>	Run UUU in daemon mode. This option is used to flash multiple boards of the same type.
<code>-i</code>	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.
<code>-dryrun</code>	Only generates a UUU script but not execute UUU with this script.
<code>-usb usb_path</code>	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

Note: `-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 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 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 demonstration 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 MIPI-to-HDMI display, it does not need to use `-d` option. Physical HDMI, LVDS-to-HDMI are also supported in this condition.
 - To test 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 OS08A20 cameras, it does not need to use `-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`. See the Android User's Guide (AUG) for more steps to make cameras work.
 - To test single Basler camera, execute the tool with `-d basler`. See the Android User's Guide (AUG) for more steps to make the camera work.
 - To test single OV5640 camera, execute the tool with `-d ov5640`. See the Android User's Guide (AUG) for more steps to make the camera work.
 - To test the combination of dual Basler cameras, execute the tool with `-d dual-basler`. 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`. See the Android User's Guide (AUG) for more steps to make the camera work.
 - To test single OS08A20 camera, execute the tool with `-d os08a20`. See the 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`.
 - To test low-power audio, execute the tool with `-d rpmsg -m`. See the Android User's Guide (AUG) for more steps to make the low-power audio work.
- If `uuu_imx_android_flash.bat` is used to flash images on a remote server through Samba, 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.

- Wait for the `uuu_imx_android_flash` execution to complete. If there is no error, the command window displays the information indicating that images are already flashed.
- Power off the board.
- Change boot device to 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=ttyMXCL,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=ttyMXCL,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
```

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=ttyMXCL,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
```

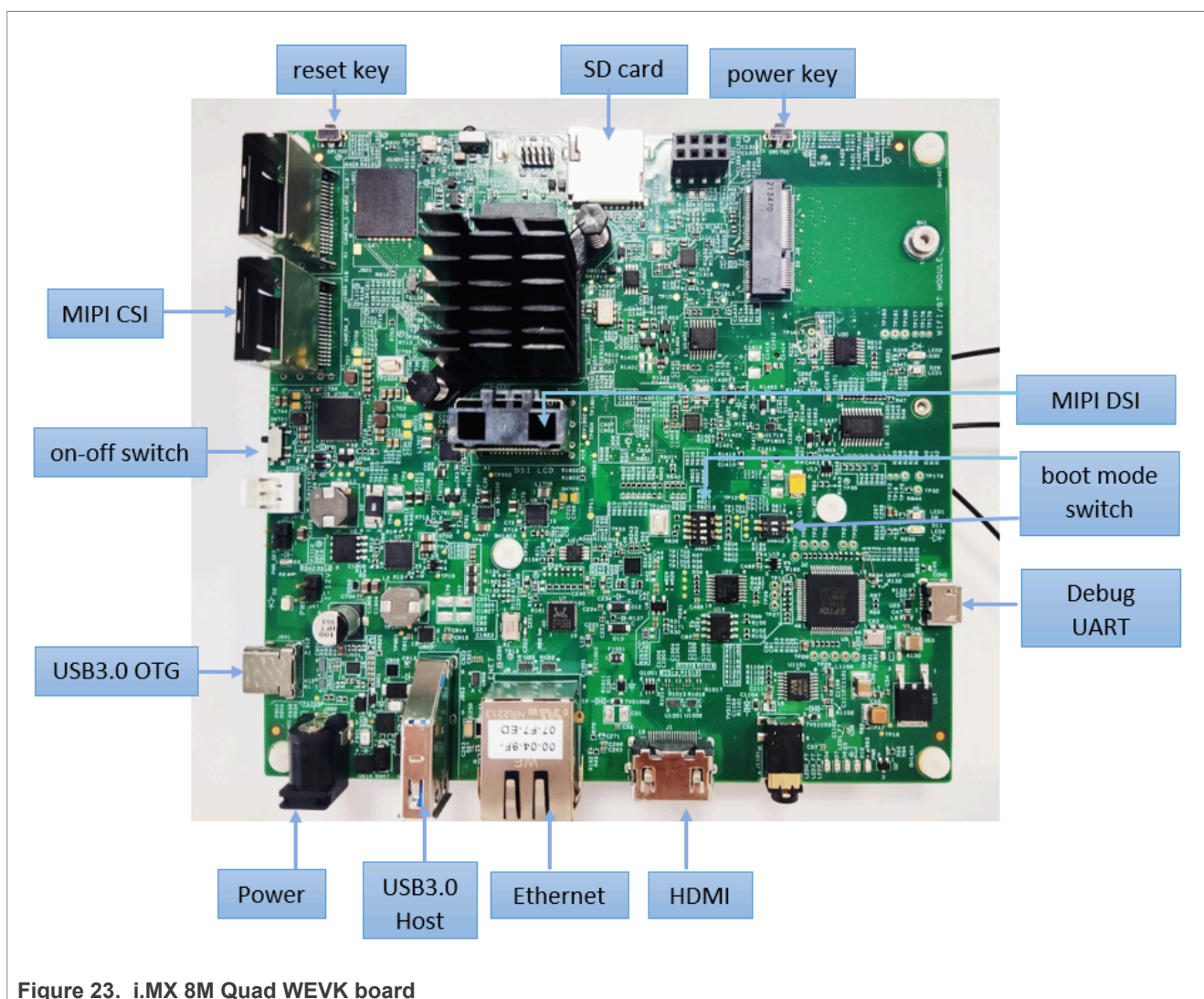
5.5 Board reboot

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 WEVK Board

6.1 Board hardware

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



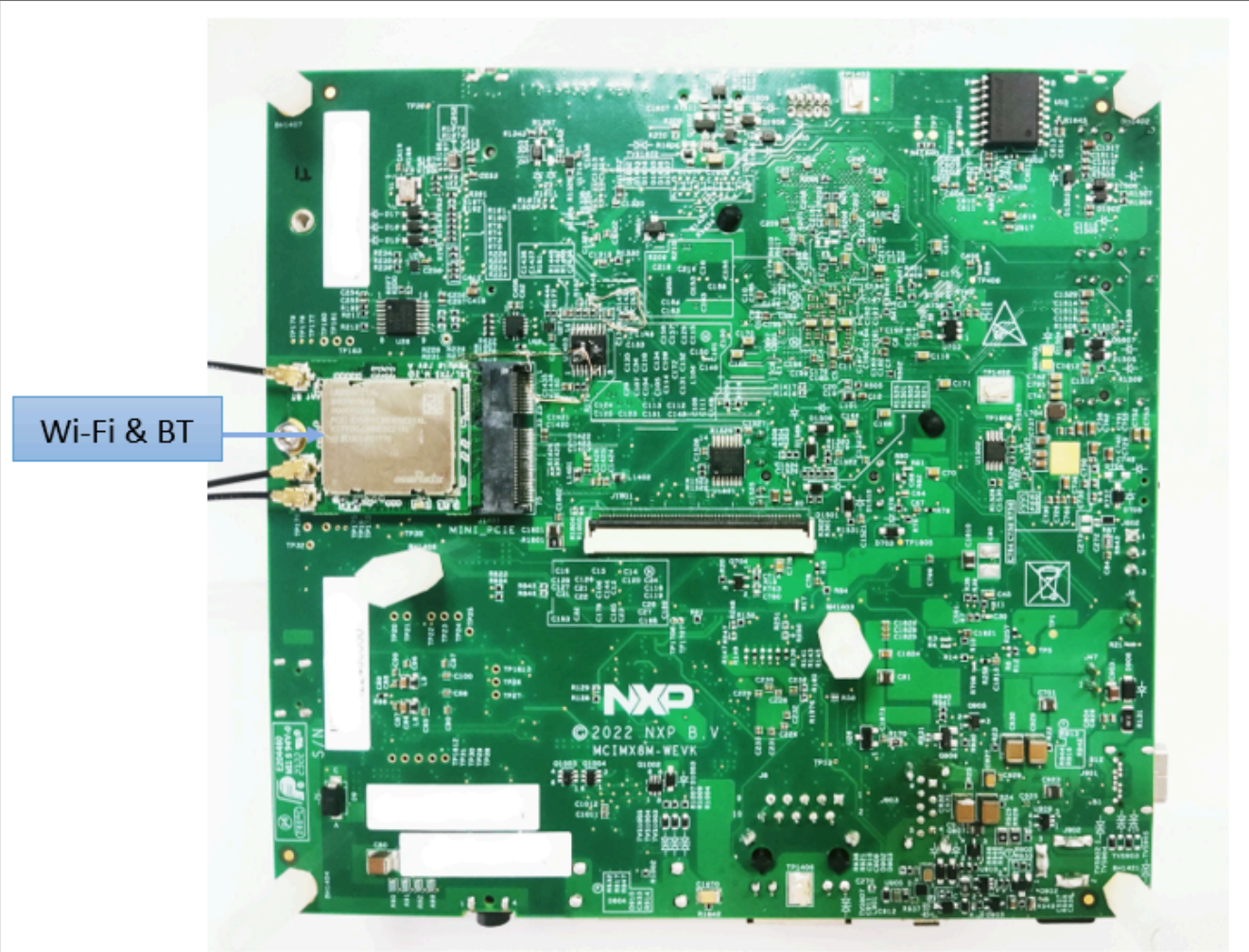


Figure 24. i.MX 8M Quad WEVK board BACK VIEW

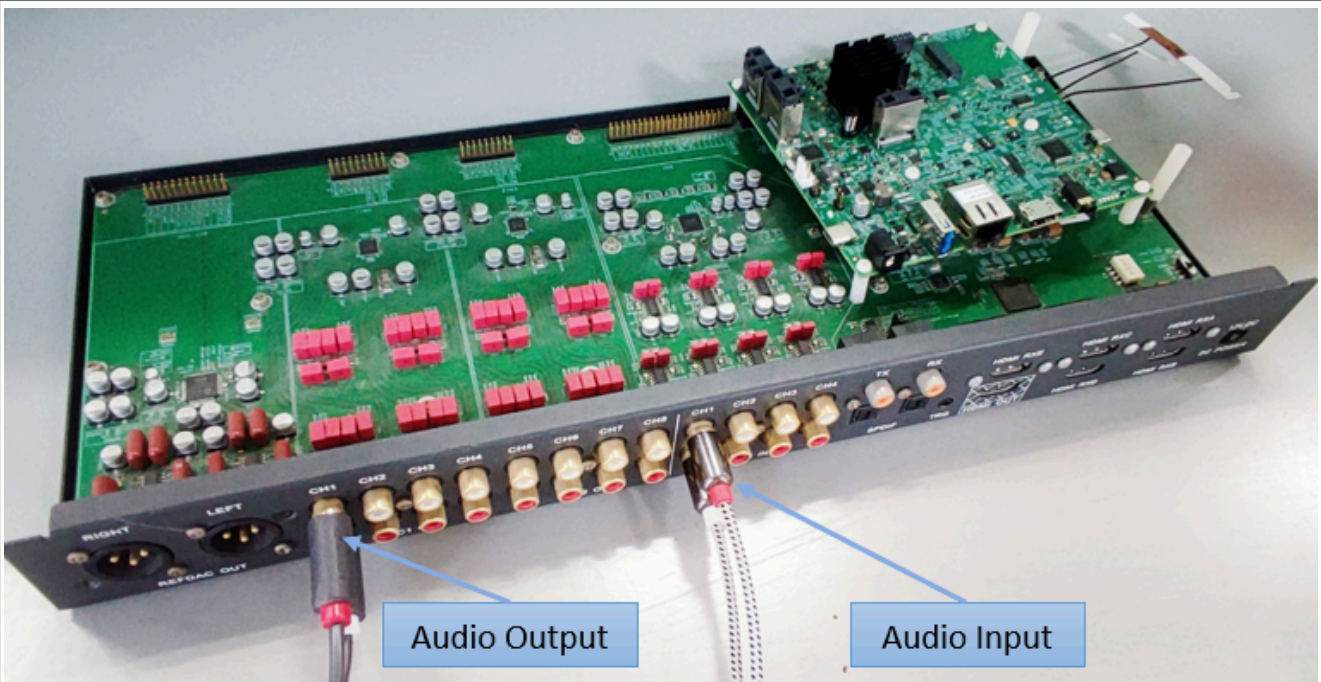


Figure 25. i.MX 8M Quad WEVK with audio board



Figure 26. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 27. MX8-DSI-OLED1 MIPI panel



Figure 28. OV5640 CSI MIPI camera



Figure 29. PCIE9098 (Murata LBEE5ZZ1XL)

Note:

- *i.MX 8M Quad WEVK 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 PCIE9098 (Murata LBEE5ZZ1XL) Wi-Fi&BT M.2 expansion card to the J1401 connector to enable Wi-Fi and Bluetooth.

6.2 Board images

The table below describes the location in the board partitions of the software images in android-14.0.0_1.0.0_image_8mqwevk.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.
spl-imx8mq-wevk-dual.bin	33 KB offset of MMC.
spl-imx8mq-trusty-wevk-dual.bin	33 KB offset of MMC.
spl-imx8mq-trusty-secure-unlock-wevk-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.
bootloader-imx8mq-trusty-secure-unlock-dual.img	bootloader_a and bootloader_b partitions.
bootloader-imx8mq-wevk-dual.img	bootloader_a and bootloader_b partitions.
bootloader-imx8mq-trusty-wevk-dual.img	bootloader_a and bootloader_b partitions.
bootloader-imx8mq-trusty-secure-unlock-wevk-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 EVK board, which is not flashed to MMC.
u-boot-imx8mq-wevk.imx	33 KB offset of MMC.
u-boot-imx8mq-wevk-uuu.imx	Bootloader used by UUU for i.MX 8M Quad WEVK 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.

Table 7. Board images...continued

Image name	Download target
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.
init_boot.img	init_boot_a and init_boot_b partitions to contain the generic ramdisk on GKI enabled system.
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 EVK Rev. A board HDMI output.
vbmeta-imx8mq-wevk.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad WEVK Rev. A board HDMI output.
vbmeta-imx8mq-mipi.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad WEVK/EVK Rev. A board MIPI-DSI-to-HDMI output.
vbmeta-imx8mq-dual.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad WEVK/EVK 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 WEVK/EVK 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 WEVK/EVK 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.
vendor_dlkm.img	Logical partition vendor_dlkm_a and logical partition vendor_dlkm_b in super partition.
product.img	Logical partition product_a and logical partition product_b in super partition.
super.img	Super partition.
dtbo-imx8mq.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad EVK Rev. A board HDMI output.
dtbo-imx8mq-wevk.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad WEVK Rev. A board HDMI output.
dtbo-imx8mq-mipi.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad WEVK/EVK Rev. A board MIPI-DSI-to-HDMI output.
dtbo-imx8mq-dual.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad WEVK/EVK 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 WEVK/EVK 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 WEVK/EVK Rev. A board RM67191 MIPI panel output.

Table 7. Board images...continued

Image name	Download target
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 WEVK 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-14.0.0_1.0.0_image_8mqwevk.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 WEVK, it should be <code>imx8mq</code> . 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 8. Options for `uuu_imx_android_flash` tool...continued

Option	Description
<code>-c card_size</code>	Specifies which partition table image file to flash. For i.MX 8M Quad WEVK, it can be followed with "28". If this option is not used, default <code>partition-table.img</code> or <code>partition-table-dual.img</code> is flashed.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&bootloader image with <code>uboot_feature</code> in their names. For i.MX 8M Quad WEVK, it can be <code>wevk</code> , <code>wevk-dual</code> , <code>trusty-wevk-dual</code> , <code>trusty-secure-unlock-wevk-dual</code> . For i.MX 8M Quad EVK, it can be <code>dual</code> , <code>trusty-dual</code> , <code>trusty-secure-unlock-dual</code> . If this option is not used, default <code>u-boot-imx8mq.img</code> is flashed.
<code>-d dtb_feature</code>	Flashes DTBO and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8M Quad WEVK, it can be <code>wevk</code> , <code>dual</code> , <code>mipi-panel</code> , <code>mipi-panel-rm67191</code> , <code>mipi</code> . For i.MX 8M Quad EVK, it can be <code>dual</code> , <code>mipi-panel</code> , <code>mipi-panel-rm67191</code> , <code>mipi</code> . If this option is not used, default <code>dtbo-imx8mq.img</code> and <code>vbmeta-imx8mq.img</code> are flashed.
<code>-e</code>	Erases user data after images are flashed.
<code>-D directory</code>	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash.bat</code> , it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
<code>-t target_dev</code>	Specifies the target device. For i.MX 8M Quad WEVK, it can be <code>emmc</code> and <code>sd</code> . If this option is not used, images are flashed to eMMC.
<code>-daemon</code>	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
<code>-i</code>	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.
<code>-dryrun</code>	Only generates a UUU script but not executes UUU with this script.
<code>-usb usb_path</code>	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 -d wevk -u trusty-wevk-dual
```

- 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 -d wevk -u trusty-wevk-dual
```

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

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 for i.MX 8MQuad EVK, execute the tool with `-u trusty-dual`.

- To test Trusty OS and dual-bootloader both in enabled condition for i.MX 8MQuad WEVK, execute the tool with `-u trusty-wevk-dual`.
- To test the demonstration implementation of secure unlock i.MX 8MQuad EVK, 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 demonstration implementation of secure unlock i.MX 8MQuad WEVK, execute the tool with `-u trusty-secure-unlock-wevk-dual`. For secure unlock details, see the i.MX Android Security User's Guide (ASUG).
- To test feature on i.MX 8M Quad WEVK Rev. A board:
 - To test HDMI display, execute the tool with `-d wevk`.
 - To test MIPI-to-HDMI display, execute the tool with `-d mipi`.
 - To test RM67199 MIPI panel display, execute the tool with `-d mipi-panel`.
 - To test RM67191 MIPI panel display, execute the tool with `-d mipi-panel-rm67191`.
 - To test HDMI and MIPI-DSI-to-HDMI dual-output, execute the tool with `-d dual`
- To test feature on i.MX 8M Quad EVK Rev. A board:
 - For i.MX 8M Quad EVK board, only HDMI display supports Wi-Fi and Bluetooth function. When connecting to other displays, Wi-Fi and Bluetooth function is not supported.
 - 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, 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.

- Wait for the `uuu_imx_android_flash` execution to complete. If there is no 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.

- Power off the board.
- Change boot device to eMMC or SD card. Change the board's SW802 (boot mode) to 10 (from 1-2 bit) to exit 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 WEVK/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 WEVK/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 WEVK/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 WEVK/EVK Rev. A Board:

```
U-Boot > setenv bootargs console=ttyMXC0,115200
earlycon=ec_imx6q,0x30860000,115200 init=/init androidboot.gui_resolution=
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.

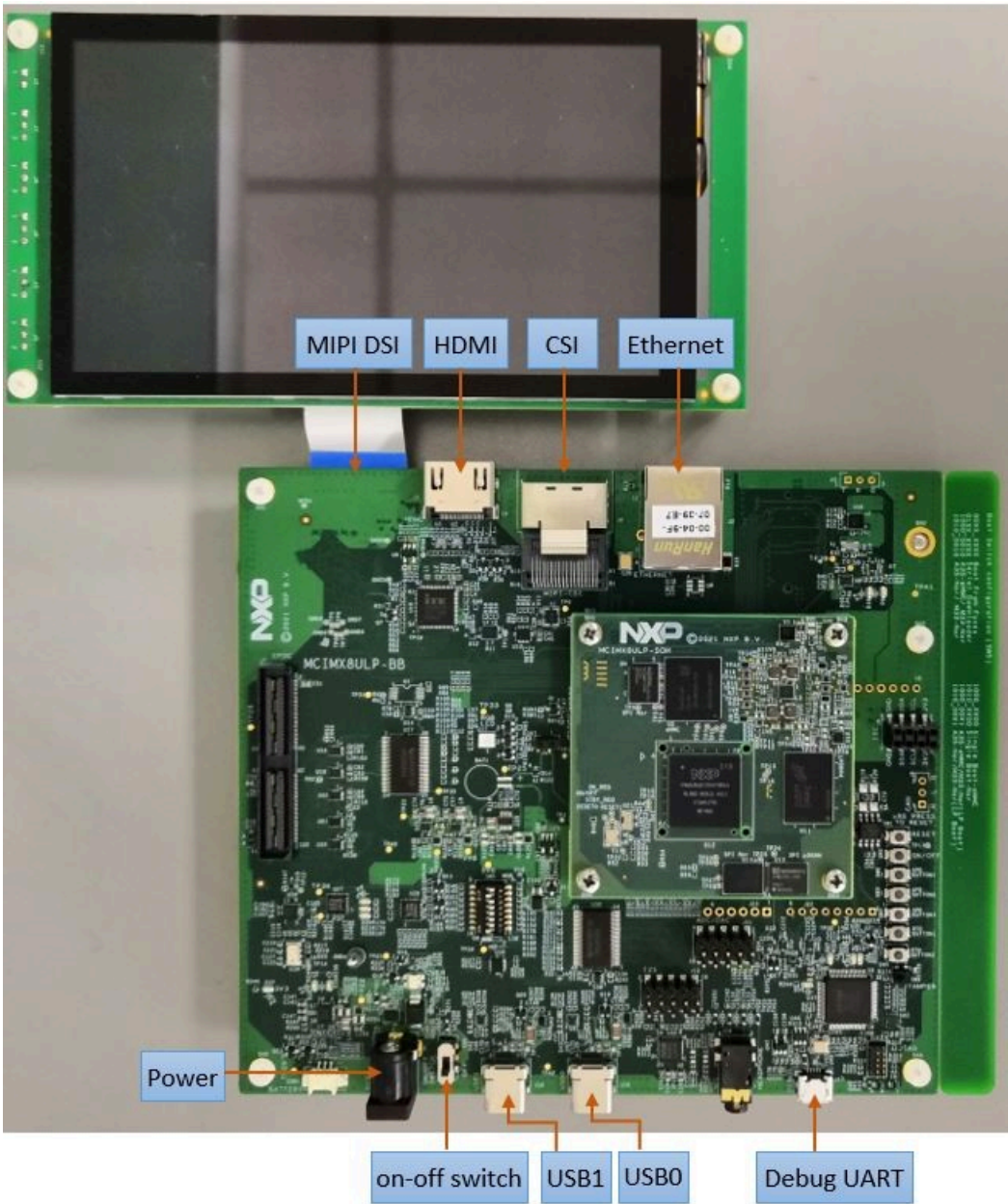


Figure 30. i.MX 8ULP EVK board with MIPI display

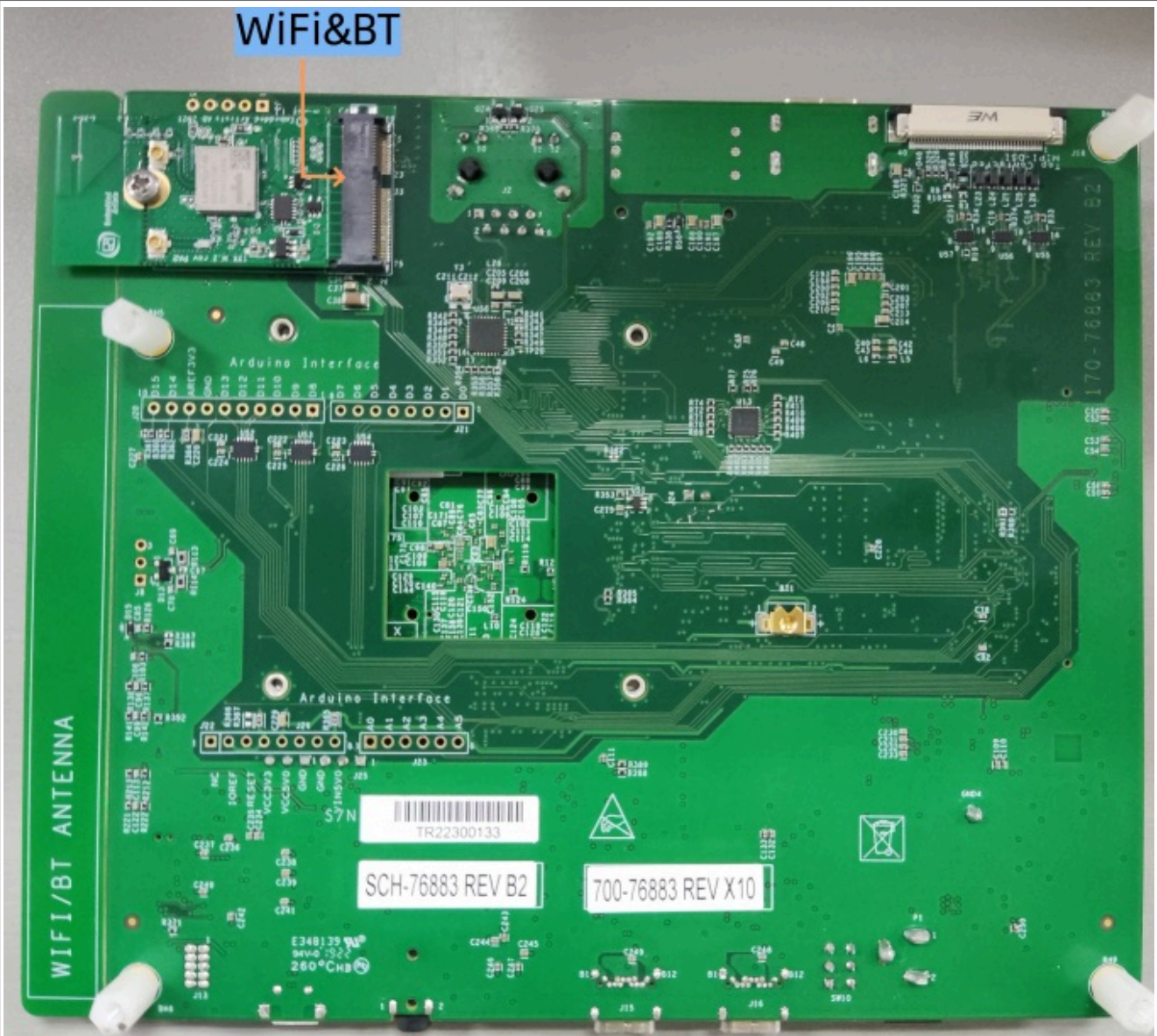


Figure 31. i.MX 8ULP EVK board back view

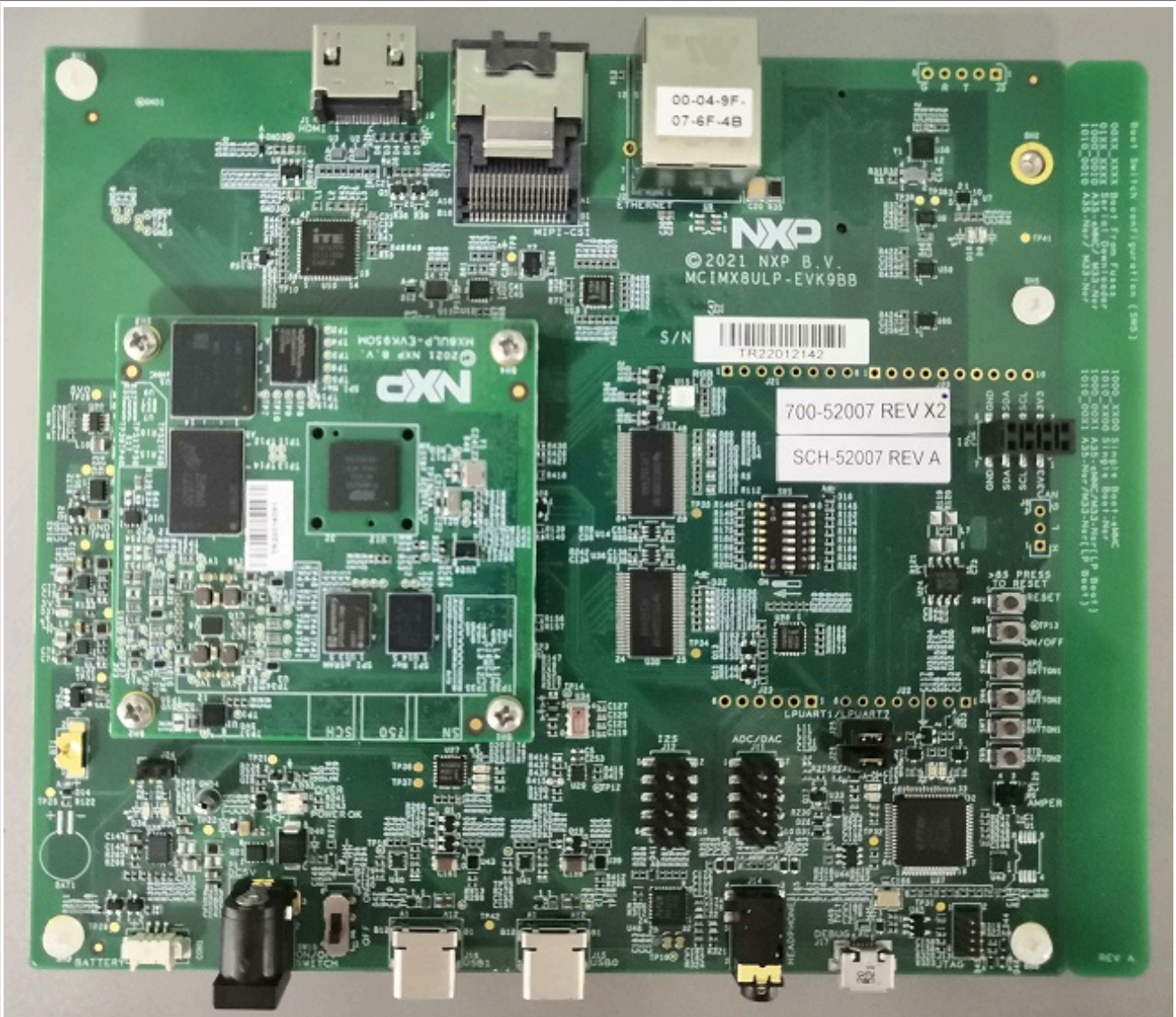


Figure 32. i.MX 8ULP EVK 9x9 board



Figure 33. i.MX CSI MIPI camera



Figure 34. EPDC display



Figure 35. SDIW416 (Murata LBEE5CJ1XK)

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.
- Connect the SDIW416 (Murata LBEE5CJ1XK) Wi-Fi&BT M.2 expansion card to the J3 connector to enable Wi-Fi and Bluetooth.

7.2 Board images

The table below describes the location in the board partitions of the software images in `android-14.0.0_1.0.0_image_8ulp-evk.tar.gz`.

Table 9. Board images

Image name	Download target
<code>spl-imx8ulp-dual.bin</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK.
<code>spl-imx8ulp-trusty-dual.bin</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK.
<code>spl-imx8ulp-trusty-9x9-dual.bin</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK 9x9 board.
<code>spl-imx8ulp-trusty-lpa-dual.bin</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK with Low Power Audio (LPA) enabled.
<code>spl-imx8ulp-trusty-dualboot-dual.bin</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK with Low Power Display (LPD) enabled.
<code>bootloader-imx8ulp-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8ULP EVK.
<code>bootloader-imx8ulp-trusty-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8ULP EVK.
<code>bootloader-imx8ulp-trusty-9x9-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8ULP EVK 9x9 board.
<code>bootloader-imx8ulp-trusty-lpa-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8ULP EVK with Low Power Audio (LPA) enabled.
<code>bootloader-imx8ulp-trusty-dualboot-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8ULP EVK with Low Power Display (LPD) enabled.
<code>u-boot-imx8ulp-9x9.imx</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK 9x9 board.
<code>u-boot-imx8ulp-trusty-9x9.imx</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK 9x9 board.
<code>u-boot-imx8ulp-9x9-evk-uuu.imx</code>	The bootloader used by UUU for i.MX 8ULP EVK 9x9 board. It is not flashed to MMC.
<code>u-boot-imx8ulp.imx</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK.
<code>u-boot-imx8ulp-trusty.imx</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK.
<code>u-boot-imx8ulp-trusty-secure-unlock.imx</code>	0 offset of eMMC boot partition for i.MX 8ULP EVK.
<code>u-boot-imx8ulp-evk-uuu.imx</code>	The bootloader used by UUU for i.MX 8ULP EVK. It is not flashed to MMC.
<code>imx8ulp_mcu_demo_sf.img</code>	MCU image that contain low power display demo, it will be flashed to serial flash in MCU side.

Table 9. Board images...continued

Image name	Download target
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.
init_boot.img	init_boot_a and init_boot_b partitions to contain the generic ramdisk on GKI enabled system.
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.
vbmeta-imx8ulp-lpd.img	vbmeta_a and vbmeta_b partitions to support the Low Power Display feature 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.
vendor_dlkm.img	Logical partition vendor_dlkm_a and logical partition vendor_dlkm_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 partitions to support HDMI output on i.MX 8ULP EVK 9x9 board.
dtbo-imx8ulp-9x9.img	dtbo_a and dtbo_b partitions to support MIPI panel output on i.MX 8ULP EVK 9x9 board.

Table 9. Board images...continued

Image name	Download target
dtbo-imx8ulp.img	dtbo_a and dtbo_b partitions to support MIPI panel output on i.MX 8ULP EVK.
dtbo-imx8ulp-hdmi.img	dtbo_a and dtbo_b partitions to support HDMI output on i.MX 8ULP EVK.
dtbo-imx8ulp-epdc.img	dtbo_a and dtbo_b partitions 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.
dtbo-imx8ulp-lpd.img	dtbo_a and dtbo_b partitions to support the Low Power Display 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.
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: [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 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-14.0.0_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
<code>-h</code>	Displays the help information of this tool.

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

Option	Description
<code>-f soc_name</code>	Specifies SoC information. For i.MX 8ULP EVK, it should be <code>imx8ulp</code> . This option is mandatory .
<code>-a</code>	Only flashes slot a. If this option and <code>-b</code> option are not used, slots a and b are both flashed.
<code>-b</code>	Only flashes slot b. If this option and <code>-a</code> option are not used, slots a and b are both flashed.
<code>-c card_size</code>	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 <code>partition-table.img</code> is flashed.
<code>-m</code>	Flash the mcu image. If this option is not used, the MCU image is not flashed.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&bootloader image with <code>uboot_feature</code> in their names. For i.MX 8ULP EVK, it can be <code>9x9</code> , <code>dual</code> , <code>9x9-dual</code> , <code>trusty-dual</code> , <code>trusty-9x9-dual</code> , <code>trusty-lpa-dual</code> , <code>trusty-dualboot-dual</code> , <code>trusty-dualboot-dual</code> . If this option is not used, the default <code>u-boot-imx8ulp.imx</code> is flashed.
<code>-d dtb_feature</code>	Flashes DTBO and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8ULP EVK, it can be <code>9x9</code> , <code>9x9-hdmi</code> , <code>hdmi</code> , <code>epdc</code> , <code>sof</code> , <code>lpa</code> and <code>lpd</code> . If this option is not used, default <code>dtbo-imx8ulp.img</code> and <code>vbmeta-imx8ulp.img</code> are flashed.
<code>-e</code>	Erases user data after images are flashed.
<code>-D directory</code>	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash.bat</code> , it must be followed with an absolute path. If this option is not used, images in current working directory are flashed.
<code>-daemon</code>	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
<code>-i</code>	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.
<code>-dryrun</code>	Only generates a UUU script but not executes UUU with this script.
<code>-usb usb_path</code>	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, and `bootloader-imx8ulp-trusty-dual.img` with other default images are flashed into the eMMC slot 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 MIPI panel display, it does not need to use the `-d` option.
 - To test the HDMI display, execute the tool with `-d hdmi`.
 - To test the EPDC display, execute the tool with `-d epdc`.

- To test low-power audio, execute the tool with `-d lpa -u trusty-lpa-dual -m`. See the Android User's Guide (AUG) for more steps to make the low-power audio work.
- To test low-power display, execute the tool with `-d lpd -u trusty-dualboot-dual -m`. See the Android User's Guide (AUG) for more steps to make the low-power display work.
- For i.MX 8ULP EVK 9x9 board:
 - To test the MIPI panel display, execute the tool with `-d 9x9`.
 - To test the 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, 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 no error, the command window displays the information indicating that images are already flashed.
7. Power off the board.
8. Change the boot device to eMMC.
 - Change SW5 to 00000001 to boot from eMMC.
 - Change SW5 to 01000001 (dual-boot mode). The Cortex-A core is booted from eMMC, and the Cortex-M core is booted from serial flash.

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=640M 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=640M 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=640M 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.

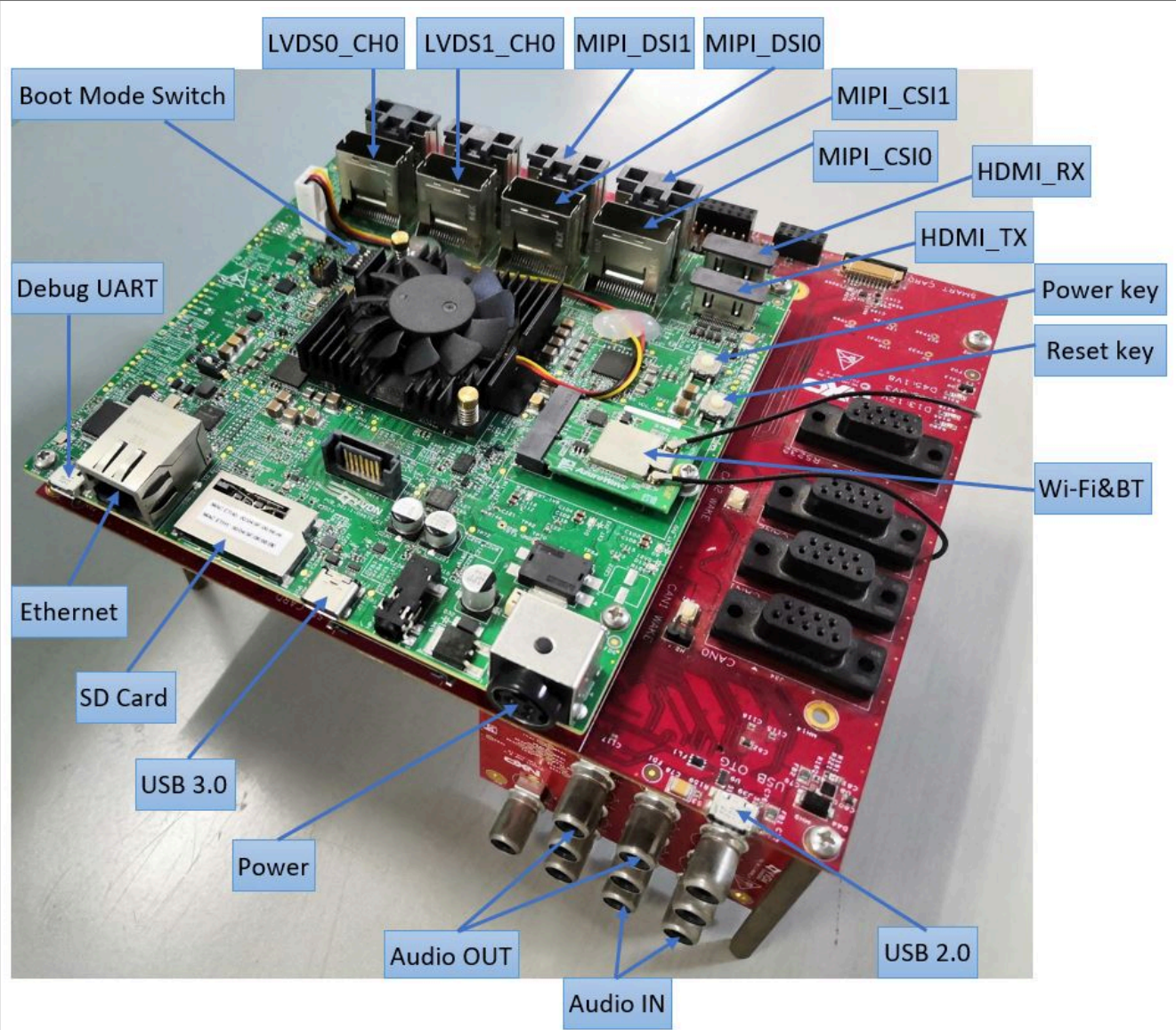


Figure 36. i.MX 8QuadMax MEK board



Figure 37. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 38. i.MX mini SAS cable with LVDS-to-HDMI adapter



Figure 39. MX8-DSI-OLED1 MIPI panel



Figure 40. i.MX MIPI camera



Figure 41. PCIE9098 (Murata LBEE5ZZ1XL)

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.
- Connect the PCIE9098 (Murata LBEE5ZZ1XL) Wi-Fi&BT M.2 expansion card to the J12 connector to enable Wi-Fi and Bluetooth.

8.2 Board images

The table below describes the location in the board partitions of the software images in android-14.0.0_1.0.0_image_8qmek.tar.gz.

Table 11. Board images

Image name	Download target
spl-imx8qm-dual.bin	0 KB offset of eMMC or 32 KB offset of SD card.
bootloader-imx8qm-dual.img	bootloader_a and bootloader_b partitions.
spl-imx8qm-trusty-dual.bin	0 KB offset of eMMC.
bootloader-imx8qm-trusty-dual.img	bootloader_a and bootloader_b partitions.
spl-imx8qm-trusty-secure-unlock-dual.bin	0 KB 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.
init_boot.img	init_boot_a and init_boot_b partitions to contain the generic ramdisk on GKI enabled system.
vendor_boot.img	vendor_boot_a and vendor_boot_b partitions.

Table 11. Board images...continued

Image name	Download target
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 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 OLED1 (RM67199) MIPI panel display.
vbmeta-imx8qm-mipi-panel-rm67191.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.
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.
vendor_dlkm.img	Logical partition vendor_dlkm_a and logical partition vendor_dlkm_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 (RM67199) MIPI panel display.
dtbo-imx8qm-mipi-panel-rm67191.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".

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-14.0.0_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
<code>-h</code>	Displays the help information of this tool.
<code>-f soc_name</code>	Specifies SoC information. For i.MX 8QuadMax MEK, it should be <code>imx8qm</code> . This option is mandatory .
<code>-a</code>	Only flashes slot a. If this option and <code>-b</code> option are not used, slots a and b are both flashed.
<code>-b</code>	Only flashes slot b. If this option and <code>-a</code> option are not used, slots a and b are both flashed.
<code>-c card_size</code>	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 <code>partition-table.img</code> is flashed.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&Bootloader image with <code>uboot_feature</code> in their names. For i.MX 8QuadMax MEK, it can be <code>hdmi</code> , <code>md</code> , <code>dual</code> , <code>trusty-dual</code> , or <code>trusty-secure-unlock-dual</code> . If this option is not used, the default <code>u-boot-imx8qm.imx</code> is flashed.
<code>-d dtb_feature</code>	Flashes DTBO and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8QuadMax MEK, it can be <code>hdmi-rx</code> , <code>hdmi</code> , <code>md</code> , <code>mipi-panel</code> , <code>mipi-panel-rm67191</code> , <code>lvds1-panel</code> , or <code>sof</code> . If this option is not used, the default <code>dtbo-imx8qm.img</code> and <code>vbmeta-imx8qm.img</code> are flashed.
<code>-e</code>	Erases user data after images are flashed.

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

Option	Description
<code>-D directory</code>	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash.bat</code> , it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
<code>-t target_dev</code>	Specifies the target device. For i.MX 8QuadMax, it can be <code>emmc</code> and <code>sd</code> . If this option is not used, images are flashed to eMMC.
<code>-daemon</code>	Runs UUU in Daemon mode. This option is used to flash multiple boards of the same type.
<code>-i</code>	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.
<code>-dryrun</code>	Only generates a UUU script but not executes UUU with this script.
<code>-usb usb_path</code>	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 demonstration 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 individually or together.
- To test MIPI panel (RM67199) display, execute the tool with `-d mipi-panel`. The MIPI_DSI0 port is used for display.
- To test MIPI panel (RM67191) display, execute the tool with `-d mipi-panel-rm67191`. The MIPI_DSI0 port is used for display.
- To test LVDS panel display, execute the tool with `-d lvds1-panel`. The LVDS1 port is used for display. The connection is channel 0 (J8) -> LCD J9, channel 1 (J3) -> LCD J8.
- To test physical HDMI display, execute the tool with `-u hdmi -d hdmi`. The HDMI_TX port is used for display.
- To test physical HDMI RX as external camera input device, execute the tool with `-d hdmi-rx`. The HDMI_RX port is used for HDMI RX. In addition, OV5640 on CSI1 could work well at the same time.

- 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, 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 no 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 to 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.

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.

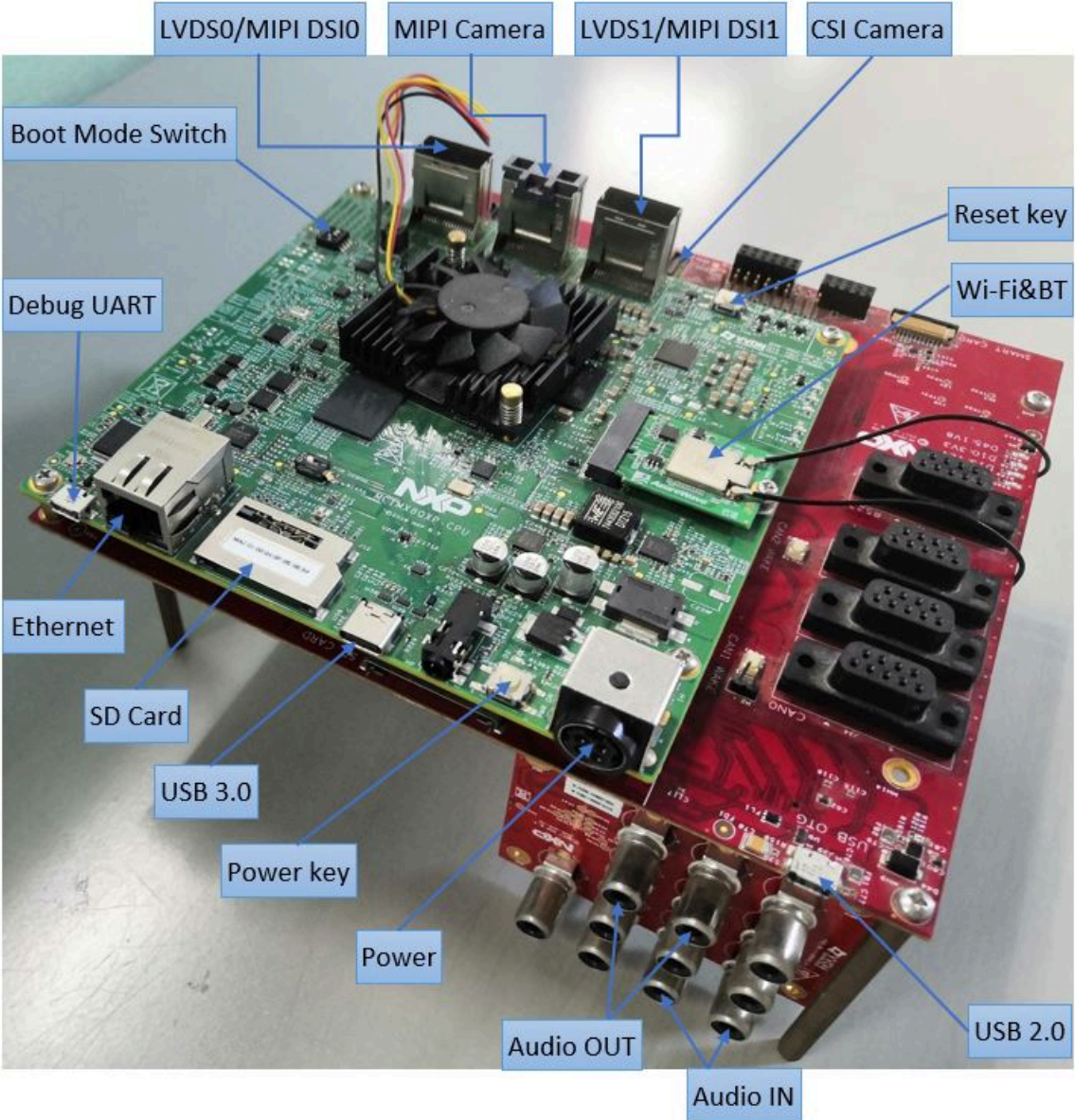


Figure 42. i.MX 8QuadXPlus MEK board



Figure 43. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 44. i.MX mini SAS cable with LVDS-to-HDMI adapter



Figure 45. OV5640 CSI MIPI camera



Figure 46. PCIE9098 (Murata LBEE5ZZ1XL)

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.
- Connect the PCIE9098 (Murata LBEE5ZZ1XL) Wi-Fi&BT M.2 expansion card to the J5 connector to enable Wi-Fi and Bluetooth.

9.2 Board images

The table below describes the location in the board partitions of the software images in `android-14.0.0_1.0.0_image_8qmek.tar.gz`.

Table 13. Board images

Image name	Download target
<code>spl-imx8qxp-dual.bin</code>	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
<code>bootloader-imx8qxp-dual.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
<code>spl-imx8qxp-c0-dual.bin</code>	0 KB offset of eMMC or 32 KB offset of SD card for i.MX 8QuadXPlus MEK board with silicon revision C0 chip.

Table 13. Board images...continued

Image name	Download target
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	32 KB 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	32 KB 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	0 KB 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.img	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with silicon revision B0 chip.
u-boot-imx8qxp-c0.img	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.img	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.img	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.
init_boot.img	init_boot_a and init_boot_b partitions to contain the generic ramdisk on GKI enabled system.
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-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support MIPI panel (RM67199) display.
vbmeta-imx8qxp-mipi-panel-rm67191.img	vbmeta_a and vbmeta_b partitions to support MIPI panel (RM67191) display.

Table 13. Board images...continued

Image name	Download target
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.
vendor_dlkmm.img	Logical partition vendor_dlkmm_a and logical partition vendor_dlkmm_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-camera support.
dtbo-imx8qxp-lvds0-panel.img	dtbo_a and dtbo_b partitions to support LVDS panel display.
dtbo-imx8qxp-mipi-panel.img	dtbo_a and dtbo_b partitions to support MIPI panel (RM67199) display.
dtbo-imx8qxp-mipi-panel-rm67191.img	dtbo_a and dtbo_b partitions to support MIPI panel (RM67191) display.
dtbo-imx8qxp-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.

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-14.0.0_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
<code>-h</code>	Displays the help information of this tool.
<code>-f soc_name</code>	Specifies SoC information. For i.MX 8QuadXPlus MEK, it should be <code>imx8qxp</code> . This option is mandatory .
<code>-a</code>	Only flashes slot a. If this option and <code>-b</code> option are not used, slots a and b are both flashed.
<code>-b</code>	Only flashes slot b. If this option and <code>-a</code> option are not used, slots a and b are both flashed.
<code>-c card_size</code>	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 <code>partition-table.img</code> is flashed.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&bootloader image with <code>uboot_feature</code> in their names. For i.MX 8QuadXPlus MEK, it can be <code>dual</code> , <code>trusty-dual</code> , <code>trusty-secure-unlock-dual</code> , <code>c0</code> , <code>c0-dual</code> or <code>trusty-c0-dual</code> . If this option is not used, the default <code>u-boot-imx8qxp.img</code> is flashed.
<code>-d dtb_feature</code>	Flash DTBO and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8QuadXPlus MEK, it can be <code>sof</code> , <code>mipi-panel</code> , <code>mipi-panel-rm67191</code> or <code>lvds0-panel</code> . If it is not used, the default <code>dtbo-imx8qxp.img</code> and <code>vbmeta-imx8qxp.img</code> are flashed.
<code>-e</code>	Erases user data after images are flashed.
<code>-D directory</code>	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash.bat</code> , it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
<code>-t target_dev</code>	Specifies the target device. For i.MX 8QuadXPlus MEK, it can be <code>emmc</code> and <code>sd</code> . If this option is not used, images are flashed to eMMC.
<code>-daemon</code>	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
<code>-i</code>	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.
<code>-dryrun</code>	Only generates a UUU script but not executes UUU with this script.
<code>-usb usb_path</code>	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 demonstration 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`.
- To test MIPI panel(RM67199) display, execute the tool with `-d mipi-panel`. The MIPI_DSI0 port is used for display.
- To test MIPI panel(RM67191) display, execute the tool with `-d mipi-panel-rm67191`. The MIPI_DSI0 port is used for display.
- To test LVDS panel display, execute the tool with `-d lvds0-panel`. The LVDS0/1 port is used for display. The connection is LVDS 0 (J3) -> LCD J9, LVDS 1 (J1) -> LCD J8.
- If `uuu_imx_android_flash.bat` is used to flash images on a remote server through Samba, 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.

- Wait for the `uuu_imx_android_flash` execution to complete. If there is no 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.

- Power off the board.
- Change boot device to 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
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 Note About the Source Code in the Document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

11 Revision History

Revision history

Revision number	Release date	Description
android-14.0.0_1.0.0	6 February 2024	i.MX 8ULP EVK, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.
android-13.0.0_2.2.0	24 October 2023	i.MX 8ULP EVK, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.
android-13.0.0_2.0.0	07/2023	i.MX 8ULP EVK Beta release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.
android-13.0.0_1.2.0	03/2023	i.MX 8ULP EVK Beta release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.
android-13.0.0_1.0.0	01/2023	i.MX 8ULP EVK Beta release, i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8QuadXPlus 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 8QuadMax, and i.MX 8QuadXPlus 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-12.0.0_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-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-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_2.2.0	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_2.0.0	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_1.0.0	12/2020	i.MX 8M Plus EVK Beta release, and all the other i.MX 8 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.
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_2.1.0	04/2020	i.MX 8M Plus Alpha and i.MX 8QuadXPlus Beta release.
android-10.0.0_1.0.0	03/2020	Deleted the Android 10 image.
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.
P9.0.0_2.0.0-ga	08/2019	Updated the location of the SCFW porting kit.
P9.0.0_2.0.0-ga	04/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.
P9.0.0_1.0.0-ga	01/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.
P9.0.0_1.0.0-beta	11/2018	Initial release

Legal information

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.

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 <https://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 document 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.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

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.

Contents

1	Overview	2	7	Working with the i.MX 8ULP EVK Board	43
2	Hardware Requirements	2	7.1	Board hardware	43
3	Working with the i.MX 8M Mini EVK Board	2	7.2	Board images	48
3.1	Board hardware	2	7.3	Flashing board images	50
3.2	Board images	7	7.4	Bootting	52
3.3	Flashing board images	8	7.4.1	Bootting with MIPI panel display	52
3.4	Bootting	11	7.4.2	Bootting with HDMI display	52
3.4.1	Bootting with Single MIPI-DSI-to-HDMI or MIPI panel display	11	7.4.3	Bootting with EPDC display	53
3.4.2	Bootting with Single MIPI-DSI-to-HDMI display and audio playback based on Cortex-M4 FreeRTOS	11	7.5	Board reboot	53
3.5	Board reboot	12	8	Working with the i.MX 8QuadMax MEK Board	53
4	Working with the i.MX 8M Nano EVK Board	12	8.1	Board hardware	53
4.1	Board hardware	12	8.2	Board images	56
4.2	Board images	15	8.3	Flashing board images	57
4.3	Flashing board images	17	8.4	Bootting	60
4.4	Bootting	19	8.4.1	Bootting with LVDS-to-HDMI/MIPI-DSI-to-HDMI display	60
4.4.1	Bootting with single MIPI-DSI-to-HDMI/MIPI panel display	19	8.4.2	Bootting with physical HDMI display	60
4.4.2	Bootting with single MIPI-DSI-to-HDMI display with an MCU image	19	8.4.3	Bootting with dual LVDS-to-HDMI displays	61
4.5	Board reboot	20	8.5	Board reboot	61
5	Working with the i.MX 8M Plus EVK Board	20	9	Working with the i.MX 8QuadXPlus MEK Board	61
5.1	Board hardware	20	9.1	Board hardware	61
5.2	Board images	27	9.2	Board images	63
5.3	Flashing board images	29	9.3	Flashing board images	65
5.4	Bootting	31	9.4	Bootting	67
5.4.1	Bootting with single display: HDMI /MIPI-to-HDMI/MIPI panel/LVDS panel/dual channel LVDS to HDMI or multiple displays	32	9.4.1	Bootting with single LVDS-to-HDMI/MIPI-DSI-to-HDMI or dual LVDS-to-HDMI displays	67
5.4.2	Bootting with single MIPI-to-HDMI display and audio playback based on Cortex-M7 FreeRTOS	32	9.4.2	Bootting with dual LVDS-to-HDMI displays	68
5.4.3	Bootting with single display: HDMI 4K display	32	9.5	Board reboot	68
5.5	Board reboot	33	10	Note About the Source Code in the Document	68
6	Working with the i.MX 8M Quad WEVK Board	33	11	Revision History	69
6.1	Board hardware	33		Legal information	70
6.2	Board images	37			
6.3	Flashing board images	39			
6.4	Bootting	41			
6.4.1	Bootting with single display: HDMI display	41			
6.4.2	Bootting with single display: MIPI-DSI-to-HDMI display	42			
6.4.3	Bootting with dual displays: HDMI and MIPI-DSI-to-HDMI displays	42			
6.4.4	Bootting with single display: MIPI panel	42			
6.5	Board reboot	43			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.