UG10157

Android Quick Start Guide

Rev. android-15.0.0_1.2.0 — 11 April 2025

User guide

Document information

Information	Content
Keywords	Android, i.MX, android-15.0.0_1.2.0
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* (UG10156) 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)
- i.MX 95

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 figures below show the different components of the i.MX 8M Mini EVK LPDDR4 board.

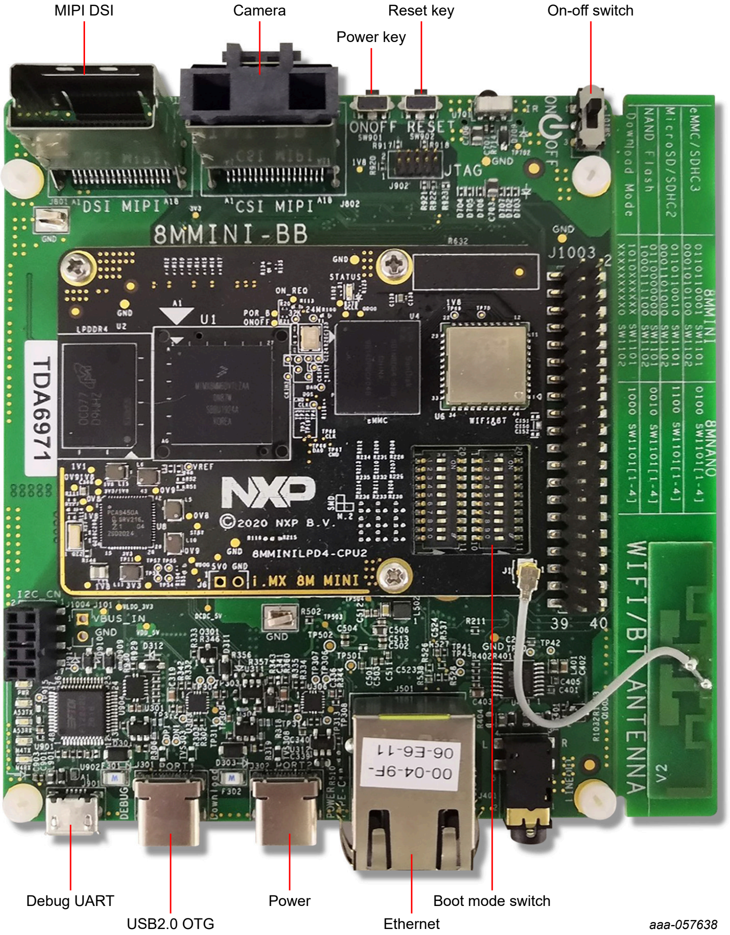


Figure 1. i.MX 8M Mini EVK LPDDR4 board

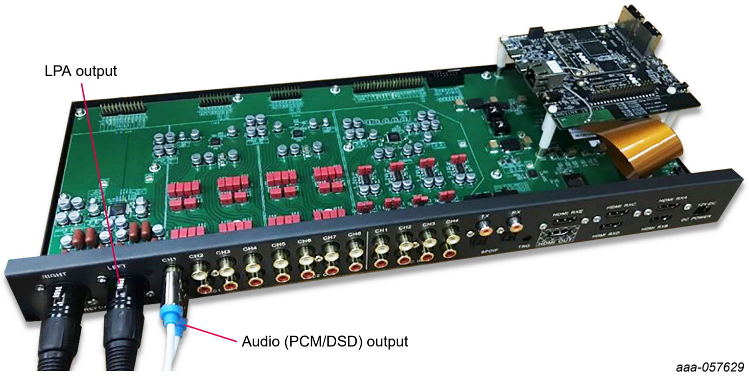
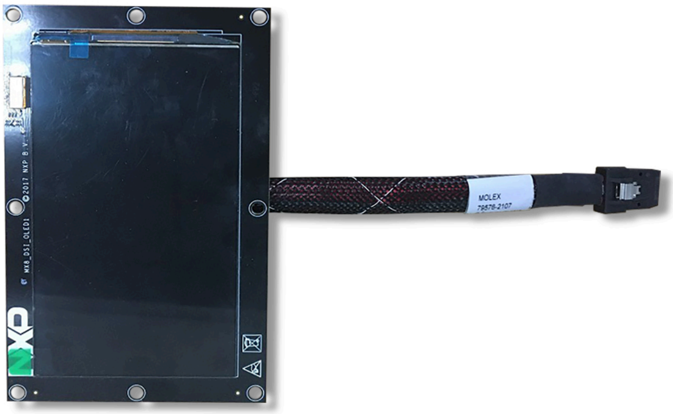


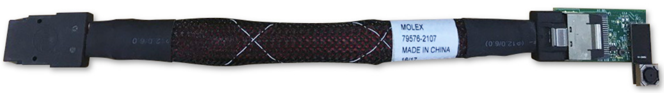
Figure 2. i.MX 8M Mini EVK with audio board

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



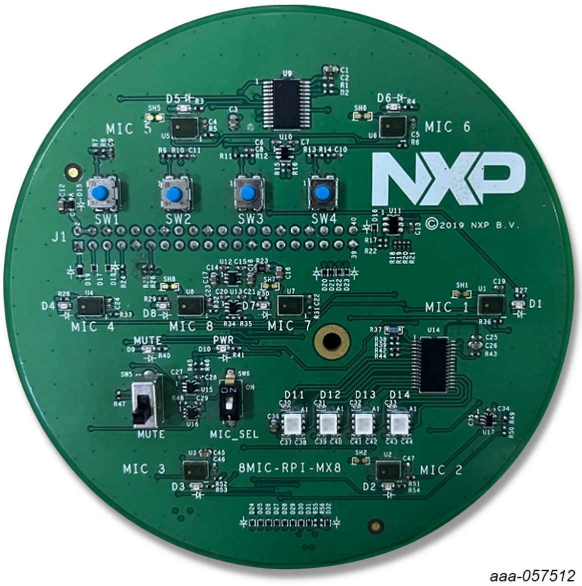
aaa-057542

Figure 4. MX8-DSI-OLED1 MIPI panel



aaa-058037

Figure 5. OV5640 CSI MIPI camera



aaa-057512

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-15.0.0_1.2.0_image_8mmevk.tar.gz`.

Table 1. Board images

Image name	Download target	Description
`spl-imx8mm-dual.bin`	33 kB offset of MMC.	Secondary program loader image without Trusty related configurations for the i.MX 8M Mini EVK LPDDR4 board.
`spl-imx8mm-trusty-dual.bin`	33 kB offset of MMC.	Secondary program loader image with Trusty related configurations for the i.MX 8M Mini EVK LPDDR4 board.
`spl-imx8mm-trusty-secure-unlock-dual.bin`	33 kB offset of MMC.	Secondary program loader image with Trusty and secure unlock related configurations for the i.MX 8M Mini EVK LPDDR4 board.
`bootloader-imx8mm-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8M Mini EVK LPDDR4 board
`bootloader-imx8mm-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8M Mini EVK LPDDR4 board.
`bootloader-imx8mm-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of secure unlock mechanism for i.MX 8M Mini EVK LPDDR4 board.
`u-boot-imx8mm.imx`	33 kB offset of MMC.	An image containing U-Boot and ATF for the i.MX 8M Mini EVK LPDDR4 board.
`u-boot-imx8mm-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8M Mini board with LPDDR4 on it. It is not flashed to MMC.
`u-boot-imx8mm-ddr4.imx`	33 kB offset of SD card.	An image containing U-Boot and ATF for the i.MX 8M Mini EVK DDR4 board.
`u-boot-imx8mm-ddr4-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8M Mini board with DDR4 on it. it is not flashed to SD card.
`imx8mm_mcu_demo.img`	5120 kB offset of MMC.	The MCU FreeRTOS image for the i.MX 8M Mini EVK board.
`partition-table.img`	0 offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image and generic ramdisk.

Table 1. Board images*...continued*

Image name	Download target	Description
boot-imx.img	boot_a and boot_b partitions.	Boot image built with the i.MX kernel tree for debugging.
init_boot.img	init_boot_a and init_boot_b partitions.	Contains generic ramdisk.
vendor_boot.img	vendor_boot_a and vendor_boot_b partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
vendor_boot-debug.img	vendor_boot_a and vendor_boot_b partitions.	Used in VTS test with the GSI system image.
vbmeta-imx8mm.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output on the i.MX 8M Mini EVK LPDDR4 board.
vbmeta-imx8mm-m4.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output and audio playback based on Cortex-M4 Free RTOS on the i.MX 8M Mini EVK LPDDR4 board.
vbmeta-imx8mm-mipi-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support RM67199 MIPI panel output on the i.MX 8M Mini EVK LPDDR4 board.
vbmeta-imx8mm-mipi-panel-rm67191.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support RM67191 MIPI panel output on the i.MX 8M Mini EVK LPDDR4 board.
vbmeta-imx8mm-ddr4.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output on the i.MX 8M Mini EVK DDR4 board.
system.img	Logical partitions system_a and system_b in super partition.	System image.
system_dlkm.img	logical partitions system_dlkm_a and system_dlkm_b in super partition.	System dynamically loadable kernel module image.
system_ext.img	Logical partitions system_ext_a and system_ext_b in super partition.	System extension image.
vendor.img	Logical partitions vendor_a and vendor_b in super partition.	Vendor image
vendor_dlkm.img	Logical partitions vendor_dlkm_a and vendor_dlkm_b in super partition.	Vendor dynamically loadable kernel module image.
product.img	Logical partitions product_a and product_b in super partition.	Product image.
super.img	Super partition.	Super image, which contains images for logical partitions.
dtbo-imx8mm.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output on the i.MX 8M Mini EVK LPDDR4 board.
dtbo-imx8mm-m4.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output and audio playback based on Cortex-M4 FreeRTOS on the i.MX 8M Mini EVK LPDDR4 board.

Table 1. Board images*...continued*

Image name	Download target	Description
dtbo-imx8mm-mipi-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support RM67199 MIPI panel output on the i.MX 8M Mini EVK LPDDR4 board.
dtbo-imx8mm-mipi-panel-rm67191.img	dtbo_a and dtbo_b partitions.	Device Tree image to support RM67191 MIPI panel output on the i.MX 8M Mini EVK LPDDR4 board.
dtbo-imx8mm-ddr4.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output playback on the i.MX 8M Mini EVK DDR4 board.
rpmb_key_test.bin	N/A	Prebuilt test RPMB key, which can be used to set the RPMB key as fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set RPMB key.
testkey_public_rsa4096.bin	N/A	Prebuilt AVB public key, which is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.

Note: *In this document, logical partition is also called dynamic partition. For more Android dynamic partitions, see https://source.android.com/docs/core/ota/dynamic_partitions.*

*With virtual A/B enabled, the super.img file generated by the Android build system only has slot a dynamic partitions filled with images. Slot b dynamic partitions like system_b is **0** in size. Therefore, the system cannot boot up from slot b with prebuilt images. To boot from slot b, take either of the following actions:*

- *Apply an OTA update.*
- *Boot the board into fastbootd mode, and flash the images to slot b logical partitions. Set slot b as the active slot.*

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 1.5.179 version. Download the 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 the 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-15.0.0_1.2.0_image_8mmevk.tar.gz`. The package contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Mini board, related options are described as follows.

Table 2. Options for uuu_imx_android_flash tool

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

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

- On the 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 the 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:

- *`-b` is not suggested to be used, because virtual A/B is enbaled and the system cannot boot up from slob b with the prebuilt images.*
- *`-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 the 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 (UG10158).*
  - *To test MIPI-DSI to HDMI output, it does not need to use `-d` option. `dtbo-imx8mm.img` is flashed in this condition to support Wi-Fi expansion card and this image does not support to be booted from the SD card.*
  - *To test RM67199 MIPI panel output, execute the tool with `-d mipi-panel`.*
  - *To test RM67191 MIPI panel output, execute the tool with `-d mipi-panel-rm67191`.*
  - *To test support MIPI-DSI-to-HDMI output and audio playback based on Cortex-M4 FreeRTOS, execute the tool with `-m` and `-d m4`.*
- *For i.MX 8M Mini EVK DDR4 board: execute the command with `-u ddr4 -d ddr4 -t sd`.*
- *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 (UG10156) 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.

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, the command window displays information indicating that images are already flashed.

   Note:

   If the target device has a DOS partition table on it, the flash process fails for the first time. Push the reset key on the board and execute the flash script again.
7. Power off the board.
8. Change boot device to eMMC or SD card.

   For Rev. C boards:
   - To boot from the SD card, change SW1101 to 0110110010 (from 1-10 bit) and change SW1102 to 0001101000 (from 1-10 bit).
   - To boot from eMMC, change SW1101 to 0110110001 (from 1-10 bit) and change SW1102 to 0001010100 (from 1-10 bit).

3.4 Booting

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

Some specific U-Boot enviroment variables may need to be set in some boot scenarios to boot the MCU image or to get better display effect.

3.4.1 Booting with 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 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
```

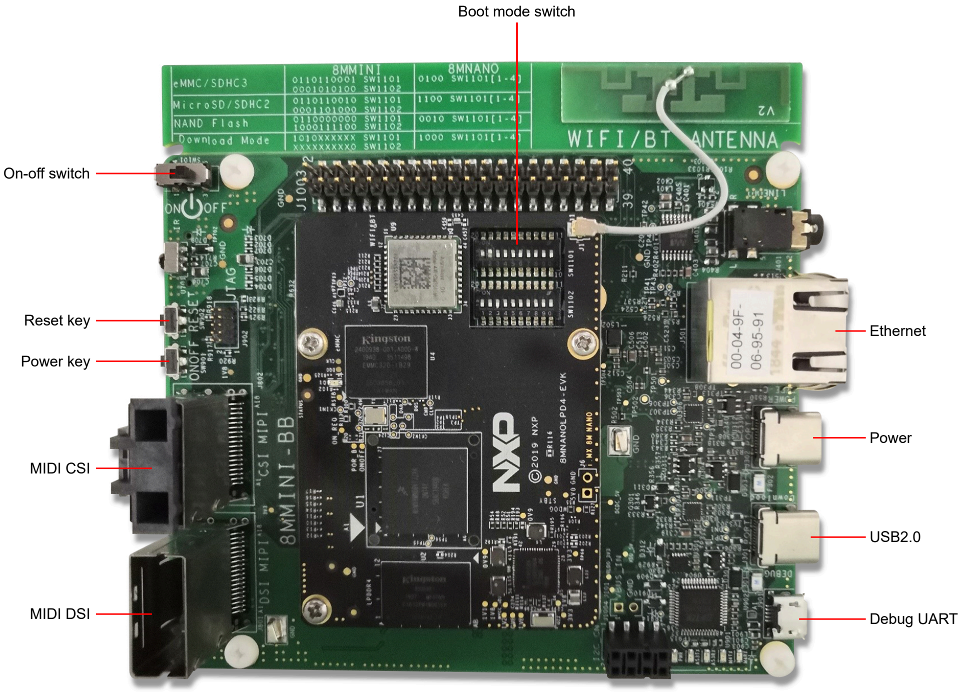
3.5 Board reboot

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.



aaa-057636

Figure 7. i.MX 8M Nano EVK board

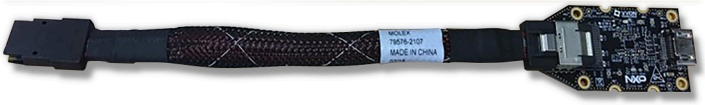


Figure 8. i.MX mini SAS cable with DSI-to-HDMI adapter

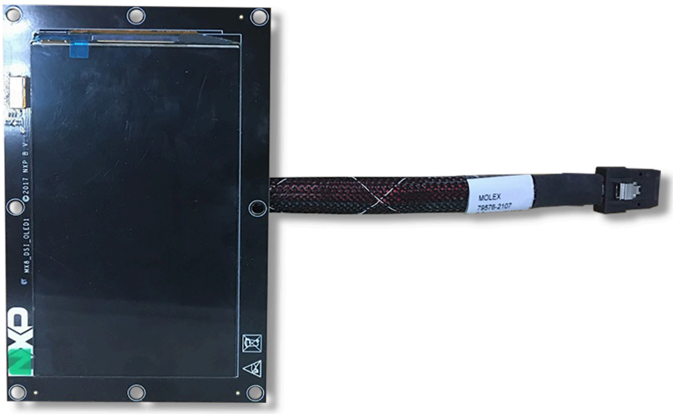


Figure 9. MX8-DSI-OLED1 MIPI panel

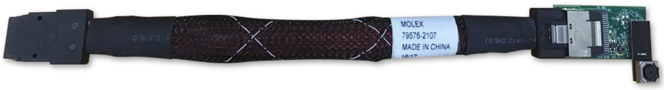


Figure 10. OV5640 CSI MIPI camera

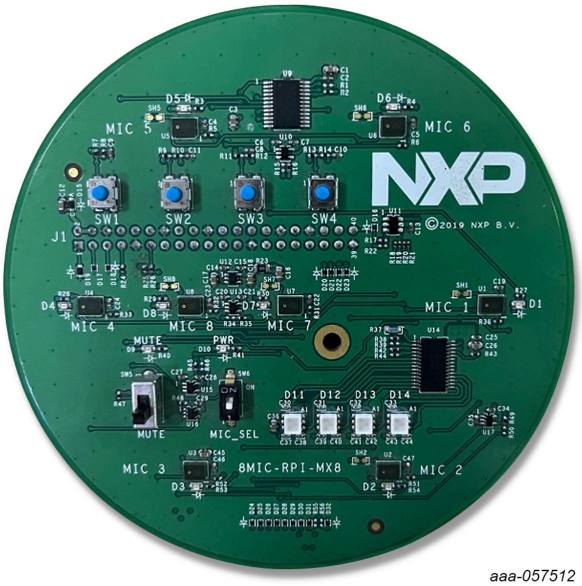


Figure 11. i.MX 8MIC PDM Microphone board

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-15.0.0_1.2.0_image_8mnevk.tar.gz`.

Table 3. Board images

Image name	Download target	Description
`spl-imx8mn-dual.bin`	32 kB offset of MMC.	Secondary program loader image without Trusty related configurations for the i.MX 8M Nano EVK LPDDR4 board.
`spl-imx8mn-trusty-dual.bin`	32 kB offset of MMC.	Secondary program loader image with Trusty related configurations for the i.MX 8M Nano EVK LPDDR4 board.
`spl-imx8mn-trusty-secure-unlock-dual.bin`	32 kB offset of MMC.	Secondary program loader image with Trusty and secure unlock related configurations for the i.MX 8M Nano EVK LPDDR4 board.
`bootloader-imx8mn-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8M Nano EVK LPDDR4 board.
`bootloader-imx8mn-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8M Nano EVK LPDDR4 board.
`bootloader-imx8mn-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partition.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of the secure unlock mechanism for the i.MX 8M Nano EVK LPDDR4 board.
`u-boot-imx8mn.imx`	32 kB offset of MMC.	An image containing U-Boot and ATF for the i.MX 8M Nano EVK LPDDR4 board.
`u-boot-imx8mn-ddr4.imx`	32 kB offset of MMC.	An image containing U-Boot and ATF for the i.MX 8M Nano EVK DDR4 board.
`u-boot-imx8mn-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8M Nano EVK LPDDR4 board. It is not flashed to MMC.
`u-boot-imx8mn-ddr4-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for i.MX 8M Nano EVK DDR4 board. It is not flashed to MMC.
`imx8mn_mcu_demo.img`	5120 kB offset of MMC.	The MCU demonstration image for the i.MX 8M Nano EVK board.
`partition-table.img`	0 kB offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 kB offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 kB offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 kB offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image, generic ramdisk.

Table 3. Board images*...continued*

Image name	Download target	Description
`boot-imx.img`	`boot_a` and `boot_b` partitions.	Boot image built with i.MX kernel tree for debugging.
`init_boot.img`	`init_boot_a` and `init_boot_b` partitions.	Contains generic ramdisk.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions.	Used in the VTS test with GSI system image.
`vbmeta-imx8mn.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output on the i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-rpmsg.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output and MCU image on the i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support RM67199 MIPI panel output on the i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support RM67191 MIPI panel output on the i.MX 8M Nano EVK LPDDR4 board.
`vbmeta-imx8mn-ddr4.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output on the i.MX 8M Nano EVK DDR4 board.
`vbmeta-imx8mn-ddr4-rpmsg.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output and MCU image on the i.MX 8M Nano EVK DDR4 board.
`vbmeta-imx8mn-ddr4-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support RM67199 MIPI panel output on the i.MX 8M Nano EVK DDR4 board.
`vbmeta-imx8mn-ddr4-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support RM67191 MIPI panel output on the i.MX 8M Nano EVK DDR4 board.
`system.img`	Logical partitions `system_a` and `system_b` in super partition.	System image.
`system_dlkm.img`	Logical partitions `system_dlkm_a` and `system_dlkm_b` in super partition.	System dynamically loadable kernel module image.
`system_ext.img`	Logical partitions `system_ext_a` and `system_ext_b` in super partition.	System extension image.
`vendor.img`	Logical partitions `vendor_a` and `vendor_b` in super partition.	Vendor image.
`vendor_dlkm.img`	Logical partitions `vendor_dlkm_a` and `vendor_dlkm_b` in super partition.	Vendor dynamically loadable kernel module image.

Table 3. Board images*...continued*

Image name	Download target	Description
`product.img`	Logical partitions `product_a` and `product_b` in super partition.	Product image.
`super.img`	Super partition.	Super image, which contains images for logical partitions.
`dtbo-imx8mn.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support MIPI-to-HDMI output on the i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-rpmsg.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support MIPI-to-HDMI output and MCU image on the i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support RM67199 MIPI panel output on the i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support RM67191 MIPI panel output on the i.MX 8M Nano EVK LPDDR4 board.
`dtbo-imx8mn-ddr4.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support MIPI-to-HDMI output on the i.MX 8M Nano EVK DDR4 board.
`dtbo-imx8mn-ddr4-rpmsg.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support MIPI-to-HDMI output and MCU image on the i.MX 8M Nano EVK DDR4 board.
`dtbo-imx8mn-ddr4-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support RM67199 MIPI panel output on the i.MX 8M Nano EVK DDR4 board.
`dtbo-imx8mn-ddr4-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support RM67191 MIPI panel output on the i.MX 8M Nano EVK DDR4 board.
`rpmb_key_test.bin`	N/A	Prebuilt test RPMB key. It can be used to set the RPMB key as fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set the RPMB key.
`testkey_public_rsa4096.bin`	N/A	Prebuilt AVB public key. It is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.

Note: *In this document, logical partition is also called dynamic partition. For more Android dynamic partitions, see https://source.android.com/docs/core/ota/dynamic_partitions.*

*With virtual A/B enabled, the `super.img` file generated by the Android build system only has slot a dynamic partitions filled with images. Slot b dynamic partitions like `system_b` is **0** in size. Therefore, the system cannot boot up from slot b with prebuilt images. To boot from slot b, take either of the following actions:*

- *Apply an OTA update.*
- *Boot the board into fastbootd mode, and flash the images to slot b logical partitions. Set slot b as the active slot.*

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 1.5.179 version. Download the 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 the 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-15.0.0_1.2.0_image_8mnevk.tar.gz`. The package contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Nano board, related options are described as follows.

Table 4. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8M Nano EVK, it should be `imx8mn`. This option is **mandatory**.
`-a`	Only flashes slot a physical partitions and other partitions do not have slot a/b. If this option and the `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b physical partitions and other partitions do not have slot a/b. If this option and the `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8M Nano EVK, it can be followed with "28". If this option is not used, default `partition-table.img` or `partition-table-dual.img` is flashed.
`-m`	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader images with `uboot_feature` in their names. For i.MX 8M Nano EVK LPDDR4 board, it can be `dual`, `trusty-dual` and `trusty-secure-unlock-dual`. For i.MX 8M Nano EVK DDR4, it should be `ddr4`. If this option is not used, the default `u-boot-imx8mn.imx` is flashed.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8M Nano EVK LPDDR4 board, it can be `rpmsg`, `mipi-panel`, `mipi-panel-rm67191`. For i.MX 8M Nano EVK DDR4 board, it can be `ddr4`, `ddr4-rpmsg`, `ddr4-mipi-panel`, or `ddr4-mipi-panel-rm67191`. If this option is not used, the default `dtbo-imx8mn.img` and `vbmeta-imx8mn.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8M Nano EVK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.

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

Option	Description
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and 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 the 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 the 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:

- *`-b` is not suggested to be used, because virtual A/B is enbaled and the system cannot boot up from slob b with the prebuilt images.*
- *`-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 (UG10158).*
  - *To test MIPI-DSI to HDMI output, it does not need to use the `-d` option.*
  - *To test MIPI-DSI to HDMI output and MCU image, execute the tool with `-d rpmsg`.*
  - *To test RM67199 MIPI panel output, execute the tool with `-d mipi-panel`.*
  - *To test RM67191 MIPI panel output, execute the tool with `-d mipi-panel-rm67191`.*
- *For i.MX 8M Nano EVK DDR4 board:*
  - *To test MIPI-DSI to HDMI output, execute the tool with `-u ddr4 -d ddr4`.*
  - *To test DSI to HDMI output and MCU image, execute the wool with `-u ddr4 -d ddr4-rpmsg`.*
  - *To test RM67199 MIPI panel output, execute the tool with `-u ddr4 -d ddr4-mipi-panel`.*
  - *To test RM67191 MIPI panel output, execute the tool with `-u ddr4 -d ddr4-mipi-panel-rm67191`.*
- *If `uuu_imx_android_flash.bat` is used to flash images on a remote server through Samba, 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.
   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 the SD card.

4.4 Booting

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

Some specific U-Boot enviroment variables may need to be set in some boot scenarios to boot the MCU image or to get better display effect.

4.4.1 Booting with an MCU image

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

```
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
```

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.

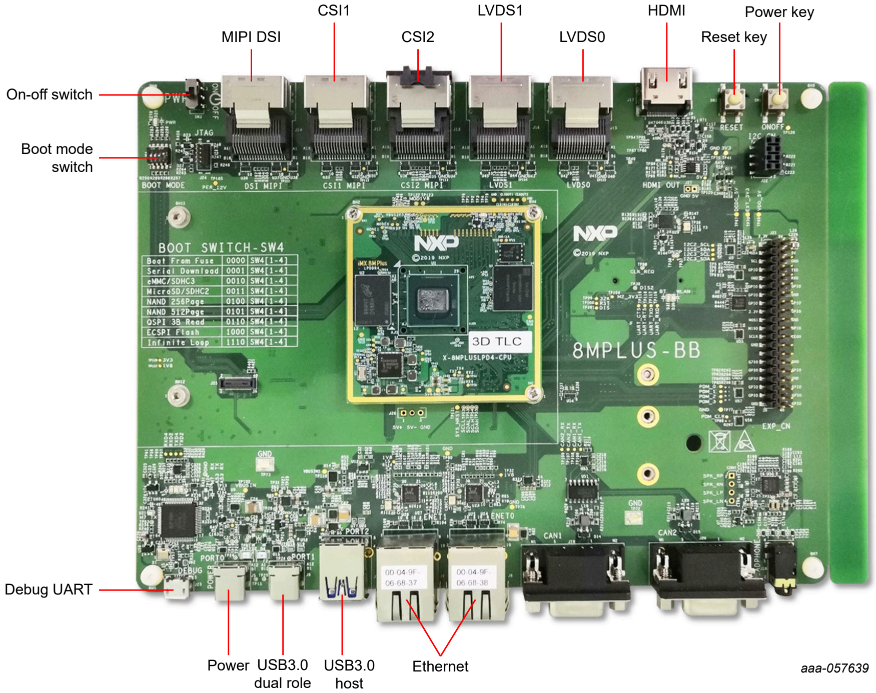


Figure 12. i.MX 8M Plus EVK board

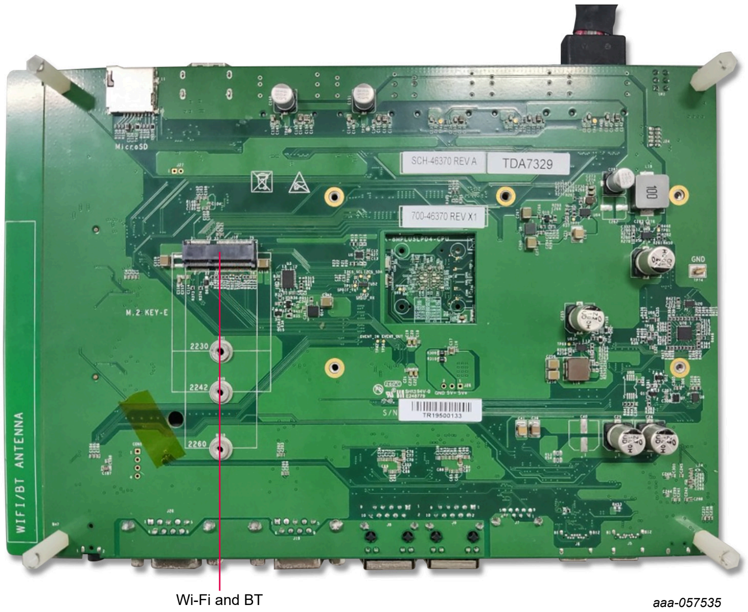


Figure 13. i.MX 8M Plus EVK board back view

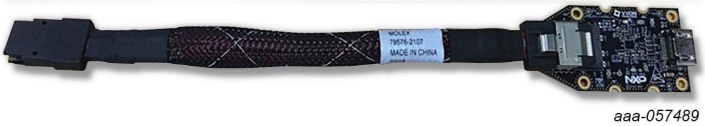


Figure 14. i.MX mini SAS cable with DSI-to-HDMI adapter

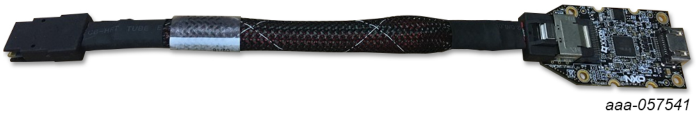


Figure 15. i.MX mini SAS cable with LVDS-to-HDMI adapter

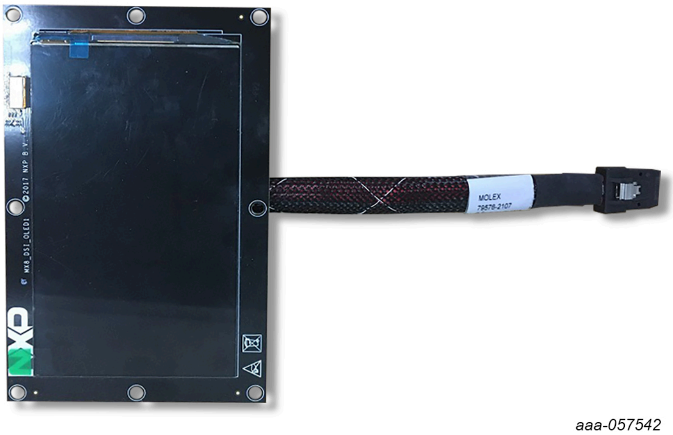


Figure 16. MX8-DSI-OLED1 MIPI panel

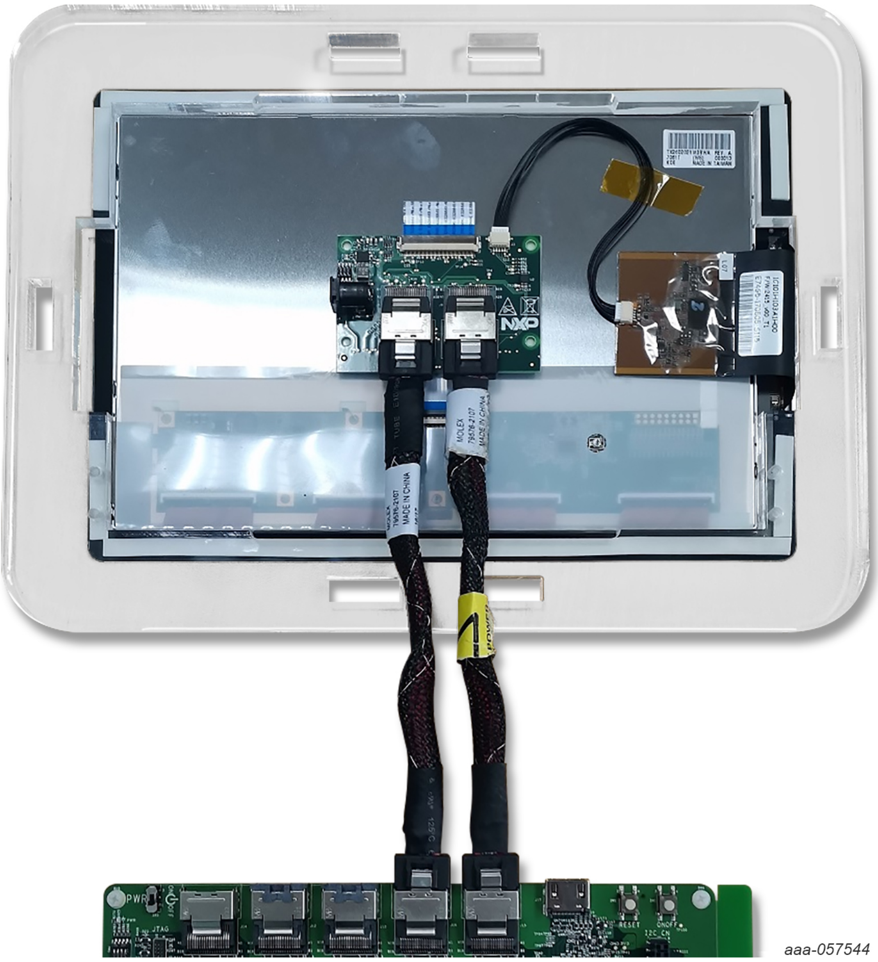
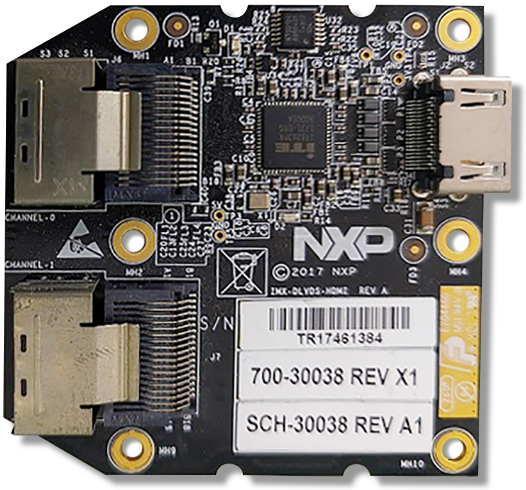
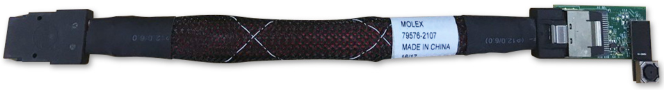


Figure 17. i.MX LVDS panel



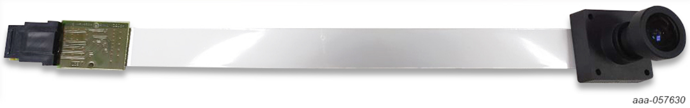
aaa-057633

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



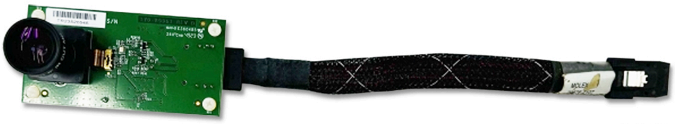
aaa-058037

Figure 19. OV5640 CSI MIPI camera



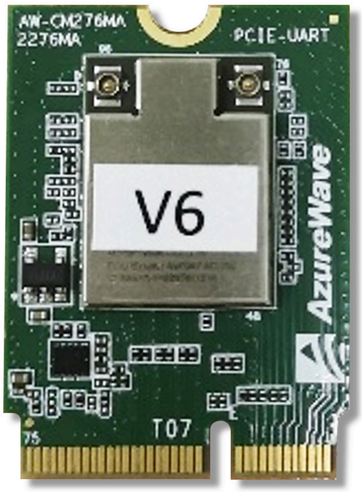
aaa-057630

Figure 20. Basler CSI MIPI camera



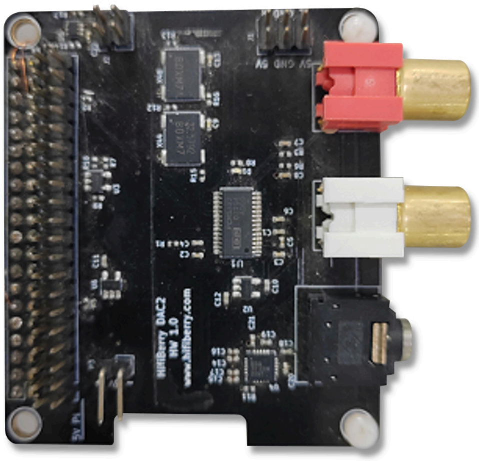
aaa-057540

Figure 21. OS08A20 CSI MIPI camera



aaa-057522

Figure 22. PCIE8997 (AW-CM276)



aaa-057502

Figure 23. i.MX 8M Plus EVK PCM512 audio board (for powersave image)

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.*
- *The Basler sensor functionality is no longer maintained due to lack of continued support from Basler.*

5.2 Board images

The table below describes the location in the board partitions of the software images in `android-15.0.0_1.2.0_image_8mpevk.tar.gz`.

Table 5. Board images

Image name	Download target	Description
`spl-imx8mp-dual.bin`	0 kB offset of eMMC `boot0` partition or 32 kB offset of SD card.	Secondary program loader image without Trusty related configurations for the i.MX 8M Plus EVK board.
`spl-imx8mp-trusty-dual.bin`	0 kB offset of eMMC `boot0` partition.	Secondary program loader image with Trusty related configurations for the i.MX 8M Plus EVK board.
`spl-imx8mp-trusty-secure-unlock-dual.bin`	0 kB offset of eMMC `boot0` partition.	Secondary program loader image with Trusty and secure unlock related configurations for the i.MX 8M Plus EVK LPDDR4 board.
`bootloader-imx8mp-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8M Plus EVK board.
`bootloader-imx8mp-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8M Plus EVK board.
`bootloader-imx8mp-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of the secure unlock mechanism for the i.MX 8M Plus EVK board.
`u-boot-imx8mp.imx`	0 kB offset of eMMC `boot0` partition or 32 kB offset of SD card.	An image containing U-Boot and ATF for the i.MX 8M Plus EVK board.
`u-boot-imx8mp-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8M Plus EVK board. It is not flashed to MMC.
`imx8mp_mcu_demo.img`	5120 kB offset of eMMC user partition or SD card.	The MCU image for the i.MX 8M Plus EVK board.
`partition-table.img`	0 kB offset of eMMC user partition or SD card.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 kB offset of eMMC user partition or SD card.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 13 GB.

Table 5. Board images*...continued*

Image name	Download target	Description
partition-table-28GB.img	0 kB offset of eMMC user partition or SD card.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 28 GB.
partition-table-28GB-dual.img	0 kB offset of eMMC user partition or SD card.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 28 GB.
boot.img	boot_a and boot_b partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image and generic ramdisk.
boot-imx.img	boot_a and boot_b partitions.	Boot image built with the i.MX kernel tree for debugging.
init_boot.img	init_boot_a and init_boot_b partitions.	Contains generic ramdisk.
vendor_boot.img	vendor_boot_a and vendor_boot_b partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
vendor_boot-debug.img	vendor_boot_a and vendor_boot_b partitions.	Used in VTS test with the GSI system image.
vbmeta-imx8mp.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support two OS08A20 cameras plug-in CSI-1 and CSI-2 ports.
vbmeta-imx8mp-basler-ov5640.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support Basler camera plug-in CSI-1 port and OV5640 camera plug-in CSI-2 port.
vbmeta-imx8mp-dual-basler.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support Basler + Basler camera.
vbmeta-imx8mp-os08a20-ov5640.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support OS08A20 (CSI-1) + OV5640 (CSI-2) camera.
vbmeta-imx8mp-os08a20.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support only OS08A20 camera plug-in CSI-1 slot.
vbmeta-imx8mp-basler.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support only Basler camera plug-in CSI-1 slot.
vbmeta-imx8mp-ov5640.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support only OV5640 CSI MIPI camera plug-in CSI-1 slot.
vbmeta-imx8mp-lvds-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support LVDS panel output.
vbmeta-imx8mp-lvds.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support dual-channel LVDS-to-HDMI output.
vbmeta-imx8mp-mipi-panel.img	vbmeta_a and vbmeta_b partitions	Android Verify Boot metadata image to support RM67199 MIPI panel output.
vbmeta-imx8mp-mipi-panel-rm67191.img	vbmeta_a and vbmeta_b partitions	Android Verify Boot metadata image to support RM67191 MIPI panel output.
vbmeta-imx8mp-rpmsg.img	vbmeta_a and vbmeta_b partitions	Android Verify Boot metadata image to support MIPI-to-HDMI output and MCU image.

Table 5. Board images*...continued*

Image name	Download target	Description
vbmeta-imx8mp-sof.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support MIPI-to-HDMI output, and support the Sound Open Firmware audio output.
system.img	Logical partitions system_a and system_b in super partition	System image.
system_dlkm.img	Logical partitions system_dlkm_a and system_dlkm_b in super partition.	System dynamically loadable kernel module image.
system_ext.img	Logical partitions system_ext_a and system_ext_b in super partition.	System extension image.
vendor.img	Logical partitions vendor_a and vendor_b in super partition.	Vendor image
vendor_dlkm.img	Logical partitions vendor_dlkm_a and vendor_dlkm_b in super partition.	Vendor dynamically loadable kernel module image.
product.img	Logical partitions product_a and product_b in super partition.	Product image.
super.img	Super partition.	Super image, which contains images for logical partitions.
dtbo-imx8mp.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output, and two OS08A20 cameras plug-in CSI-1 and CSI-2 ports.
dtbo-imx8mp-basler-ov5640.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output, and support Basler camera plug-in CSI-1 port and OV5640 camera plug-in CSI-2 port.
dtbo-imx8mp-basler.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output, and support only Basler camera plug-in CSI-1 slot.
dtbo-imx8mp-ov5640.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output, and support only OV5640 CSI MIPI camera plug-in CSI-1 slot.
dtbo-imx8mp-dual-basler.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output, and support Basler + Basler camera.
dtbo-imx8mp-os08a20-ov5640.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output, and support OS08A20 (CSI-1) + OV5640 (CSI-2) camera.
dtbo-imx8mp-os08a20.img	dtbo_a and dtbo_b partitions.	Device Tree image to support MIPI-to-HDMI output, and support only OS08A20 camera plug-in CSI-1 slot.
dtbo-imx8mp-lvds-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support LVDS panel output.
dtbo-imx8mp-lvds.img	dtbo_a and dtbo_b partitions.	Device Tree image to support dual-channel LVDS to HDMI output.
dtbo-imx8mp-mipi-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support RM67199 MIPI panel output.
dtbo-imx8mp-mipi-panel-rm67191.img	dtbo_a and dtbo_b partitions.	Device Tree image to support RM67191 MIPI panel output.

Table 5. Board images*...continued*

Image name	Download target	Description
`dtbo-imx8mp-rpmsg.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support MIPI-to-HDMI output and MCU image.
`dtbo-imx8mp-sof.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the Sound Open Firmware audio output.
`rpmb_key_test.bin`	N/A	Prebuilt test RPMB key, which can be used to set the RPMB key as fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set the RPMB key.
`testkey_public_rsa4096.bin`	N/A	Prebuilt AVB public key, which is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.

Note: *In this document, logical partition is also called dynamic partition. For more Android dynamic partitions, see https://source.android.com/docs/core/ota/dynamic_partitions.*

*With virtual A/B enabled, the `super.img` file generated by the Android build system only has slot a dynamic partitions filled with images. Slot b dynamic partitions like `system_b` is **0** in size. Therefore, the system cannot boot up from slot b with prebuilt images. To boot from slot b, take either of the following actions:*

- *Apply an OTA update.*
- *Boot the board into fastbootd mode, and flash the images to slot b logical partitions. Set slot b as the active slot.*

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 1.5.179 version. Download the 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-15.0.0_1.2.0_image_8mpevk.tar.gz`. The package contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Plus board, related options are described as follows.

Table 6. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8M Plus EVK, it should be `imx8mp`. This option is **mandatory**.
`-a`	Only flashes slot a physical partitions and other partitions do not have slot a/b. If this option and the `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b physical partitions and other partitions do not have slot a/b. If this option and the `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8M Plus EVK, it can be followed with "28". If this option is not used, default `partition-table.img` or `partition-table-dual.img` is flashed.
`-m`	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&Bootloader images with `uboot_feature` in their names. For i.MX 8M Plus EVK board, it can be `dual`, `trusty-dual`, `trusty-secure-unlock-dual`. If this option is not used, the default `u-boot-imx8mp.imx` is flashed.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8M Plus EVK with Non-Rev. B4 BB, it can be `basler-ov5640`, `basler`, `ov5640`, `lvds-panel`, `lvds`, `mipi-panel`, `rpmsg`, `dual-basler`, `os08a20-ov5640`, `os08a20`, `mipi-panel-rm67191`, or `sof`. If this option is not used, the default `dtbo-imx8mp.img` and `vbmeta-imx8mp.img` are flashed.
`-e`	Erases the user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8M Plus EVK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and 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.

Note: *`-m` should be used together with `-d rpmsg`.*

- On the 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 the 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:

- *`-b` is not suggested to be used, because virtual A/B is enbaled and the system cannot boot up from slob b with the prebuilt images.*
- *`-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 (UG10158).*
- *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 (UG10156) for more steps to make cameras work.*
  - *To test single Basler camera, execute the tool with `-d basler`. See the Android User's Guide (UG10156) 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 (UG10156) 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 (UG10156) 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 (UG10156) 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 (UG10156) 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 (UG10156) 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.

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 boot device to eMMC or SD card.
   - Change SW4 to switch the board back to 0011 (SD boot mode, from 1-4 bit).
   - Change SW4 to switch the board back to 0010 (eMMC boot mode, from 1-4 bit).

5.4 Booting

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

Some specific U-Boot enviroment variables may need to be set in some boot scenarios to boot the MCU image or to get better display effect.

5.4.1 Booting with audio playback based on Cortex-M7 FreeRTOS

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

```
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
```

5.4.2 Booting with a single display: HDMI 4K display

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

```
U-Boot > setenv append_bootargs androidboot.displaymode=4kp30
 androidboot.lcd_density=480
U-Boot > saveenv
```

Note:

If other boot arguments need to be appended to the U-Boot environment variable `bootargs`, set them together in one `setenv append_bootargs` command.

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 WEVK board.

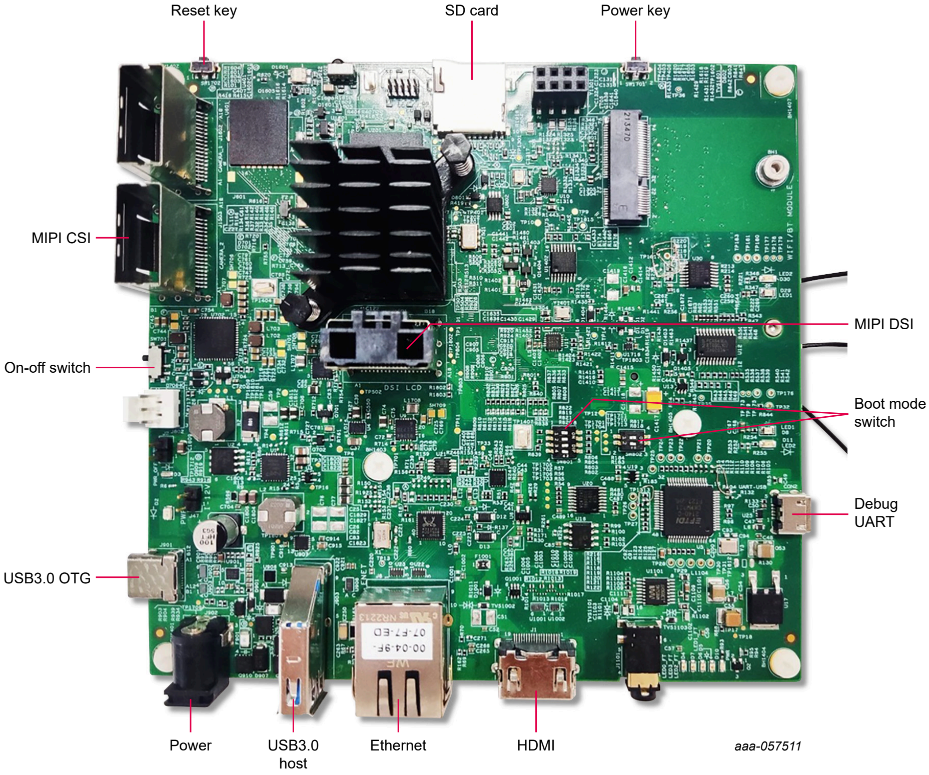


Figure 24. i.MX 8M Quad WEVK board

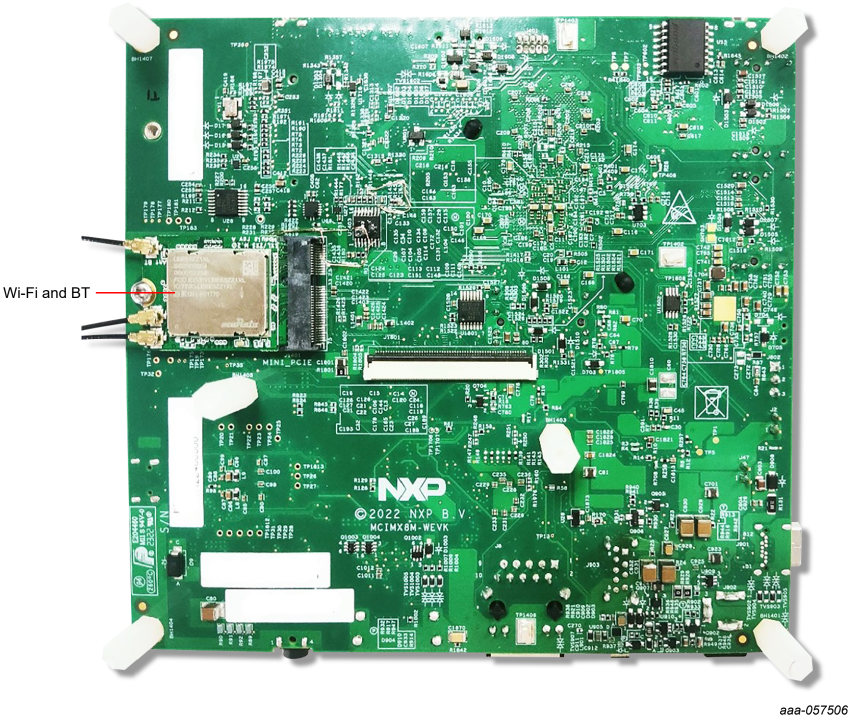


Figure 25. i.MX 8M Quad WEVK board back view

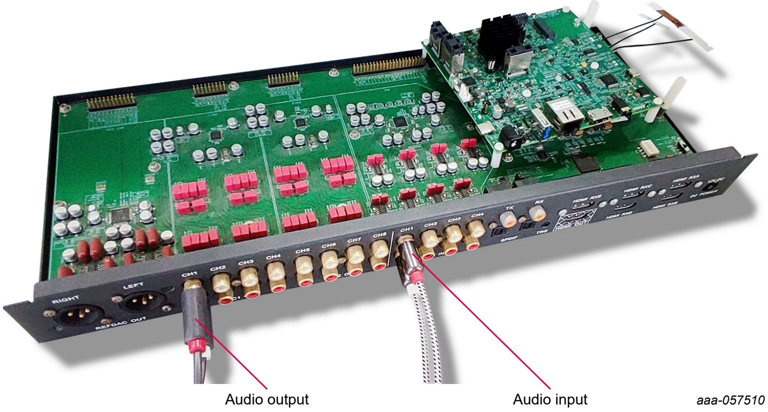


Figure 26. i.MX 8M Quad WEVK with audio board

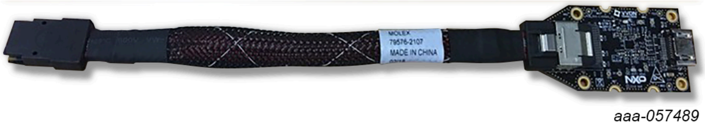


Figure 27. i.MX mini SAS cable with DSI-to-HDMI adapter

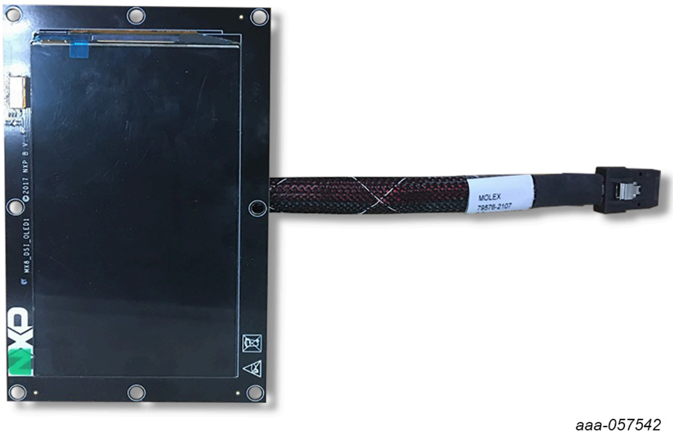


Figure 28. MX8-DSI-OLED1 MIPI panel

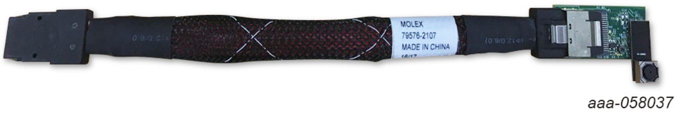


Figure 29. OV5640 CSI MIPI camera

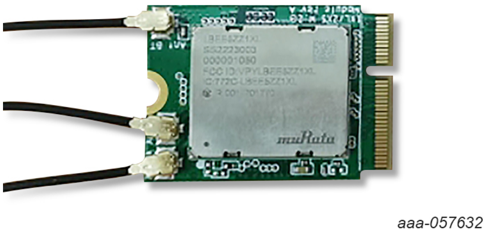


Figure 30. PCIE9098 (Murata LBEE5ZZ1XL)

Note:

- *i.MX 8M Quad WEVK 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 and Bluetooth 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-15.0.0_1.2.0_image_8mqwevk.tar.gz`.

Table 7. Board images

Image name	Download target	Description
`spl-imx8mq-dual.bin`	33 kB offset of MMC.	Secondary program loader image without Trusty related configurations for the i.MX 8M Quad EVK board.
`spl-imx8mq-trusty-dual.bin`	33 kB offset of MMC.	Secondary program loader image with Trusty related configurations for the i.MX 8M Quad EVK board.
`spl-imx8mq-trusty-secure-unlock-dual.bin`	33 kB offset of MMC.	Secondary program loader image with Trusty and secure unlock related configurations for the i.MX 8M Quad EVK LPDDR4 board.
`spl-imx8mq-wevk-dual.bin`	33 kB offset of MMC.	Secondary program loader image without Trusty related configurations for the i.MX 8M Quad WEVK board.
`spl-imx8mq-trusty-wevk-dual.bin`	33 kB offset of MMC.	Secondary program loader image with Trusty related configurations for the i.MX 8M Quad WEVK board.
`spl-imx8mq-trusty-secure-unlock-wevk-dual.bin`	33 kB offset of MMC.	Secondary program loader image with Trusty and secure unlock related configurations for the i.MX 8M Quad WEVK LPDDR4 board.
`bootloader-imx8mq-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8M Quad EVK board.
`bootloader-imx8mq-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8M Quad EVK board.
`bootloader-imx8mq-trusty-secure-unlock-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of the secure unlock mechanism for the i.MX 8M Quad EVK LPDDR4 board.
`bootloader-imx8mq-wevk-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for i.MX 8M Quad WEVK board.
`bootloader-imx8mq-trusty-wevk-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8M Quad WEVK board.
`bootloader-imx8mq-trusty-secure-unlock-wevk-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of the secure unlock mechanism for the i.MX 8M Quad WEVK LPDDR4 board.
`u-boot-imx8mq.imx`	33 kB offset of MMC.	An image containing U-Boot and ATF for the i.MX 8M Quad EVK board.
`u-boot-imx8mq-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8M Quad EVK board. It is not flashed to MMC.

Table 7. Board images*...continued*

Image name	Download target	Description
`u-boot-imx8mq-wevk.imx`	33 kB offset of MMC.	An image containing U-Boot and ATF for i.MX 8M Quad WEVK board.
`u-boot-imx8mq-wevk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for i.MX 8M Quad WEVK board. It is not flashed to MMC.
`partition-table.img`	0 kB offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-dual.img`	0 kB offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 13 GB.
`partition-table-28GB.img`	0 kB offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 28 GB.
`partition-table-28GB-dual.img`	0 kB offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image and generic ramdisk.
`boot-imx.img`	`boot_a` and `boot_b` partitions.	Boot image built with the i.MX kernel tree for debugging.
`init_boot.img`	`init_boot_a` and `init_boot_b` partitions.	Contains generic ramdisk.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions.	Used in VTS test with the GSI system image.
`vbmeta-imx8mq.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the i.MX 8M Quad EVK board HDMI output.
`vbmeta-imx8mq-wevk.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the i.MX 8M Quad WEVK board HDMI output.
`vbmeta-imx8mq-mipi.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the i.MX 8M Quad WEVK/EVK board MIPI-DSI-to-HDMI output.
`vbmeta-imx8mq-dual.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the i.MX 8M Quad WEVK/EVK board HDMI and MIPI-DSI-to-HDMI dual output.
`vbmeta-imx8mq-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the i.MX 8M Quad WEVK/EVK board RM67199 MIPI panel output.
`vbmeta-imx8mq-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the i.MX 8M Quad WEVK/EVK board RM67191 MIPI panel output.
`system.img`	Logical partitions `system_a` and `system_b` in super partition.	System image.
`system_dlkm.img`	Logical partitions `system_dlkm_a` and `system_dlkm_b` in super partition.	System dynamically loadable kernel module image.

Table 7. Board images...*continued*

Image name	Download target	Description
`system_ext.img`	Logical partitions `system_ext_a` and `system_ext_b` in super partition.	System extension image.
`vendor.img`	Logical partitions `vendor_a` and `vendor_b` in super partition.	Vendor image
`vendor_dlkm.img`	Logical partitions `vendor_dlkm_a` and `vendor_dlkm_b` in super partition.	Vendor dynamically loadable kernel module image.
`product.img`	Logical partition `product_a` and `product_b` in super partition.	Product image.
`super.img`	Super partition.	Super image, which contains the images for logical partitions.
`dtbo-imx8mq.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the i.MX 8M Quad EVK board HDMI output.
`dtbo-imx8mq-wevk.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the i.MX 8M Quad WEVK board HDMI output.
`dtbo-imx8mq-mipi.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the i.MX 8M Quad WEVK/EVK board MIPI-DSI-to-HDMI output.
`dtbo-imx8mq-dual.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the i.MX 8M Quad WEVK/EVK board HDMI and MIPI-DSI-to-HDMI dual output.
`dtbo-imx8mq-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the i.MX 8M Quad WEVK/EVK board RM67199 MIPI panel output.
`dtbo-imx8mq-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the i.MX 8M Quad WEVK/EVK board RM67191 MIPI panel output.
`rpmb_key_test.bin`	N/A	Prebuilt test RPMB key, which can be used to set the RPMB key to fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set the RPMB key.
`testkey_public_rsa4096.bin`	N/A	Prebuilt AVB public key, which is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.

Note: *In this document, logical partition is also called dynamic partition. For more Android dynamic partitions, see https://source.android.com/docs/core/ota/dynamic_partitions.*

*With virtual A/B enabled, the `super.img` file generated by the Android build system only has slot a dynamic partitions filled with images. Slot b dynamic partitions like `system_b` is **0** in size. Therefore, the system cannot boot up from slot b with prebuilt images. To boot from slot b, take either of the following actions:*

- *Apply an OTA update.*
- *Boot the board into fastbootd mode, and flash the images to slot b logical partitions. Set slot b as the active slot.*

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 1.5.179 version. Download the 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-15.0.0_1.2.0_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 `imx8mq`. This option is **mandatory**.
`-a`	Only flashes slot a physical partitions and other partitions do not have slot a/b. If this option and the `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b physical partitions and other partitions do not have slot a/b. If this option and the `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8M Quad WEVK, it can be followed with "28". If this option is not used, default `partition-table.img` or `partition-table-dual.img` is flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader images with `uboot_feature` in their names. For i.MX 8M Quad WEVK, it can be `wevk`, `wevk-dual`, `trusty-wevk-dual`, `trusty-secure-unlock-wevk-dual`. For i.MX 8M Quad EVK, it can be `dual`, `trusty-dual`, `trusty-secure-unlock-dual`. If this option is not used, default `u-boot-imx8mq.imx` is flashed.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8M Quad WEVK, it can be `wevk`, `dual`, `mipi-panel`, `mipi-panel-rm67191`, `mipi`. For i.MX 8M Quad EVK, it can be `dual`, `mipi-panel`, `mipi-panel-rm67191`, `mipi`.

Table 8. Options for uuu_imx_android_flash tool*...continued*

Option	Description
	If this option is not used, default `dtbo-imx8mq.img` and `vbmeta-imx8mq.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8M Quad WEVK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no images are flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not executes UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

- On the 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 the 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:

- *`-b` is not suggested to be used, because virtual A/B is enbaled and the system cannot boot up from slob b with the prebuilt images.*
- *`-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 (UG10158).*

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

- *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 (UG10158).*
- *To test feature on i.MX 8M Quad WEVK 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 multiple displays, execute the tool with `-d dual`*
- *To test feature on i.MX 8M Quad EVK 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 multiple displays, 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.

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 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, from 1-4 bit).
   - Change SW801 to switch the board back to 0010 (eMMC boot mode, from 1-4 bit).

6.4 Booting

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

It does not need to set U-Boot enviroment variables for the currently avaiable boot scenarios.

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.

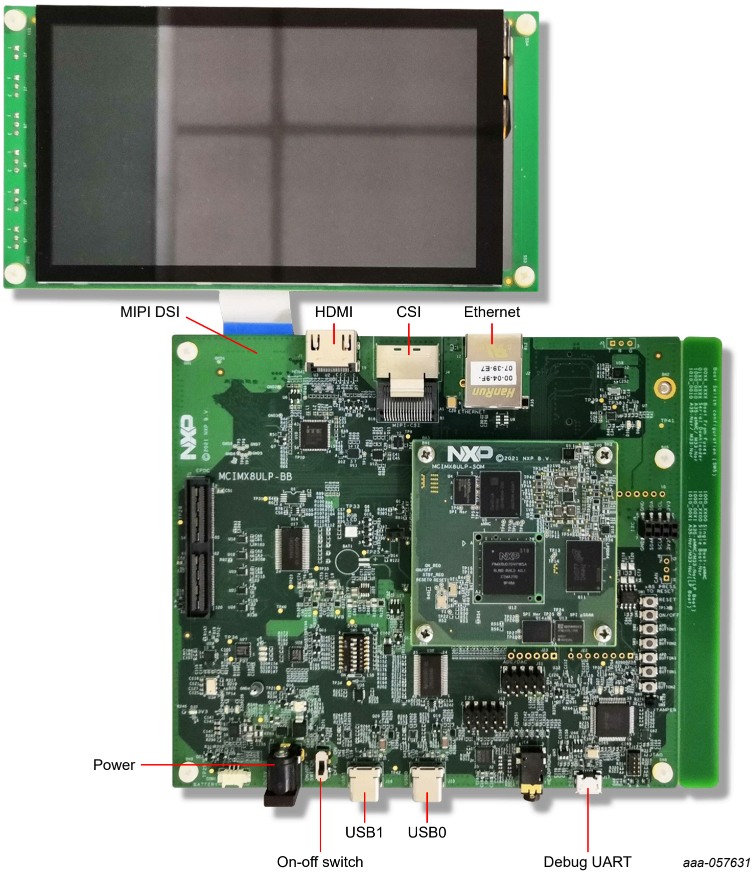


Figure 31. i.MX 8ULP EVK board with MIPI display

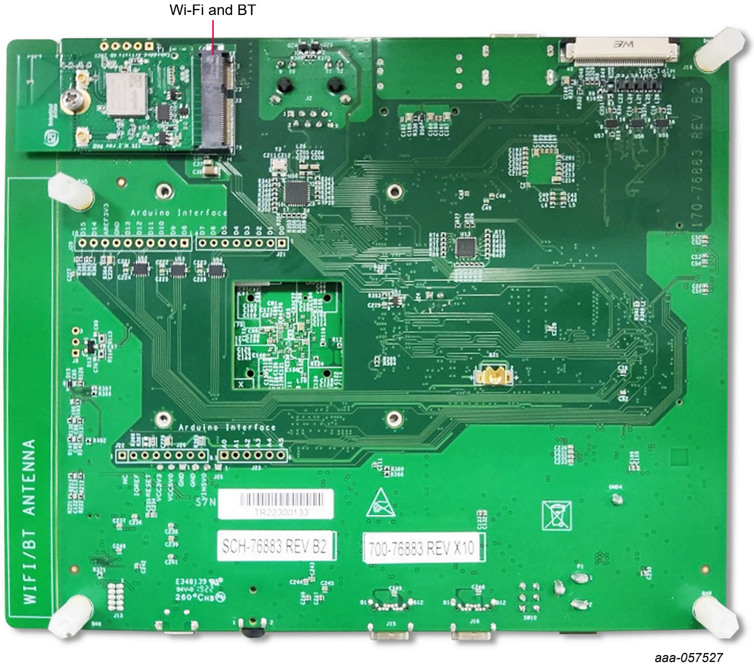


Figure 32. i.MX 8ULP EVK board back view

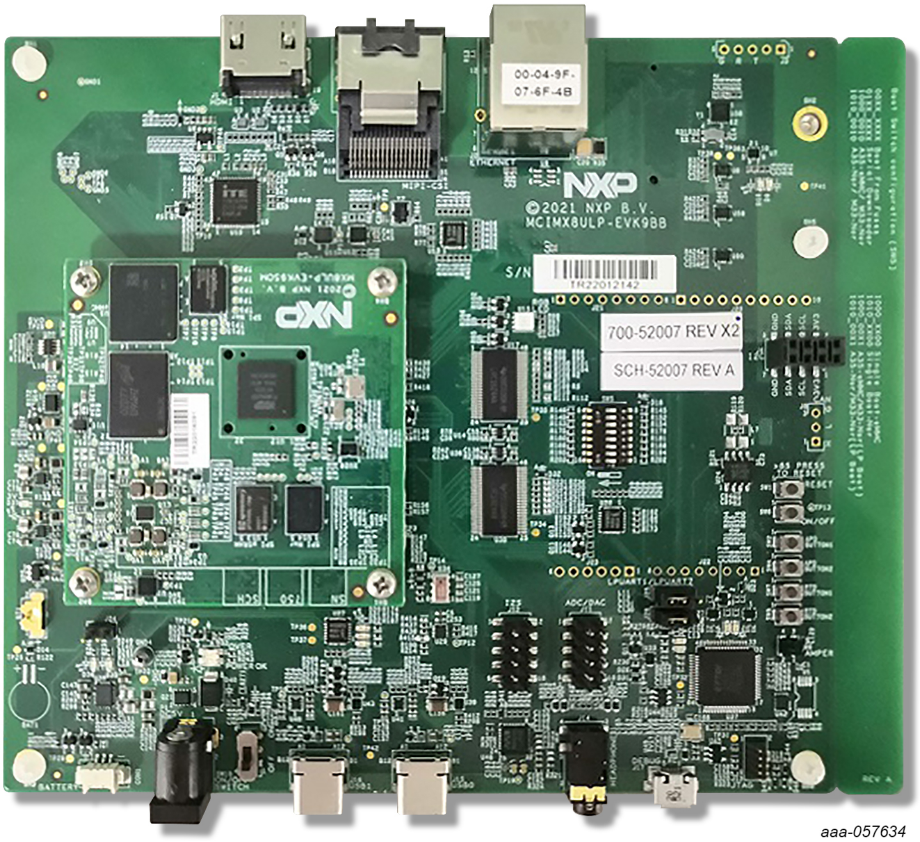


Figure 33. i.MX 8ULP EVK 9x9 board

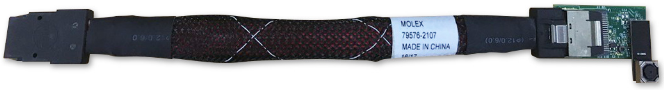
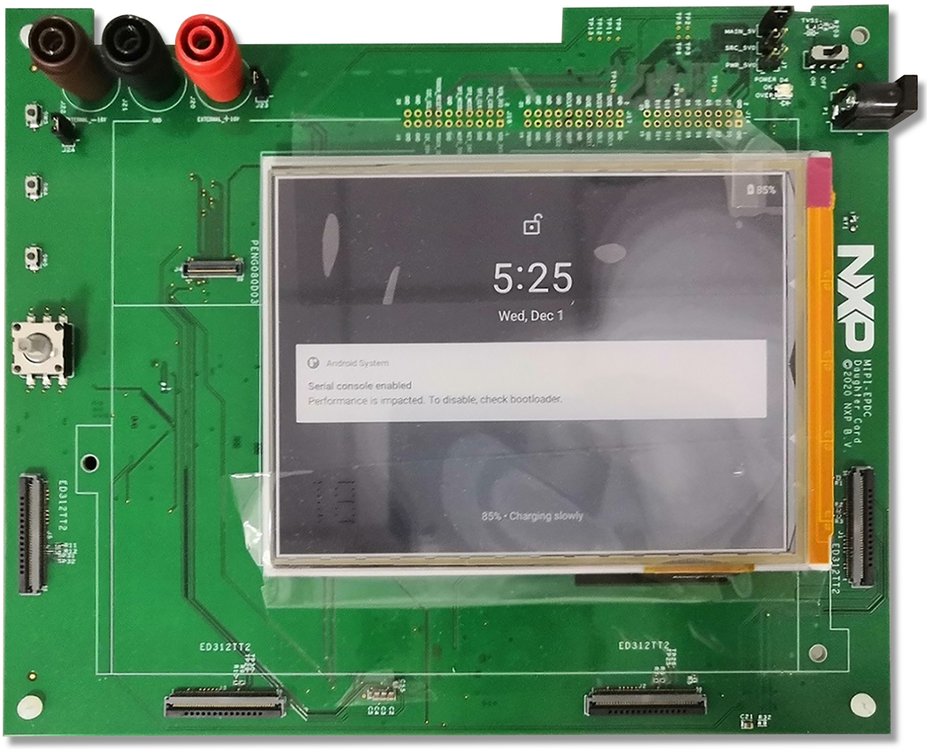
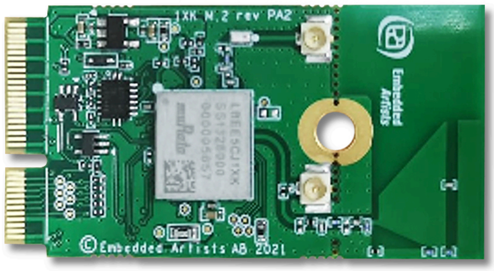


Figure 34. i.MX CSI MIPI camera



aaa-057635

Figure 35. EPDC display



aaa-057515

Figure 36. 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-15.0.0_1.2.0_image_8ulp-evk.tar.gz`.

Table 9. Board images

Image name	Download target	Descriptions
`spl-imx8ulp-dual.bin`	0 offset of eMMC boot partition.	Secondary program loader image without Trusty related configurations for the i.MX 8ULP EVK board.
`spl-imx8ulp-trusty-dual.bin`	0 offset of eMMC boot partition.	Secondary program loader image with Trusty related configurations for the i.MX 8ULP EVK board.
`spl-imx8ulp-9x9-dual.bin`	0 offset of eMMC boot partition.	Secondary program loader image without Trusty related configurations for the i.MX 8ULP EVK 9x9 board.
`spl-imx8ulp-trusty-9x9-dual.bin`	0 offset of eMMC boot partition.	Secondary program loader image with Trusty related configurations for the i.MX 8ULP EVK 9x9 board.
`spl-imx8ulp-trusty-lpa-dual.bin`	0 offset of eMMC boot partition.	Secondary program loader image with Trusty related configurations and Low Power Audio enabled for the i.MX 8ULP EVK board.
`spl-imx8ulp-trusty-dualboot-dual.bin`	0 offset of eMMC boot partition.	Secondary program loader image with Trusty related configurations and dual boot enabled (low-power display) for the i.MX 8ULP EVK board.
`bootloader-imx8ulp-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8ULP EVK board.
`bootloader-imx8ulp-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8ULP EVK board.
`bootloader-imx8ulp-9x9-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8ULP 9x9 EVK board.
`bootloader-imx8ulp-trusty-9x9-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8ULP 9x9 EVK board.
`bootloader-imx8ulp-trusty-lpa-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, Low Power Audio MCU firmware, and Trusty OS. It is for the i.MX 8ULP EVK board.
`bootloader-imx8ulp-trusty-dualboot-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8ULP EVK board.
`u-boot-imx8ulp-9x9.imx`	0 offset of eMMC boot partition.	An image containing U-Boot and ATF for the i.MX 8ULP EVK 9x9 board.
`u-boot-imx8ulp-9x9-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8ULP EVK 9x9 board. It is not flashed to MMC.
`u-boot-imx8ulp.imx`	0 offset of eMMC boot partition.	An image containing U-Boot and ATF for the i.MX 8ULP EVK board.
`u-boot-imx8ulp-evk-uuu.imx`	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8ULP EVK. It is not flashed to MMC.
`imx8ulp_mcu_demo_sf.img`	NOR flash	MCU image that contains a low-power display demo. It is not flashed to the serial NOR flash on the MCU side.

Table 9. Board images*...continued*

Image name	Download target	Descriptions
`partition-table.img`	0 offset of eMMC user data area.	GPT partition table image 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.	GPT partition table image 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.	GPT partition table image 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.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 28 GB.
`boot.img`	`boot_a` and `boot_b` partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image and generic ramdisk.
`boot-imx.img`	`boot_a` and `boot_b` partitions.	Boot image built with the i.MX kernel tree for debugging.
`init_boot.img`	`init_boot_a` and `init_boot_b` partitions.	Contains generic ramdisk.
`vendor_boot.img`	`vendor_boot_a` and `vendor_boot_b` partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
`vendor_boot-debug.img`	`vendor_boot_a` and `vendor_boot_b` partitions.	Used in the VTS test with the GSI system image.
`vbmeta-imx8ulp-9x9-hdmi.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support HDMI output on the i.MX 8ULP EVK 9x9 board.
`vbmeta-imx8ulp-9x9.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support MIPI panel output on the i.MX 8ULP EVK 9x9 board.
`vbmeta-imx8ulp.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support MIPI panel output on the i.MX 8ULP EVK.
`vbmeta-imx8ulp-hdmi.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support HDMI output on the i.MX 8ULP EVK.
`vbmeta-imx8ulp-epdc.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support EPDC output on the i.MX 8ULP EVK.
`vbmeta-imx8ulp-sof.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the Sound Open Firmware audio output on the i.MX 8ULP EVK.
`vbmeta-imx8ulp-lpa.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the Low Power Audio feature on the i.MX 8ULP EVK.
`vbmeta-imx8ulp-lpd.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the Low Power Display feature on the i.MX 8ULP EVK.
`system.img`	Logical partitions `system_a` and `system_b` in super partition.	System image.
`system_dlkm.img`	Logical partitions `system_dlkm_a` and `system_dlkm_b` in super partition.	System dynamically loadable kernel module image.

Table 9. Board images*...continued*

Image name	Download target	Descriptions
`system_ext.img`	Logical partitions `system_ext_a` and `system_ext_b` in super partition.	System extension image.
`vendor.img`	Logical partitions `vendor_a` and `vendor_b` in super partition.	Vendor image.
`vendor_dlkm.img`	Logical partitions `vendor_dlkm_a` and `vendor_dlkm_b` in super partition.	Vendor dynamically loadable kernel module image.
`product.img`	Logical partitions `product_a` and `product_b` in super partition.	Product image.
`super.img`	Super partition.	Super image, which contains images for logical partitions.
`dtbo-imx8ulp-9x9-hdmi.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support HDMI output on the i.MX 8ULP EVK 9x9 board.
`dtbo-imx8ulp-9x9.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support MIPI panel output on the i.MX 8ULP EVK 9x9 board.
`dtbo-imx8ulp.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support MIPI panel output on the i.MX 8ULP EVK.
`dtbo-imx8ulp-hdmi.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support HDMI output on the i.MX 8ULP EVK.
`dtbo-imx8ulp-epdc.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support EPDC output on the i.MX 8ULP EVK.
`dtbo-imx8ulp-sof.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the Sound Open Firmware audio output on the i.MX 8ULP EVK.
`dtbo-imx8ulp-lpa.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support Low Power Audio on the i.MX 8ULP EVK.
`dtbo-imx8ulp-lpd.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the Low Power Display on the i.MX 8ULP EVK.
`rpmb_key_test.bin`	N/A	Prebuilt test RPMB key. It can be used to set the RPMB key as fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set the RPMB key.
`testkey_public_rsa4096.bin`	N/A	Prebuilt AVB public key. It is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.

Note: *In this document, logical partition is also called dynamic partition. For more Android dynamic partitions, see https://source.android.com/docs/core/ota/dynamic_partitions.*

*With virtual A/B enabled, the `super.img` file generated by the Android build system only has slot a dynamic partitions filled with images. Slot b dynamic partitions like `system_b` is **0** in size. Therefore, the system cannot boot up from slot b with prebuilt images. To boot from slot b, take either of the following actions:*

- *Apply an OTA update.*
- *Boot the board into fastbootd mode, and flash the images to slot b logical partitions. Set slot b as the active slot.*

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 1.5.179 version. Download the 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-15.0.0_1.2.0_image_8ulpevk.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX the 8ULP EVK board, related options are described as follows.

Table 10. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8ULP EVK, it should be `imx8ulp`. This option is **mandatory**.
`-a`	Only flashes slot a physical partitions and other partitions do not have slot a/b. If this option and the `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b physical partitions and other partitions do not have slot a/b. If this option and the `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. i.MX 8ULP EVK, it can be followed with "28". If this option is not used, default `partition-table.img` is flashed.
`-m`	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader images with `uboot_feature` in their names. For i.MX 8ULP EVK, it can be `9x9`, `dual`, `9x9-dual`, `trusty-dual`, `trusty-9x9-dual`, `trusty-lpa-dual`, `trusty-dualboot-dual`, `trusty-dualboot-dual`. If this option is not used, the default `u-boot-imx8ulp.imx` is flashed.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8ULP EVK, it can be `9x9`, `9x9-hdmi`, `hdmi`, `epdc`, `sof`, `lpa` and `lpd`. If this option is not used, default `dtbo-imx8ulp.img` and `vbmeta-imx8ulp.img` are flashed.
`-e`	Erases user data after images are flashed.

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

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

- On the 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 the 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:

- *`-b` is not suggested to be used, because virtual A/B is enbaled and the system cannot boot up from slob b with the prebuilt images.*
- *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 (UG10156) 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 (UG10156) 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 (from 1-8 bit) to boot from eMMC.

- Change SW5 to 01000001 (dual boot mode, from 1-8 bit), 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.

Some specific U-Boot enviroment variables may need to be set in some boot scenarios to boot the MCU image or to get better display effect.

7.4.1 Booting with an HDMI display

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

```
U-Boot > setenv append_bootargs androidboot.lcd_density=120
U-Boot > saveenv
```

Note:

If other boot arguments need to be appended to the U-Boot environment variable `bootargs`, set them together in one `setenv append_bootargs` command.

As the camera orientation is set to **90** for the default image (MIPI Panel, 720x1280), for the 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
```

7.5 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.

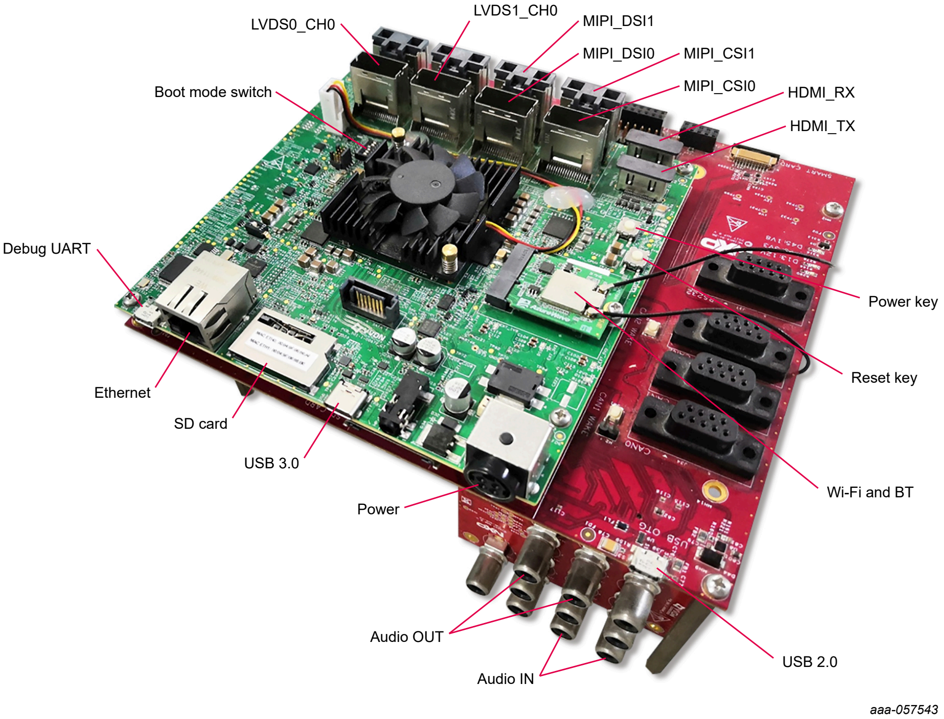


Figure 37. i.MX 8QuadMax MEK board

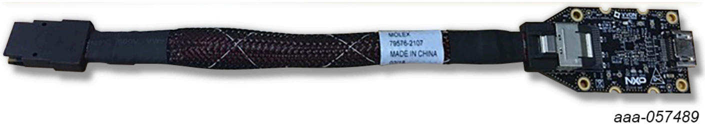


Figure 38. i.MX mini SAS cable with DSI-to-HDMI adapter

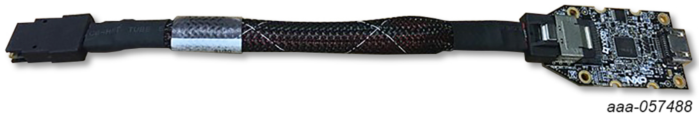


Figure 39. i.MX mini SAS cable with LVDS-to-HDMI adapter

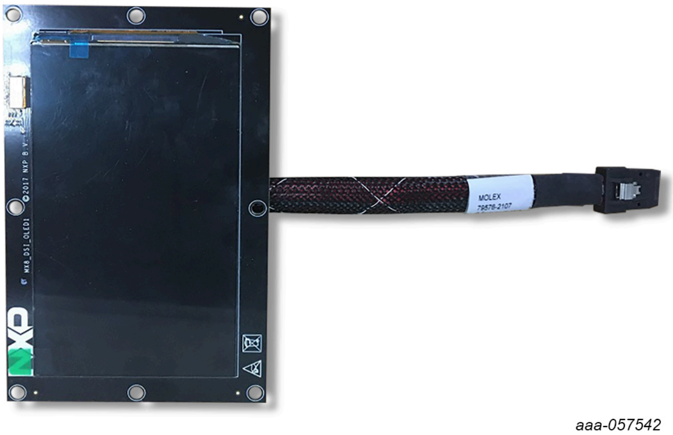


Figure 40. MX8-DSI-OLED1 MIPI panel

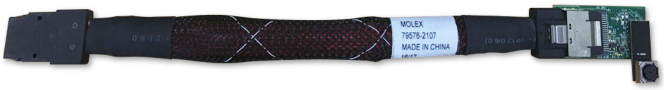


Figure 41. i.MX MIPI camera

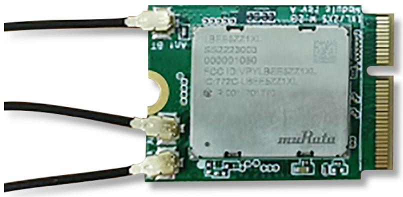


Figure 42. PCIE9098 (Murata LBEE5ZZ1XL)

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-15.0.0_1.2.0_image_8qmek.tar.gz`.

Table 11. Board images

Image name	Download target	Description
`spl-imx8qm-dual.bin`	0 kB offset of eMMC or 32 kB offset of SD card.	Secondary program loader image without Trusty related configurations for the i.MX 8QuadMax MEK board.
`bootloader-imx8qm-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8QuadMax MEK board.
`spl-imx8qm-trusty-dual.bin`	0 kB offset of eMMC.	Secondary program loader image with Trusty related configurations for the i.MX 8QuadMax MEK board.
`bootloader-imx8qm-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8QuadMax MEK board.
`spl-imx8qm-trusty-secure-unlock-dual.bin`	0 kB offset of eMMC.	Secondary program loader image with Trusty and secure unlock related configurations for i.MX 8QuadMax EVK board.

Table 11. Board images*...continued*

Image name	Download target	Description
bootloader-imx8qm-trusty-secure-unlock-dual.img	bootloader_a and bootloader_b partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of the secure unlock mechanism for the i.MX 8QuadMax EVK board.
u-boot-imx8qm.imx	0 kB offset of eMMC and 32 kB offset of SD card.	An image containing U-Boot and ATF for the i.MX 8Quad Max MEK board.
u-boot-imx8qm-hdmi.imx	0 kB offset of eMMC and 32 kB offset of SD card.	An image containing U-Boot and ATF for the i.MX 8Quad Max MEK board to support the physical HDMI display.
u-boot-imx8qm-md.imx	0 kB offset of eMMC and 32 kB offset of SD card.	An image containing U-Boot and ATF for i.MX 8QuadMax MEK board to support multiple displays.
u-boot-imx8qm-mek-uuu.imx	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8QuadMax MEK board. It is not flashed to MMC.
partition-table.img	Programs to the first 17 kB, and then back up to last 17 kB of the boot storage.	GPT partition table image used for single-bootloader condition and for boot storage larger than 13 GB.
partition-table-dual.img	Programs to the first 17 kB, and then back up to last 17 kB of the boot storage.	GPT partition table image used for dual-bootloader condition and for boot storage larger than 13 GB.
partition-table-28GB.img	Programs to the first 17 kB, and then back up to last 17 kB of the boot storage.	GPT partition table image used for single-bootloader condition and for boot storage larger than 28 GB.
partition-table-28GB-dual.img	Programs to the first 17 kB, and then back up to last 17 kB of the boot storage.	GPT partition table image used for dual-bootloader condition and for boot storage larger than 28 GB.
boot.img	boot_a and boot_b partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image and generic ramdisk.
boot-imx.img	boot_a and boot_b partitions.	Boot image built with the i.MX kernel tree for debugging.
init_boot.img	init_boot_a and init_boot_b partitions.	Contains generic ramdisk.
vendor_boot.img	vendor_boot_a and vendor_boot_b partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
vendor_boot-debug.img	vendor_boot_a and vendor_boot_b partitions.	Used in the VTS test with the GSI system image.
vbmeta-imx8qm.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the LVDS-to-HDMI/MIPI-DSI-to-HDMI display.
vbmeta-imx8qm-md.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support multiple displays.
vbmeta-imx8qm-hdmi.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the physical HDMI display.
vbmeta-imx8qm-mipi-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the OLED1 (RM67199) MIPI panel display.
vbmeta-imx8qm-mipi-panel-rm67191.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the OLED1 (RM67191) MIPI panel display.
vbmeta-imx8qm-hdmi-rx.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the HDMI input.

Table 11. Board images*...continued*

Image name	Download target	Description
vbmeta-imx8qm-lvds1-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the LVDS panel display.
vbmeta-imx8qm-sof.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the Sound Open Firmware audio output.
vbmeta-imx8qm-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the LVDS-to-HDMI/MIPI-to-HDMI display on the i.MX 8QuadMax MEK REVD board.
vbmeta-imx8qm-md-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support multiple displays on the i.MX 8QuadMax MEK REVD board.
vbmeta-imx8qm-hdmi-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the physical HDMI display on the i.MX 8QuadMax MEK REVD board.
vbmeta-imx8qm-mipi-panel-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI panel (RM67199) display on the i.MX 8QuadMax MEK REVD board.
vbmeta-imx8qm-mipi-panel-rm67191-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI panel (RM67191) display on the i.MX 8QuadMax MEK REVD board.
vbmeta-imx8qm-hdmi-rx-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the HDMI input on the i.MX 8QuadMax MEK REVD board.
vbmeta-imx8qm-lvds1-panel-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the LVDS panel display on the i.MX 8QuadMax MEK REVD board.
vbmeta-imx8qm-sof-revd.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the Sound Open Firmware audio output on the i.MX 8QuadMax MEK REVD board.
system.img	Logical partitions system_a and system_b in super partition.	System image.
system_dlkm.img	Logical partitions system_dlkm_a and system_dlkm_b in super partition.	System dynamically loadable kernel module image.
system_ext.img	Logical partitions system_ext_a and system_ext_b in super partition.	System extension image.
vendor.img	Logical partitions vendor_a and vendor_b in super partition.	Vendor image.
vendor_dlkm.img	Logical partitions vendor_dlkm_a and vendor_dlkm_b in super partition.	Vendor dynamically loadable kernel module image.
product.img	Logical partitions product_a and product_b in super partition.	Product image.
super.img	Super partition.	Super image, which contains images for logical partitions.
dtbo-imx8qm.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the LVDS-to-HDMI/MIPI-DSI-to-HDMI display.

Table 11. Board images*...continued*

Image name	Download target	Description
dtbo-imx8qm-md.img	dtbo_a and dtbo_b partitions.	Device Tree image to support multiple displays.
dtbo-imx8qm-hdmi.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the physical HDMI display.
dtbo-imx8qm-mipi-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the OLED1 (RM67199) MIPI panel display.
dtbo-imx8qm-mipi-panel-rm67191.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the OLED1 (RM67191) MIPI panel display.
dtbo-imx8qm-hdmi-rx.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the HDMI input.
dtbo-imx8qm-lvds1-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the LVDS panel display.
dtbo-imx8qm-sof.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the Sound Open Firmware audio output.
dtbo-imx8qm-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the i.MX 8QuadMax MEK REVD board.
dtbo-imx8qm-md-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support multiple displays on the i.MX 8QuadMax MEK REVD board.
dtbo-imx8qm-hdmi-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the physical HDMI display on the i.MX 8QuadMax MEK REVD board.
dtbo-imx8qm-mipi-panel-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI panel (RM67199) display on the i.MX 8QuadMax MEK REVD board.
dtbo-imx8qm-mipi-panel-rm67191-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI panel (RM67191) display on the i.MX 8QuadMax MEK REVD board.
dtbo-imx8qm-hdmi-rx-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the HDMI input on the i.MX 8 QuadMax MEK REVD board.
dtbo-imx8qm-lvds1-panel-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the LVDS panel display on the i.MX 8QuadMax MEK REVD board.
dtbo-imx8qm-sof-revd.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the Sound Open Firmware audio output on the i.MX 8QuadMax MEK REVD board.
rpmb_key_test.bin	N/A	Prebuilt test RPMB key. It can be used to set the RPMB key to fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set the RPMB key.
testkey_public_rsa4096.bin	N/A	Prebuilt AVB public key, which is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.

Note: *In this document, logical partition is also called dynamic partition. For more Android dynamic partitions, see https://source.android.com/docs/core/ota/dynamic_partitions.*

*With virtual A/B enabled, the `super.img` file generated by the Android build system only has slot a dynamic partitions filled with images. Slot b dynamic partitions like `system_b` is **0** in size. Therefore, the system cannot boot up from slot b with prebuilt images. To boot from slot b, take either of the following actions:*

- *Apply an OTA update.*

- *Boot the board into fastbootd mode, and flash the images to slot b logical partitions. Set slot b as the active slot.*

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 1.5.179 version. Download the 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-15.0.0_1.2.0_image_8qmek.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8QuadMax board, related options are described as follows.

Table 12. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8QuadMax MEK, it should be `imx8qm`. This option is **mandatory**.
`-a`	Only flashes slot a physical partitions and other partitions do not have slot a/b. If this option and the `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b physical partitions and other partitions do not have slot a/b. If this option and the `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8QuadMax, it can be followed with "28". If this option is not used, default `partition-table.img` is flashed.

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

Option	Description
`-u uboot_feature`	Flashes U-Boot or SPL&Bootloader images with `uboot_feature` in their names. For i.MX 8QuadMax MEK, it can be `hdmi`, `md`, `dual`, `trusty-dual`, or `trusty-secure-unlock-dual`. If this option is not used, the default `u-boot-imx8qm.imx` is flashed.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8 QuadMax MEK, it can be `hdmi-rx`, `hdmi`, `md`, `mipi-panel`, `mipi-panel-rm67191`, `lvds1-panel`, or `sof`. For i.MX 8QuadMax MEK REVD, it can be `revd`, `hdmi-rx-revd`, `hdmi-revd`, `md-revd`, `mipi-panel-revd`, `mipi-panel-rm67191-revd`, `lvds1-panel-revd`, or `sof-revd`. If this option is not used, the default `dtbo-imx8qm.img` and `vbmeta-imx8qm.img` are flashed.
-e	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8QuadMax, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Runs UUU in Daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not executes UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

- On the 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 the 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:

- *`-b` is not suggested to be used, because virtual A/B is enbaled and the system cannot boot up from slob b with the prebuilt images.*
- *`-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 (UG10158).*

- *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 the MIPI panel (RM67199) display, execute the tool with `-d mipi-panel`. The MIPI_DSI0 port is used for display.*
- *To test the MIPI panel (RM67191) display, execute the tool with `-d mipi-panel-rm67191`. The MIPI_DSI0 port is used for display.*
- *To test the 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 the physical HDMI display, execute the tool with `-u hdmi -d hdmi`. The HDMI_TX port is used for display.*
- *To test the 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.*
- *To test the i.MX 8QuadMax MEK REVD board (the board with tags "700-29420 REV D" and "SCH-29420 REV E" on the back of the board), the `-d` option with the parameter containing `revd` needs to be used, the corresponding `-d` option parameters are `revd`, `hdmi-rx-revd`, `hdmi-revd`, `md-revd`, `mipi-panel-revd`, `mipi-panel-rm67191-revd`, `lvds1-panel-revd`, and `sof-revd`.*
- *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.

Note:
If Trusty OS is used, to make video playback work, follow Section "Secure firmware loader" in the i.MX Android Security User's Guide (UG10158) to flash the keys related to the firmware loader to ensure that the VPU firmware can be loaded successfully. The relevant keys can be obtained from the `firmware_test_keys` directory in the directory where the image is located.

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.

It does not need to set U-Boot enviroment variables for the currently avaiable boot scenarios.

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.

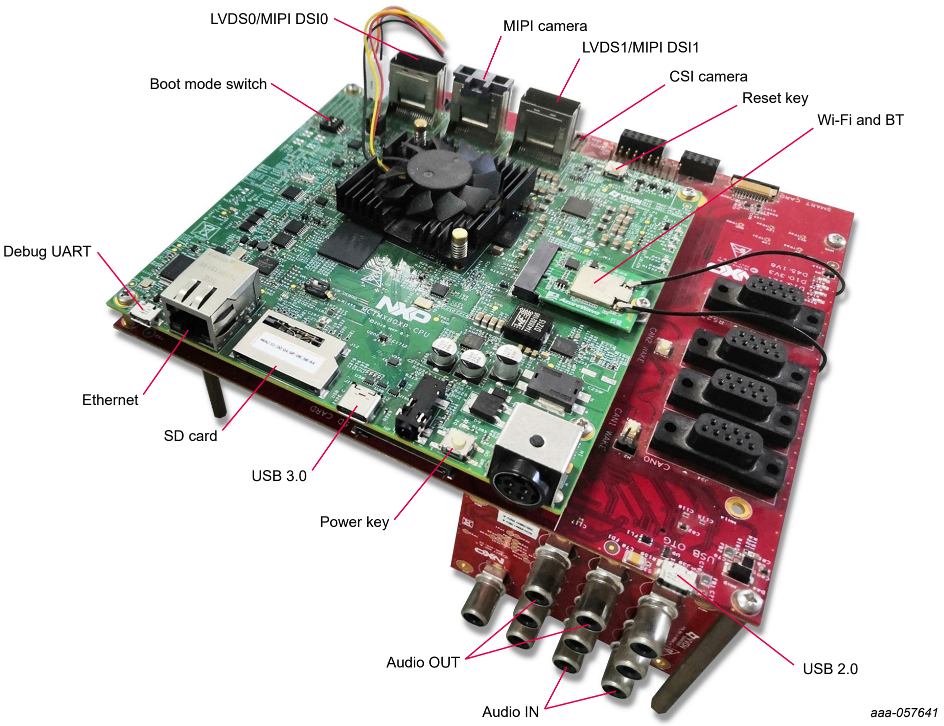


Figure 43. i.MX 8QuadXPlus MEK board

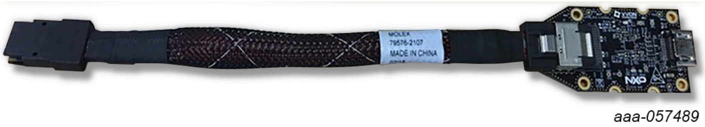


Figure 44. i.MX mini SAS cable with DSI-to-HDMI adapter

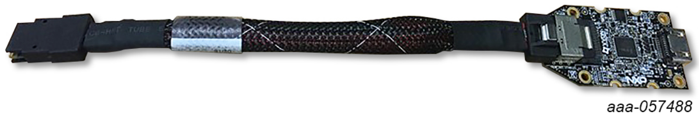


Figure 45. i.MX mini SAS cable with LVDS-to-HDMI adapter

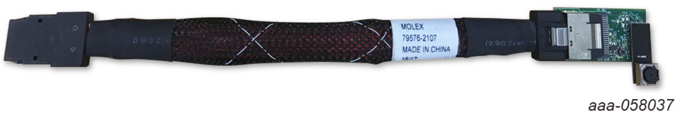
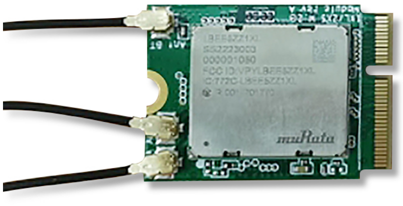


Figure 46. OV5640 CSI MIPI camera



aaa-057632

Figure 47. 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-15.0.0_1.2.0_image_8qmek.tar.gz`.

Table 13. Board images

Image name	Download target	Description
`spl-imx8qxp-dual.bin`	32 kB offset of MMC.	Secondary program loader image without Trusty related configurations for the i.MX 8QuadXPlus MEK board with the Silicon revision B0 chip.
`bootloader-imx8qxp-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8 QuadXPlus MEK board with the Silicon revision B0 chip.
`spl-imx8qxp-c0-dual.bin`	0 kB offset of eMMC or 32 kB offset of SD card.	Secondary program loader image without Trusty related configurations for the i.MX 8QuadXPlus MEK board with the Silicon revision C0 chip.
`bootloader-imx8qxp-c0-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 8 QuadXPlus MEK board with the Silicon revision C0 chip.
`spl-imx8qxp-trusty-dual.bin`	32 kB offset of MMC.	Secondary program loader image with Trusty related configurations for the i.MX 8QuadXPlus MEK board with the Silicon revision B0 chip.
`bootloader-imx8qxp-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8QuadXPlus MEK board with the Silicon revision B0 chip.
`spl-imx8qxp-trusty-secure-unlock-dual.bin`	32 kB offset of MMC.	Secondary program loader image with Trusty and secure unlock related configurations for the i.MX 8QuadXPlus EVK board with the Silicon revision B0 chip.

Table 13. Board images*...continued*

Image name	Download target	Description
bootloader-imx8qxp-trusty-secure-unlock-dual.img	bootloader_a and bootloader_b partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of secure unlock mechanism for the i.MX 8Quad XPlus EVK board with the Silicon revision B0 chip.
spl-imx8qxp-trusty-c0-dual.bin	0 kB offset of eMMC.	Secondary program loader image with Trusty related configurations for the i.MX 8QuadXPlus MEK board with Silicon revision C0 chip.
bootloader-imx8qxp-trusty-c0-dual.img	bootloader_a and bootloader_b partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 8QuadXPlus MEK board with Silicon revision C0 chip.
u-boot-imx8qxp.imx	32 kB offset of MMC.	An image containing U-Boot and ATF for the i.MX 8QuadXPlus MEK board with the Silicon revision B0 chip.
u-boot-imx8qxp-c0.imx	0 kB offset of MMC or 32 kB offset of SD card.	An image containing U-Boot and ATF for the i.MX 8QuadXPlus MEK board with Silicon revision C0 chip.
u-boot-imx8qxp-mek-uuu.imx	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 8 QuadXPlus MEK board with the Silicon revision B0 chip. It is not flashed to MMC.
u-boot-imx8qxp-mek-c0-uuu.imx	N/A	An image containing U-Boot and ATF, used by UUU for the 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 partition table image used for single-bootloader condition and for boot storage larger than 13 GB.
partition-table-dual.img	Programed to first 17 kB, and then back up to last 17 kB of the boot storage.	GPT partition table image used 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 partition table image used 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 partition table image used for dual-bootloader condition and for boot storage larger than 28 GB.
boot.img	boot_a and boot_b partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image and generic ramdisk.
boot-imx.img	boot_a and boot_b partitions.	Boot image built with i.MX kernel tree for debugging.
init_boot.img	init_boot_a and init_boot_b partitions.	Contains generic ramdisk.
vendor_boot.img	vendor_boot_a and vendor_boot_b partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
vendor_boot-debug.img	vendor_boot_a and vendor_boot_b partitions.	Used in the VTS test with GSI system image.
vbmeta-imx8qxp.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support single LVDS-to-HDMI/MIPI-DSI-to-HDMI or dual LVDS-to-HDMI displays with dual-cameras support.
vbmeta-imx8qxp-lvds0-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the LVDS panel display.

Table 13. Board images*...continued*

Image name	Download target	Description
`vbmeta-imx8qxp-mipi-panel.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the MIPI panel (RM67199) display.
`vbmeta-imx8qxp-mipi-panel-rm67191.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the MIPI panel (RM67191) display.
`vbmeta-imx8qxp-sof.img`	`vbmeta_a` and `vbmeta_b` partitions.	Android Verify Boot metadata image to support the Sound Open Firmware audio output.
`system.img`	Logical partitions `system_a` and logical partition `system_b` in super partition.	System image.
`system.img`	Logical partitions `system_a` and `system_b` in super partition.	System image.
`system_ext.img`	Logical partitions `system_ext_a` and `system_ext_b` in super partition.	System extension image.
`vendor.img`	Logical partitions `vendor_a` and `vendor_b` in super partition.	Vendor image.
`vendor_dlkm.img`	Logical partitions `vendor_dlkm_a` and `vendor_dlkm_b` in super partition.	Vendor dynamically loadable kernel module image.
`product.img`	Logical partitions `product_a` and `product_b` in super partition.	Product image.
`super.img`	Super partition.	Super image which contains images for logical partitions.
`dtbo-imx8qxp.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the 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.	Device Tree image to support the LVDS panel display.
`dtbo-imx8qxp-mipi-panel.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the MIPI panel (RM67199) display.
`dtbo-imx8qxp-mipi-panel-rm67191.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the MIPI panel (RM67191) display.
`dtbo-imx8qxp-sof.img`	`dtbo_a` and `dtbo_b` partitions.	Device Tree image to support the Sound Open Firmware audio output.
`rpmb_key_test.bin`	N/A	Prebuilt test RPMB key. It can be used to set the RPMB key to fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set the RPMB key.
`testkey_public_rsa4096.bin`	N/A	Prebuilt AVB public key, which is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.

Note: *In this document, logical partition is also called dynamic partition. For more Android dynamic partitions, see https://source.android.com/docs/core/ota/dynamic_partitions.*

*With virtual A/B enabled, the `super.img` file generated by the Android build system only has slot a dynamic partitions filled with images. Slot b dynamic partitions like `system_b` is **0** in size. Therefore, the system cannot boot up from slot b with prebuilt images. To boot from slot b, take either of the following actions:*

- *Apply an OTA update.*
- *Boot the board into fastbootd mode, and flash the images to slot b logical partitions. Set slot b as the active slot.*

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 1.5.179 version. Download the 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-15.0.0_1.2.0_image_8qmek.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
5. Execute the `uuu_imx_android_flash` tool to flash images.
   The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For 8QuadXPlus board, related options are described as follows.

Table 14. Options for uuu_imx_android_flash tool

Option	Description
`-h`	Displays the help information of this tool.
`-f soc_name`	Specifies SoC information. For i.MX 8QuadXPlus MEK, it should be `imx8qxp`. This option is **mandatory**.
`-a`	Only flashes slot a physical partitions and other partitions do not have slot a/b. If this option and the `-b` option are not used, slots a and b are both flashed.

Table 14. Options for uuu_imx_android_flash tool*...continued*

Option	Description
`-b`	Only flashes slot b physical partitions and other partitions do not have slot a/b. If this option and the `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 8QuadXPlus MEK, it can be followed with "28". If this option is not used, default `partition-table.img` is flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader images with `uboot_feature` in their names. For i.MX 8QuadXPlus MEK, it can be `dual`, `trusty-dual`, `trusty-secure-unlock-dual`, `c0`, `c0-dual` or `trusty-c0-dual`. If this option is not used, the default `u-boot-imx8qxp.imx` is flashed.
`-d dtb_feature`	Flash DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 8Quad XPlus MEK, it can be `sof`, `mipi-panel`, `mipi-panel-rm67191` or `lvds0-panel`. If it is not used, the default `dtbo-imx8qxp.img` and `vbmeta-imx8qxp.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 8QuadXPlus MEK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not executes UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

- On the 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 the 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:

- *`-b` is not suggested to be used, because virtual A/B is enbaled and the system cannot boot up from slob b with the prebuilt images.*
- *`-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 (UG10158).*
- *To test the Sound Open Firmware audio output, execute the tool with* `-d sof`*.*
- *To test the MIPI panel (RM67199) display, execute the tool with* `-d mipi-panel`*. The MIPI_DSI0 port is used for display.*
- *To test the MIPI panel (RM67191) display, execute the tool with* `-d mipi-panel-rm67191`*. The MIPI_DSI0 port is used for display.*
- *To test the 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.

Note:
If Trusty OS is used, to make video playback work, follow Section "Secure firmware loader" in the i.MX Android Security User's Guide (UG10158) to flash the keys related to the firmware loader to ensure that the VPU firmware can be loaded successfully. The relevant keys can be obtained from the `firmware_test_keys` *directory in the directory where the image is located.*

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 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.

It does not need to set U-Boot enviroment variables for the currently avaiable boot scenarios.

9.5 Board reboot

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

10 Working with the i.MX 95 EVK Board

10.1 Board hardware

The figures below show the different components of the i.MX 95 EVK board.

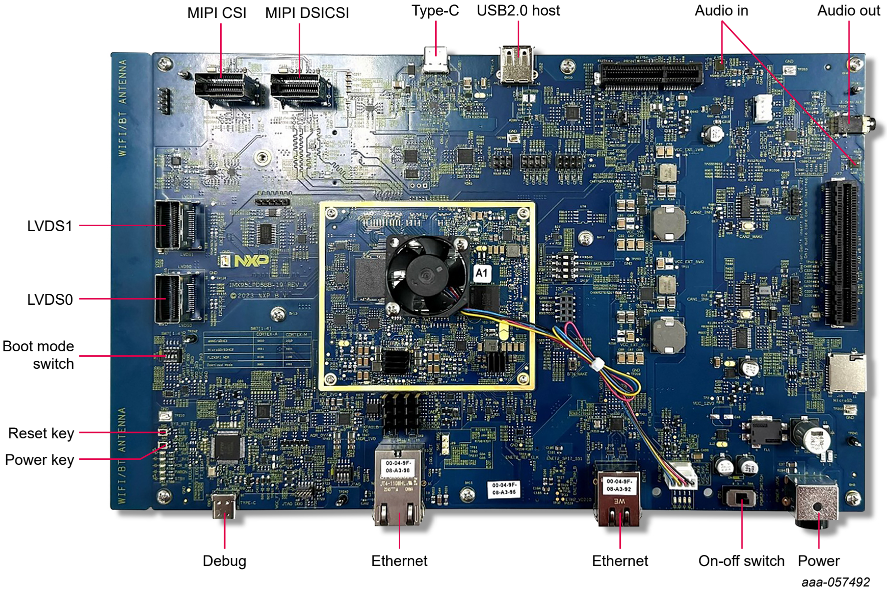


Figure 48. i.MX 95 EVK

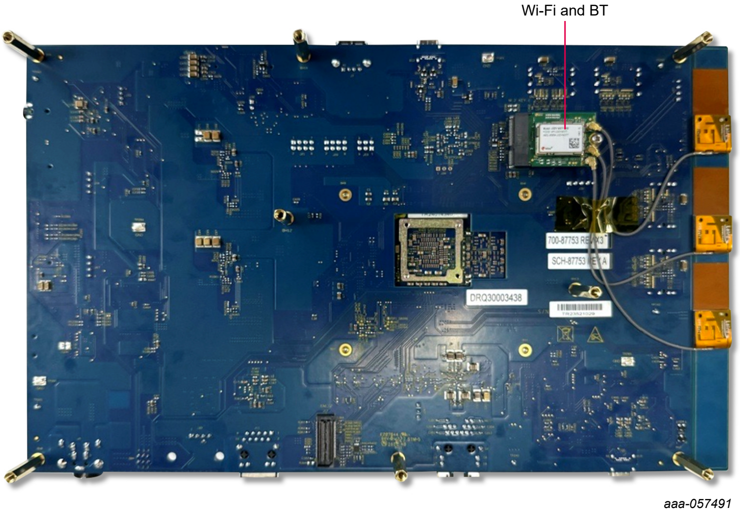


Figure 49. i.MX 95 EVK board back view

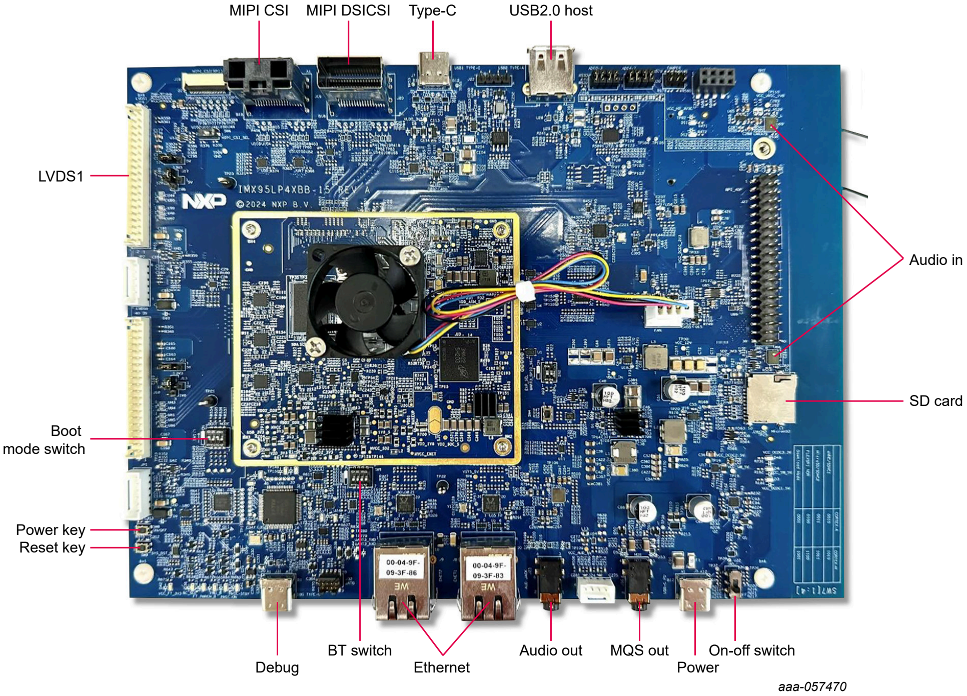


Figure 50. i.MX 95 15x15 EVK

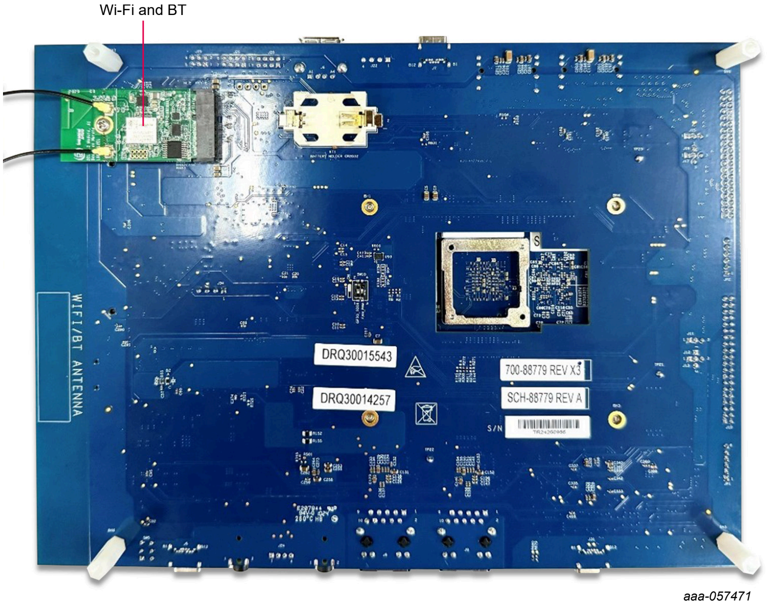


Figure 51. i.MX 95 15x15 EVK board back view

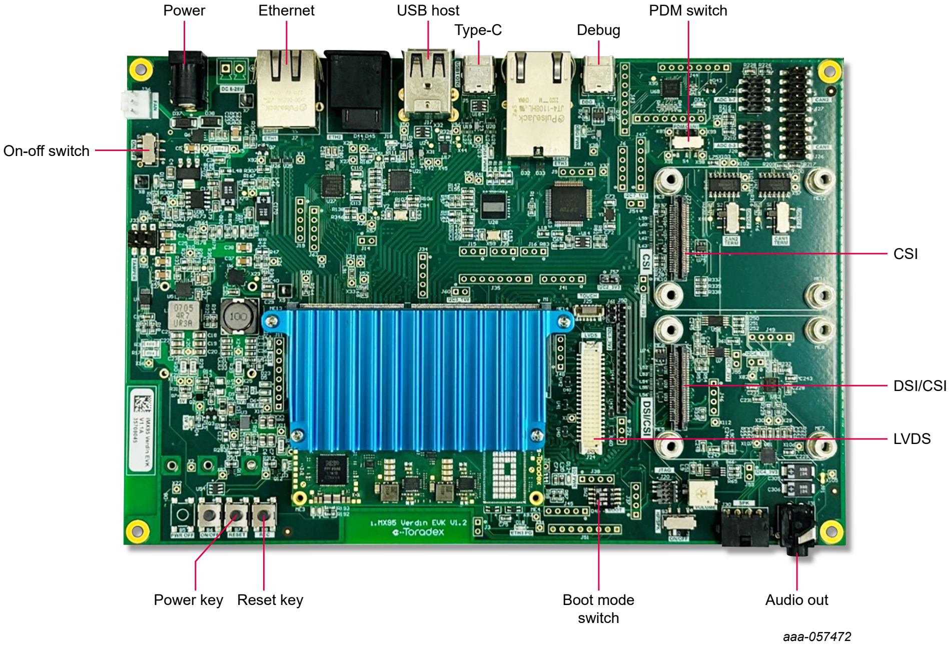


Figure 52. i.MX 95 Verdin EVK

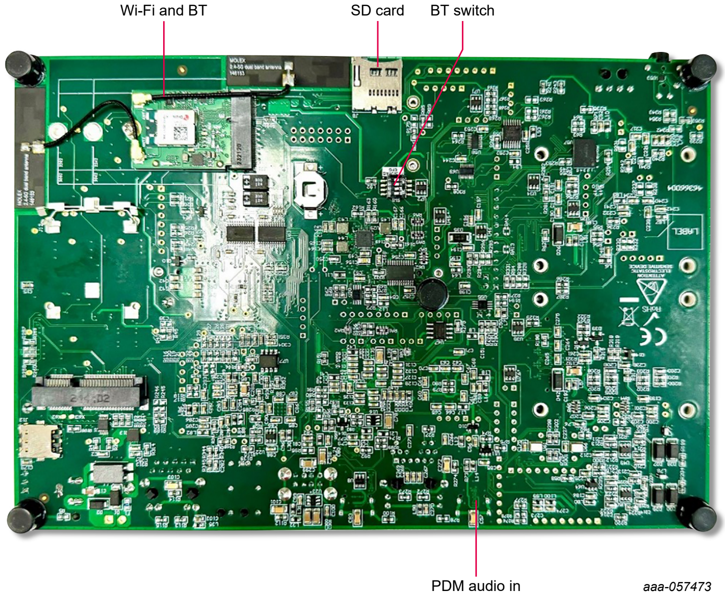


Figure 53. i.MX 95 Verdin EVK board back view

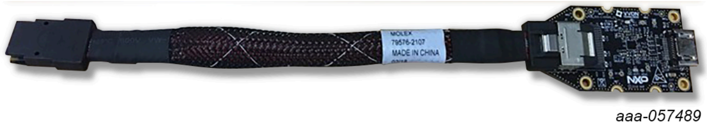


Figure 54. i.MX mini SAS cable with DSI-to-HDMI adapter

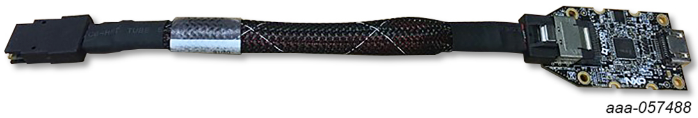
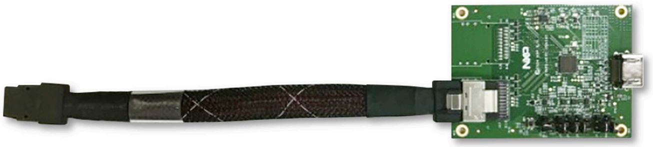
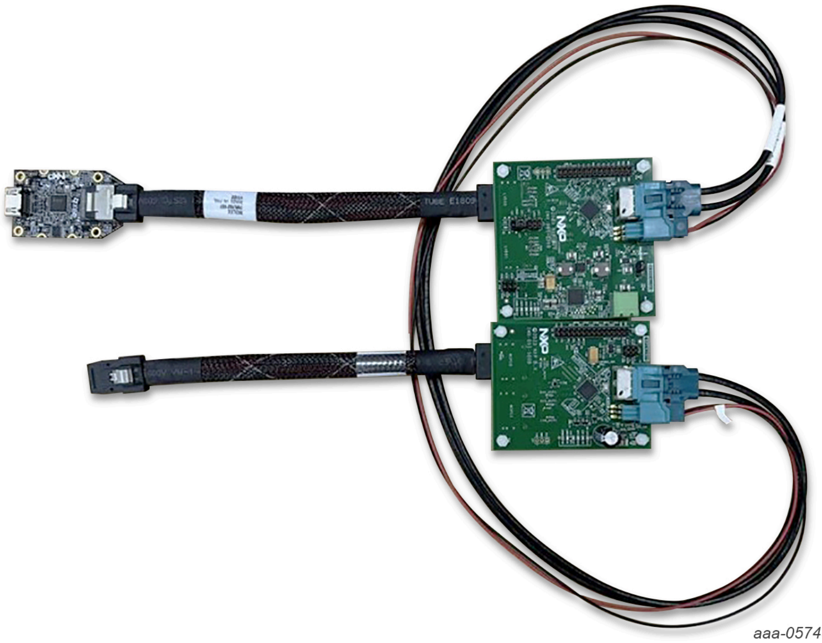


Figure 55. i.MX mini SAS cable with LVDS-to-HDMI adapter



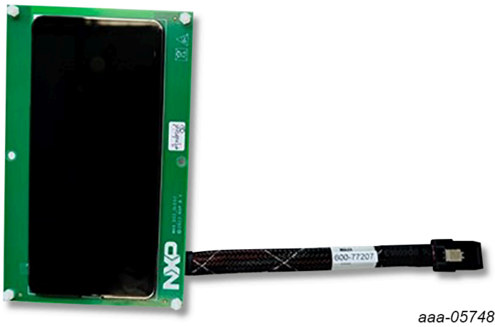
aaa-057457

Figure 56. i.MX mini SAS cable with DSI-to-HDMI4K adapter



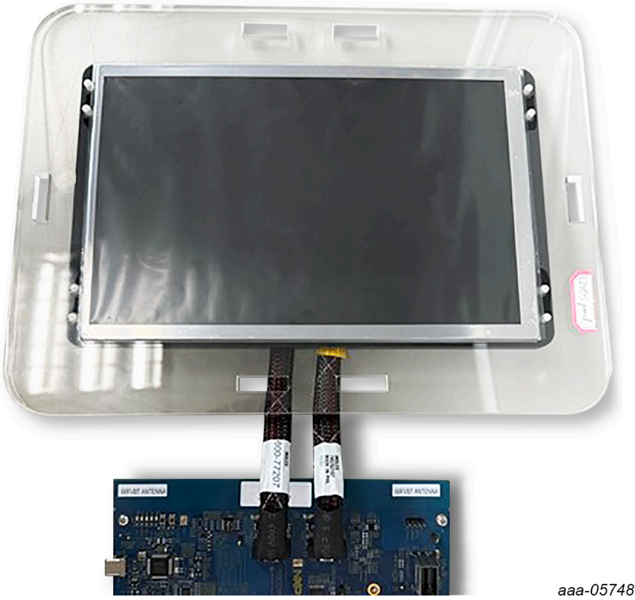
aaa-057456

Figure 57. i.MX mini SAS cable with DSI-to-SDSB (Serialiser and Deserialiser) adapter



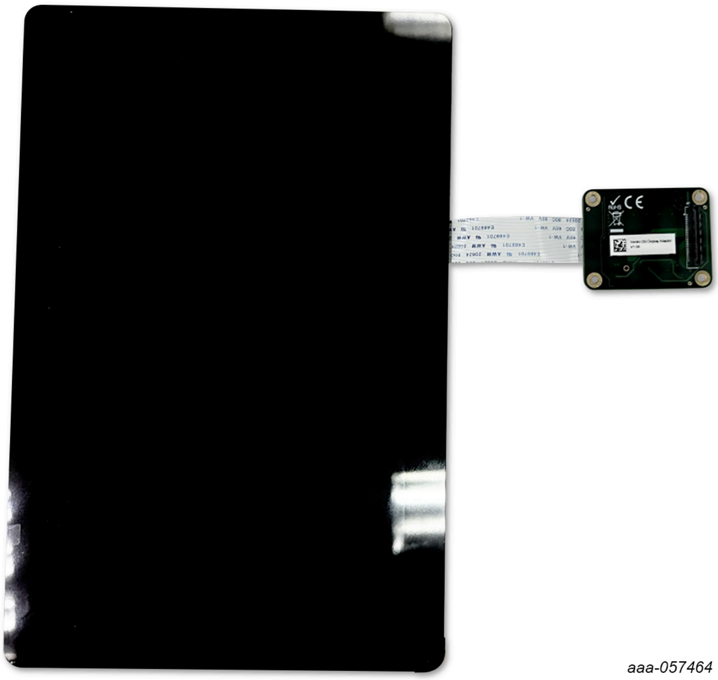
aaa-057483

Figure 58. i.MX MIPI panel RM692C9



aaa-057482

Figure 59. i.MX LVDS panel



aaa-057464

Figure 60. i.MX 10-inch DSI panel

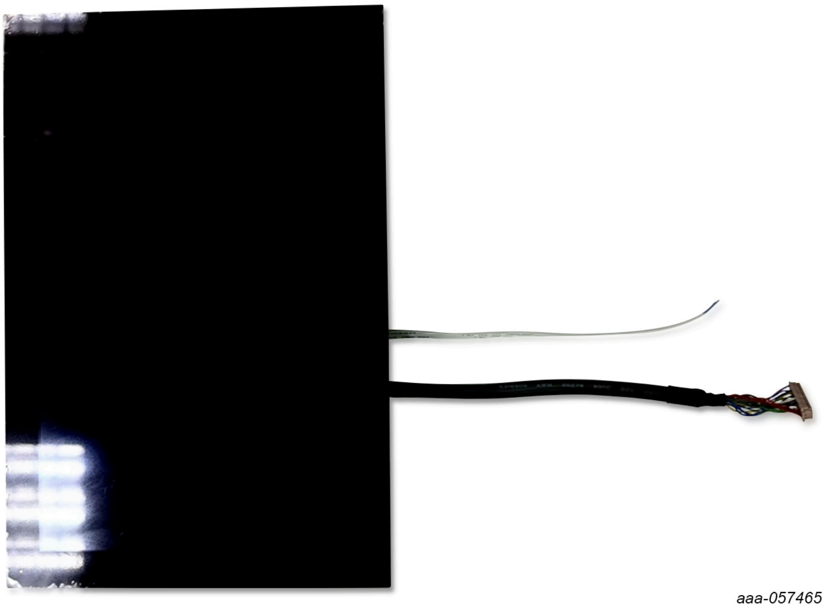


Figure 61. i.MX 10-inch LVDS panel

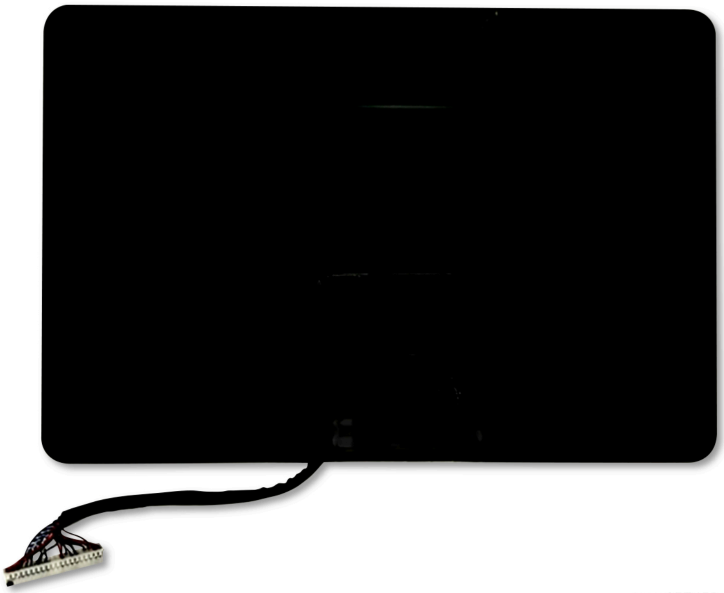


Figure 62. i.MX BOE panel

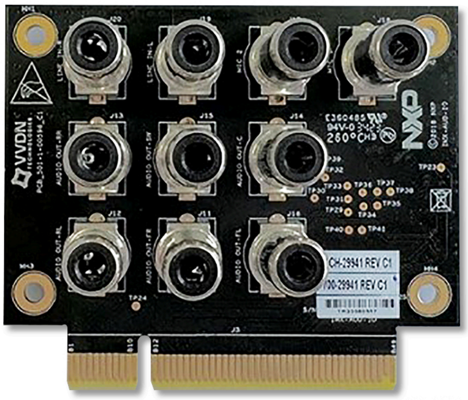


Figure 63. AUDIO-IO board

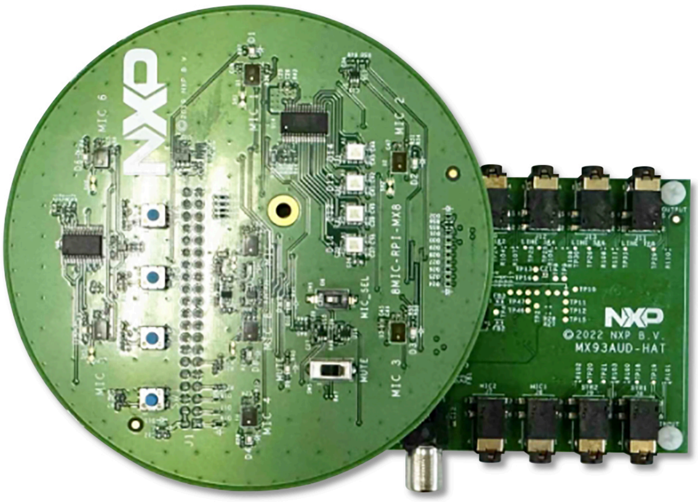


Figure 64. Audio HAT board + i.MX 8MIC PDM Microphone board

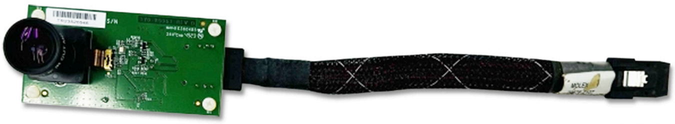


Figure 65. i.MX CSI MIPI Camera OS08A20

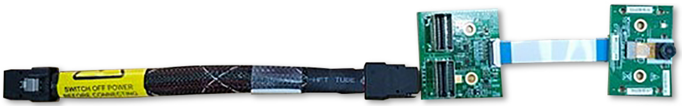
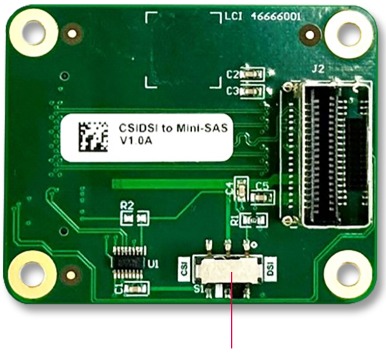
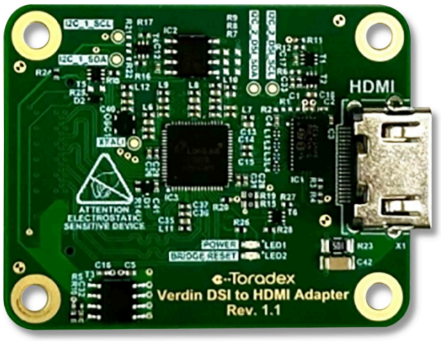


Figure 66. i.MX CSI MIPI Camera AP1302



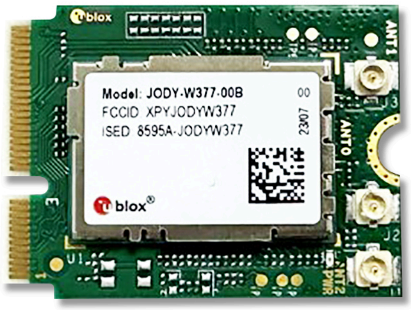
aaa-057474

Figure 67. i.MX 95 Verdin CSI/DSI to Mini-SAS adapter



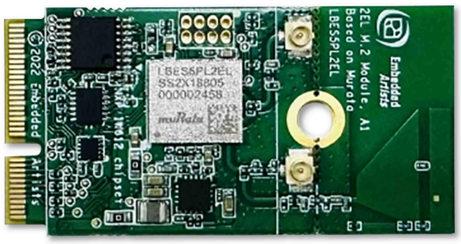
aaa-057461

Figure 68. i.MX 95 Verdin DSI-to-HDMI adapter



aaa-057501

Figure 69. PCIE9098 (U-Blox JODY-W377)



aaaa-057479

Figure 70. SDIW612 (Murata LBES5PL2EL)

Figure 71. SDIW612 (U-Blox MAYA-W271)

Note:

- *i.MX 95 EVK, i.MX 95 15x15 EVK, and i.MX 95 Verdin EVK boards are supported in this release.*
- *To test the MIPI-DSI-to-HDMI display on i.MX 95 EVK and i.MX 95 15x15 EVK, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "MIPI DSICSI" port.*
- *To test the LVDS-to-HDMI display on i.MX 95 EVK, use the i.MX mini SAS cable to connect the LVDS-to-HDMI adapter to the "LVDS0" port.*
- *To test the MIPI panel display on i.MX 95 EVK and i.MX 95 15x15 EVK, connect the i.MX MIPI panel RM692C9 to the "MIPI DSICSI" port.*
- *To test the LVDS panel display on i.MX 95 EVK, use two i.MX mini SAS cables to connect the LVDS panel to the "LVDS0" and "LVDS1" ports, connect "LVDS0" to J8 of the LVDS panel, and connect "LVDS1" to J9 of the LVDS panel.*
- *To test the dual LVDS-to-HDMI displays on i.MX 95 EVK, use two i.MX mini SAS cables to connect two LVDS-to-HDMI adapters to the "LVDS0" and "LVDS1" ports.*
- *To test the MIPI-DSI-to-HDMI and LVDS-to-HDMI displays on i.MX 95 EVK, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "MIPI DSI" port, and use the i.MX mini SAS cable to connect the LVDS-to-HDMI adapter to the "LVDS1" port.*
- *To test the DSI-to-HDMI4KI display on i.MX 95 EVK and i.MX 95 15x15 EVK, use the i.MX mini SAS cable to connect the DSI-to-HDMI4KI adapter to the "MIPI DSICSI" port on boards.*
- *To test the DSI-to-HDMI4KI display on i.MX 95 Verdin EVK, use the i.MX mini SAS cable to connect the DSI-to-HDMI4KI adapter to the "Verdin CSI/DSI to Mini-SAS" adapter, and then connect the "Verdin CSI/DSI to Mini-SAS" adapter with the "DSI/CSI" port on the i.MX 95 Verdin EVK board. Then switch the "CSI/DSI Switch" to DSI.*
- *To test the DSI-to-SDSB display on i.MX 95 EVK, connect 1-3, 2-4 of the J16 jumper on Deserialiser, use the i.MX mini SAS cable to connect the DSI-to-SDSB (Serialiser and Deserialiser) adapter to the "MIPI DSICSI" port on boards.*
- *To test the BOE panel display on i.MX 95 15x15 EVK, connect the i.MX BOE panel to the "LVDS1" port on i.MX 95 15x15 EVK.*
- *To test the 10-inch DSI panel diaplay on i.MX 95 Verdin EVK, connect the i.MX 10-inch DSI panel to the "DSI/CSI" port on i.MX 95 Verdin EVK.*
- *To test the 10-inch LVDS panel display on i.MX 95 Verdin EVK, connect the i.MX 10-inch LVDS panel to the "LVDS" port on i.MX 95 Verdin EVK.*
- *To test the MIPI panel display on i.MX 95 Verdin EVK, connect the "Verdin CSI/DSI to Mini-SAS adapter" to the i.MX MIPI panel RM692C9, and then to the "DSI/CSI" port on the i.MX 95 Verdin EVK board. Then switch the "CSI/DSI Switch" to DSI.*
- *To test the MIPI-to-HDMI (ADV7535) display on the i.MX 95 Verdin EVK, use the i.MX mini SAS cable to connect the "Verdin CSI/DSI to Mini-SAS" adapter to the DSI-to-HDMI adapter, and then connect the DSI-to-*

HDMI adapter with the "DSI/CSI" port on the i.MX 95 Verdin EVK board. Then switch the "CSI/DSI Switch" to DSI.

- *To test the MIPI-to-HDMI (LT8912) display on i.MX 95 Verdin EVK, connect the Verdin DSI-to-HDMI adapter to the "DSI/CSI" port.*
- *To test the camera on i.MX 95 EVK and i.MX 95 15x15 EVK, connect the i.MX CSI MIPI Camera OS08A20 or AP1302 to the "MIPI CSI" port.*
- *To test the camera on i.MX 95 Verdin EVK, connect the "Verdin CSI/DSI to Mini-SAS" adapter to the OS08A20 or AP1302 to the "CSI" port. Then switch the "CSI/DSI Switch" to CSI.*
- *To test the AUDIO-IO board, connect the AUDIO-IO board to the J27 connector on the i.MX 95 EVK board.*
- *To test the Audio HAT board + i.MX 8MIC PDM Microphone board, connect the Audio HAT board + i.MX 8MIC PDM Microphone board to J27 on the i.MX95 EVK 15x15 board.*
- *To test the PDM Audio in, turn on the PDM switch on the i.MX 95 Verdin EVK board.*
- *Connect the PCIE9098 (U-Blox JODY-W377) Wi-Fi and Bluetooth M.2 expansion card to the J24 connector to enable the Wi-Fi and Bluetooth on the i.MX 95 EVK board.*
- *Connect the SDIW612 (Murata LBES5PL2EL) Wi-Fi and Bluetooth M.2 expansion card to the J33 connector and change the Bluetooth switch SW1 to 1111 (from 1-4 bit) to enable the Wi-Fi and Bluetooth on the i.MX 95 15x15 EVK board.*
- *Connect the SDIW612 (U-Blox MAYA-W271) Wi-Fi and Bluetooth M.2 expansion card to the J21 connector and change Bluetooth switch SW1 to 1111 (from 1-4 bit) to enable the Wi-Fi and Bluetooth on the i.MX 95 EVK Verdin board.*

10.2 Board images

The table below describes the location in the board partitions of the software images in `android-15.0.0_1.2.0_image_95evk.tar.gz`.

Image name	Download target	Description
`spl-imx95-dual.bin`	`bootloader0` partition.	Secondary program loader image without Trusty related configurations for the i.MX 95 EVK board.
`spl-imx95-trusty-dual.bin`	`bootloader0` partition.	Secondary program loader image with Trusty related configurations for the i.MX 95 EVK board.
`spl-imx95-15x15-dual.bin`	`bootloader0` partition.	Secondary program loader image without Trusty related configurations for the i.MX 95 15x15 EVK board.
`spl-imx95-trusty-15x15-dual.bin`	`bootloader0` partition.	Secondary program loader image without Trusty related configurations for the i.MX 95 15x15 EVK board.
`spl-imx95-trusty-verdin-dual.bin`	`bootloader0` partition.	Secondary program loader image with Trusty related configurations for i.MX 95 Verdin EVK board.
`spl-imx95-trusty-secure-unlock-dual.bin`	`bootloader0` partition.	Secondary program loader image with Trusty and secure unlock related configurations for the i.MX 95 EVK board.
`bootloader-imx95-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 95 EVK board.
`bootloader-imx95-trusty-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 95 EVK board.
`bootloader-imx95-15x15-dual.img`	`bootloader_a` and `bootloader_b` partitions.	An image containing U-Boot proper and ATF. It is for the i.MX 95 15x15 EVK board.

Image name	Download target	Description
bootloader-imx95-trusty-15x15-dual.img	bootloader_a and bootloader_b partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 95 15x15 EVK board.
bootloader-imx95-trusty-verdin-dual.img	bootloader_a and bootloader_b partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is for the i.MX 95 Verdin EVK board.
bootloader-imx95-trusty-secure-unlock-dual.img	bootloader_a and bootloader_b partitions.	An image containing U-Boot proper, ATF, and Trusty OS. It is a demonstration of the secure unlock mechanism for the i.MX 95 EVK board.
u-boot-imx95.imx	bootloader0 partition.	An image containing U-Boot and ATF for the i.MX 95 EVK board.
u-boot-imx95-evk-uuu.imx	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 95 EVK board. It is not flashed to MMC.
u-boot-imx95-15x15.imx	bootloader0 partition.	An image containing U-Boot and ATF for the i.MX 95 15x15 EVK board.
u-boot-imx95-15x15-evk-uuu.imx	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 95 15x15 EVK board. It is not flashed to MMC.
u-boot-imx95-verdin.imx	bootloader0 partition.	An image containing U-Boot and ATF for the i.MX 95 Verdin EVK board.
u-boot-imx95-verdin-uuu.imx	N/A	An image containing U-Boot and ATF, used by UUU for the i.MX 95 Verdin EVK board. It is not flashed to MMC.
u-boot-imx95-rpmsg.imx	bootloader0 partition.	Android Verify Boot metadata image to support the MIPI-to-HDMI output and MCU image on the i.MX 95 EVK board.
partition-table.img	0 offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 28 GB.
partition-table-dual.img	0 offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 28 GB.
partition-table-13GB.img	0 offset of MMC.	GPT partition table image for single-bootloader condition and the target storage device should be larger than 13 GB.
partition-table-13GB-dual.img	0 offset of MMC.	GPT partition table image for dual-bootloader condition and the target storage device should be larger than 13 GB.
boot.img	boot_a and boot_b partitions.	AOSP GKI boot image. It contains the AOSP generic kernel image and generic ramdisk.
boot-imx.img	boot_a and boot_b partitions.	Boot image built with the i.MX kernel tree for debugging.
init_boot.img	init_boot_a and init_boot_b partitions.	Contains generic ramdisk.
vendor_boot.img	vendor_boot_a and vendor_boot_b partitions.	Vendor boot image. It contains vendor ramdisk, kernel commandline, and bootconfig.
vendor_boot-debug.img	vendor_boot_a and vendor_boot_b partitions.	Used in the VTS test with the GSI system image.
vbmeta-imx95.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display, and support OS08A20 camera on the i.MX 95 EVK board.

Image name	Download target	Description
vbmeta-imx95-ap1302.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display, and support the the AP1302 camera on the i.MX 95 EVK board.
vbmeta-imx95-mipi-lvds1.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI and LVDS-to-HDMI multiple display, and support AP1302 camera on the i.MX 95 EVK board.
vbmeta-imx95-mipi-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the RM692C9 MIPI panel display on the i.MX 95 EVK board.
vbmeta-imx95-lvds0.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the LVDS-to-HDMI display, and support the OS08A20 camera on the i.MX 95 EVK board.
vbmeta-imx95-lvds-dualdisp.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the dual LVDS-to-HDMI displays on the i.MX 95 EVK board.
vbmeta-imx95-lvds-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the LVDS panel display on the i.MX 95 EVK board.
vbmeta-imx95-cs42888.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display, AUDIO-IO board, and AP1302 camera on the i.MX 95 EVK board.
vbmeta-imx95-rpmsg.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display and MCU image on the i.MX 95 EVK board.
vbmeta-imx95-mipi4k.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI4 K display, and support the the AP1302 camera on the i.MX 95 EVK board.
vbmeta-imx95-dsi-serdes.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the DSI-to-SDSB display on the i.MX 95 EVK board.
vbmeta-imx95-15x15.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display, and support the OS08A20 camera on the i.MX 95 15x15 EVK board.
vbmeta-imx95-15x15-ap1302.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display, and support the AP1302 camera on the i.MX 95 15x15 EVK board.
vbmeta-imx95-15x15-mipi-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the RM692C9 MIPI panel display on the i.MX 95 15x15 EVK board.
vbmeta-imx95-15x15-aud-hat.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display and Audio HAT board + i.MX 8MIC PDM Microphone board on the i.MX 95 15x15 EVK board.
vbmeta-imx95-15x15-mqs.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI display and Audio MQS output and WM8962 input on the i.MX 95 15x15 EVK board.
vbmeta-imx95-15x15-mipi4k.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI4 K display, and support AP1302 on the i.MX 95 15x15 EVK board.
vbmeta-imx95-15x15-boe-panel-lvds1.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the BOE panel display on the i.MX 95 15x15 EVK board.
vbmeta-imx95-verdin.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI (ADV7535) display, and support the OS08A20 camera on the i.MX 95 Verdin EVK board.

Image name	Download target	Description
vbmeta-imx95-verdin-ap1302.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI(ADV7535) display, and support AP1302 camera on the i.MX 95 Verdin EVK board.
vbmeta-imx95-verdin-lt8912.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI(LT8912) display, and support AP1302 camera on the i.MX 95 Verdin EVK board.
vbmeta-imx95-verdin-10inch-panel-lvds.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the 10-inch LVDS panel display on the i.MX 95 Verdin EVK board.
vbmeta-imx95-verdin-10inch-panel-dsi.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the 10-inch DSI panel display on the i.MX 95 Verdin EVK board.
vbmeta-imx95-verdin-mipi-panel.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the RM692C9 MIPI panel display on the i.MX 95 Verdin EVK board.
vbmeta-imx95-verdin-mipi4k.img	vbmeta_a and vbmeta_b partitions.	Android Verify Boot metadata image to support the MIPI-to-HDMI4K display, and support the AP1302 camera on the i.MX 95 Verdin EVK board.
system.img	Logical partitions system_a and system_b in super partition.	System image.
system_ext.img	Logical partitions system_ext_a and system_ext_b in super partition.	System extension image.
system_dlkm.img	Logical partitions system_dlkm_a and system_dlkm_b in super partition.	System dynamically loadable kernel module image.
vendor.img	Logical partitions vendor_a and vendor_b in super partition.	Vendor image
vendor_dlkm.img	Logical partitions vendor_dlkm_a and vendor_dlkm_b in super partition.	Vendor dynamically loadable kernel module image.
product.img	Logical partitions product_a and product_b in super partition.	Product image.
super.img	Super partition.	Super image, which contains images for logical partitions.
dtbo-imx95.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display, and support the OS08A20 camera on the i.MX 95 EVK board.
dtbo-imx95-ap1302.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display, and support the AP1302 camera on the i.MX 95 EVK board.
dtbo-imx95-mipi-lvds1.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI and LVDS-to-HDMI multiple display, and support the AP1302 camera on the i.MX 95 EVK board.

Image name	Download target	Description
dtbo-imx95-mipi-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the RM692C9 MIPI panel display on the i.MX 95 EVK board.
dtbo-imx95-lvds0.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the LVDS-to-HDMI display, and support the OS08A20 camera on the i.MX 95 EVK board.
dtbo-imx95-lvds-dualdisp.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the dual-channel LVDS-to-HDMI display on the i.MX 95 EVK board.
dtbo-imx95-lvds-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the LVDS panel display on the i.MX 95 EVK board.
dtbo-imx95-cs42888.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display, AUDIO-IO board, and AP1302 camera on the i.MX 95 EVK board.
dtbo-imx95-rpmsg.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display and MCU image on the i.MX 95 EVK board.
dtbo-imx95-mipi4k.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI4K display, and support the AP1302 camera on the i.MX 95 EVK board.
dtbo-imx95-dsi-serdes.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the DSI-to-SDSB display i.MX 95 EVK board.
dtbo-imx95-15x15.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display, and support the OS08A20 camera on the i.MX 95 15x15 EVK board.
dtbo-imx95-15x15-ap1302.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display, and support the AP1302 camera on the i.MX 95 15x15 EVK board.
dtbo-imx95-15x15-mipi-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the RM692C9 MIPI panel display on the i.MX 95 15x15 EVK board.
dtbo-imx95-15x15-aud-hat.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display and Audio HAT board + i.MX 8MIC PDM Microphone board on the i.MX 95 15x15 EVK board.
dtbo-imx95-15x15-mqs.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI display and Audio MQS output on the i.MX 95 15x15 EVK board.
dtbo-imx95-15x15-mipi4k.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI4K display, and support AP1302 on the i.MX 95 15x15 EVK board.
dtbo-imx95-15x15-boe-panel-lvds1.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the BOE panel display on the i.MX 95 15x15 EVK board.
dtbo-imx95-verdin.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI (ADV7535) display, and support the OS08A20 camera on the i.MX 95 Verdin EVK board.
dtbo-imx95-verdin-ap1302.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI (ADV7535) display, and support the AP1302 camera on the i.MX 95 Verdin EVK board.
dtbo-imx95-verdin-lt8912.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI (LT8912) display, and support the AP1302 camera on the i.MX 95 Verdin EVK board.
dtbo-imx95-verdin-10inch-panel-lvds.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the 10-inch LVDS panel display on the i.MX 95 Verdin EVK board.
dtbo-imx95-verdin-10inch-panel-dsi.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the 10-inch DSI panel display on the i.MX 95 Verdin EVK board.
dtbo-imx95-verdin-mipi-panel.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the RM692C9 MIPI panel display on the i.MX 95 Verdin EVK board.

Image name	Download target	Description
dtbo-imx95-verdin-mipi4k.img	dtbo_a and dtbo_b partitions.	Device Tree image to support the MIPI-to-HDMI4K display, and support the AP1302 camera on the i.MX 95 Verdin EVK board.
rpmb_key_test.bin	N/A	Prebuilt test RPMB key, which can be used to set the RPMB key as fixed 32 bytes 0x00. See the *Android User's Guide* (UG10156) for how to set the RPMB key.
testkey_public_rsa4096.bin	N/A	Prebuilt AVB public key. It is extracted from the default AVB private key. See the *Android User's Guide* (UG10156) for how to provision the public key.
android-15.0.0_1.2.0_image_trout.tar.gz	N/A	Trout Android image, which runs as an i.MX 95 Xen Guest. For more details, see Section 11.4 "Building images" and Section 11.6 "Installing the Trout image" in the *Android User's Guide* (UG10156).

10.3 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 1.5.179 version. Download the 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.
   a. For i.MX 95 EVK and i.MX 95 15X15 EVK:
      Change the board's SW7 to 1001 (from 1-4 bit) to enter serial download mode.
   b. For i.MX 95 Verdin EVK:
      Change the board's SW2 to 1001 (from 1-4 bit) to enter serial download mode.
3. Power on the board. Use the USB cable on the board Type-C port to connect your PC with the board.
   Note: *The debug PORT can be used to watch the logs of the hardware boot processing.*
4. Decompress release_package/android-15.0.0_1.2.0_image_95evk.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 the i.MX 95 EVK board, related options are described as follows.

Table 15. Options for uuu_imx_android_flash tool

Option	Description
-h	Displays the help information of this tool.

Table 15. Options for uuu_imx_android_flash tool*...continued*

Option	Description
`-f soc_name`	Specifies the SoC information. For i.MX 95 EVK, it should be `imx95`. This option is **mandatory**.
`-a`	Only flashes slot a. If this option and `-b` option are not used, slots a and b are both flashed.
`-b`	Only flashes slot b. If this option and `-a` option are not used, slots a and b are both flashed.
`-c card_size`	Specifies which partition table image file to flash. For i.MX 95 EVK, it can be followed with "13". If this option is not used, default `partition-table.img` or `partition-table-dual.img` is flashed.
`-m`	Flashes the MCU image. If it is not used, the MCU image is not flashed.
`-u uboot_feature`	Flashes U-Boot or SPL&bootloader images with `uboot_feature` in their names. For i.MX 95 EVK, it can be `dual`, `trusty-dual`, `trusty-secure-unlock-dual` and `rpmsg`; For i.MX 95 15x15 EVK, it can be `15x15-dual` and `trusty-15x15-dual`; For i.MX 95 Verdin EVK, it can be `trusty-verdin-dual`.
`-d dtb_feature`	Flashes DTBO and vbmeta images with `dtb_feature` in their names. For i.MX 95 EVK, it can be `ap1302`, `mipi-lvds1`, `mipi-panel`, `lvds0`, `lvds-dualdisp`, `lvds-panel`, `cs42888`, `rpmsg`, `mipi4k`, and `dsi-serdes`; For i.MX 95 15x15 EVK, it can be `15x15`, `15x15-ap1302`, `15x15-mipi-panel`, `15x15-aud-hat`, `15x15-mqs`, `15x15-mipi4k`, and `15x15-boe-panel-lvds1`; For i.MX 95 Verdin EVK, it can be `verdin`, `verdin-ap1302`, `verdin-lt8912`, `verdin-10inch-panel-lvds`, `verdin-10inch-panel-dsi`, `verdin-mipi-panel`, and `verdin-mipi4k`. If it is not used, the default `dtbo-imx95.img` and `vbmeta-imx95.img` are flashed.
`-e`	Erases user data after images are flashed.
`-D directory`	Specifies the directory in which there are the images to be flashed. For `uuu_imx_android_flash.bat`, it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
`-t target_dev`	Specifies the target device. For i.MX 95 EVK, i.MX 95 15x15 EVK, and i.MX 95 Verdin EVK, it can be `emmc` and `sd`. If this option is not used, images are flashed to eMMC.
`-daemon`	Runs UUU in daemon mode. This option is used to flash multiple boards of the same type.
`-i`	If the script is executed with this option, no image is flashed. The script just loads U-Boot to RAM and execute to fastboot mode. This option is used for development.
`-dryrun`	Only generates a UUU script but not execute UUU with this script.
`-usb usb_path`	Specifies a USB path like 1:1 to monitor. It can be used multiple times to specify more than one path.

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

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

```
> sudo ./uuu_imx_android_flash.sh -f imx95 -a -e -u trusty-dual
```

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

```
> .\uuu_imx_android_flash.bat -f imx95 -a -e -u trusty-dual
```

When the command above is executed, `spl-imx95-trusty-dual.bin` is flashed, `bootloader-imx95-trusty-dual.img` with other default images are flashed into eMMC slot a for i.MX 95 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 32 GB or uses onboard eMMC as the boot device, it does not need to use `-c` option.*
- *If your SD card is 16 GB, execute the tool with `-c 13`.*
- *To test dual-bootloader, execute the tool with `-u dual`.*
- *To test the features on the i.MX 95 EVK board:*
  - *To test Trusty OS and dual bootloader both enabled condition for i.MX 95 EVK , execute the tool with `-u trusty-dual`.*
  - *To test the demostration implementation of secure unlock, execute the tool with `-u trusty-secure-unlock-dual`. For secure unlock details, see the i.MX Android Security User Guide (UG10158).*
  - *To test the MIPI-to-HDMI display, it does not need to use the `-d` option.*
  - *To test the MIPI-to-HDMI and LVDS-to-HDMI dual displays, execute the tool with `-d mipi-lvds1`.*
  - *To test the MIPI panel (RM692C9) display, execute the tool with `-d mipi-panel`.*
  - *To test the LVDS-to-HDMI display, execute the tool with `-d lvds0`.*
  - *To test the dual-channel LVDS-to-HDMI display, execute the tool with `-d lvds-dualdisp`.*
  - *To test the LVDS panel display, execute the tool with `-d lvds-panel`.*
  - *To test the MIPI-to-HDMI4K display, execute the tool with `-d mipi4k`.*
  - *To test the DSI-to-SDSB display, execute the tool with `-d dsi-serdes`.*
  - *To test the OS08A20 camera, it does not need to use the `-d` option, or execute the tool with `-d lvds0`.*
  - *To test the AP1302 camera, execute the tool with `-d ap1302`, `-d mipi-lvds1`, `-d -cs42888`, or `-d mipi4k`.*
  - *To test the AUDIO-IO board, execute the tool with `-d cs42888` and use "MIPI DSI" for the MIPI-to-HDMI display.*
  - *To test the MIPI-to-HDMI display and MCU image, execute the tool with `-u rpmsg -d rpmsg -m`.*
- *To test the features on the i.MX 95 15x15 EVK board:*
  - *To test the Trusty OS and dual-bootloader both enabled condition for i.MX 95 15x15 EVK, execute the tool with `-u trusty-15x15-dual`.*
  - *To test the MIPI-to-HDMI display, execute the tool with `-d 15x15`.*
  - *To test the MIPI panel (RM692C9) display, execute the tool with `-d 15x15-mipi-panel`.*
  - *To test the MIPI-to-HDMI4K display, execute the tool with `-d 15x15-mipi4k`.*
  - *To test the BOE panel display, execute the tool with `-d 15x15-boe-panel-lvds1`.*
  - *To test the OS08A20 camera, execute the tool with `-d 15x15`.*
  - *To test the AP1302 camera, execute the tool with `-d 15x15-ap1302` or `-d 15x15-mipi4k`.*
  - *To test the Audio HAT with 8MIC PDM Microphone, execute the tool with `-d 15x15-aud-hat`.*
  - *To test MQS out, execute the tool with `-d 15x15-mqs`.*
- *To test the features on the i.MX 95 Verdin EVK board:*
  - *To test the Trusty OS and dual-bootloader both enabled condition for i.MX 95 Verdin EVK, execute the tool with `-u trusty-verdin-dual`.*
  - *To test the MIPI-to-HDMI (ADV7535) display, execute the tool with `-d verdin`.*
  - *To test the MIPI-to-HDMI (LT8912) display, execute the tool with `-d verdin-lt8912`.*
  - *To test the 10-inch panel LVDS display, execute the tool with `-d verdin-10inch-panel-lvds`.*
  - *To test the 10-inch panel DSI display, execute the tool with `-d verdin-10inch-panel-dsi`.*
  - *To test the MIPI panel (RM692C9) display, execute the tool with `-d verdin-mipi-panel`.*
  - *To test the MIPI-to-HDMI4K display, execute the tool with `-d verdin-mipi4k`.*

    - *To test the OS08A20 camera, execute the tool with `-d verdin`.*
    - *To test the AP1302 camera, execute the tool with `-d 15x15-ap1302`, `-d verdin-lt8912`.*
  - *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.
  - ***If the Trusty OS is used, to make the video playback work, see Section "Secure firmware loader" in the i.MX Android Security User's Guide (UG10158) to flash the keys related to the firmwareloader to ensure that the the VPU firmware can be loaded successfully. The relevant keys can be obtained from the `firmware_test_keys` directory where the image is located.***
6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, the command window displays information indicating that images are already flashed.
   Note:
   If the target device has a DOS partition table on it, the flash process fails for the first time. Push the reset key on the board and execute the flash script again.
7. Power off the board.
8. Change boot device to eMMC or SD card.
   - For i.MX 95 EVK and i.MX 95 15X15 EVK:
     - Change SW7 to switch the board back to 1010 (form 1-4 bit) to enter eMMC boot mode.
     - Change SW7 to switch the board back to 1011 (form 1-4 bit) to enter SD boot mode.
   - For i.MX 95 Verdin EVK:
     - Change SW2 to switch the board back to 1010 (form 1-4 bit) to enter eMMC boot mode.
     - Change SW2 to switch the board back to 1011 (form 1-4 bit) to enter SD boot mode.

10.4 Booting

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

Some specific U-Boot enviroment variables may need to be set in some boot scenarios to boot the MCU image or to get better display effect.

10.4.1 Booting with an MCU image

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

```
U-Boot > setenv append_bootargs androidboot.lpa.enable=1
 snd_pcm.max_alloc_per_card=134217728 pd_ignore_unused cma=600M
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
```

10.4.2 Booting with a display: HDMI 4K display

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

```
U-Boot > setenv append_bootargs androidboot.displaymode=4kp30
 androidboot.lcd_density=480
U-Boot > saveenv
```

Note:

If other boot arguments need to be appended to the U-Boot environment variable `bootargs`, set them together in one `setenv append_bootargs` command.

10.5 Board reboot

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

11 Note About the Source Code in the Document

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

Copyright 2025 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.

12 Revision History

Revision history

Document ID	Release date	Description
UG10157 v.android-15.0.0_1.2.0	11 April 2025	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8ULP, i.MX 8QuadMax, i.MX 8QuadXPlus GA release, and i.MX 95 Beta release.
UG10157 v.android-15.0.0_1.0.0	24 January 2025	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8ULP, i.MX 8QuadMax, i.MX 8QuadXPlus GA release, and i.MX 95 Beta release.

Revision history*...continued*

Document ID	Release date	Description
UG10157 v.android-14.0.0_2.2.0	18 October 2024	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8ULP, i.MX 8QuadMax, i.MX 8QuadXPlus GA release, i.MX 95 (A1 15x15) Alpha release, and i.MX 95 (A1 19x19) Beta release.
UG10157 v.android-14.0.0_2.0.0	9 August 2024	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Plus, i.MX 8M Quad, i.MX 8ULP, i.MX 8QuadMax, i.MX 8QuadXPlus GA release, and i.MX 95 Alpha release. Updated the document ID.
AQSUG v.android-14.0.0_1.2.0	19 April 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 8Quad XPlus GA release.
AQSUG v.android-14.0.0_1.0.0	6 Feburary 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 8Quad XPlus GA release.
AQSUG v.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 8Quad XPlus GA release.
AQSUG v.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 8 QuadXPlus GA release.
AQSUG v.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 8 QuadXPlus GA release.
AQSUG v.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 8 QuadXPlus GA release.
AQSUG v.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 8 QuadXPlus GA release.
AQSUG v.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.
AQSUG v.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.
AQSUG v.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.
AQSUG v.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.
AQSUG v.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.
AQSUG v.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.
AQSUG v.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.
AQSUG v.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.

Revision history*...continued*

Document ID	Release date	Description
AQSUG v.android-10.0.0_2.0.0	05/2020	i.MX 8M Mini, i.MX 8M Nano, i.MX 8M Quad, i.MX 8Quad Max, and i.MX 8QuadXPlus GA release.
AQSUG v.android-10.0.0_2.1.0	04/2020	i.MX 8M Plus Alpha and i.MX 8QuadXPlus Beta release.
AQSUG v.android-10.0.0_1.0.0	03/2020	Deleted the Android 10 image.
AQSUG v.android-10.0.0_1.0.0	02/2020	i.MX 8M Mini, i.MX 8M Quad, i.MX 8QuadMax, and i.MX 8 QuadXPlus GA release.
AQSUG v.P9.0.0_2.0.0-ga	08/2019	Updated the location of the SCFW porting kit.
AQSUG v.P9.0.0_2.0.0-ga	04/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.
AQSUG v.P9.0.0_1.0.0-ga	01/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.
AQSUG v.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.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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

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