

UG10177

Android Automotive Quick Start Guide

Rev. automotive-16.0.0_1.1.0 —
2 March 2026

User guide

Document information

Information	Content
Keywords	Android, Automotive, i.MX, automotive-16.0.0_1.1.0, UG10177
Abstract	This document guides you through the processes of downloading and running this release package. It only explains how to download and run the default release image with default configuration.



1 Overview

This document guides you through the processes of downloading and running this release package. It only explains how to download and run the default release image with default configuration. For details on using the release package, see the *Android Automotive User's Guide* (UG10176) 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 8QuadMax (Silicon Revision B0)
- i.MX 95 19x19 (Silicon Revision B0)

Supported boards:

- i.MX 8QuadMax MEK Board and Platform
- i.MX 95 EVK/Verdin Boards

3 Working with the i.MX 8QuadXPlus/8QuadMax MEK Board

3.1 Board hardware

The figures below show the different components of the i.MX 8QuadMax/8QuadXPlus MEK boards.

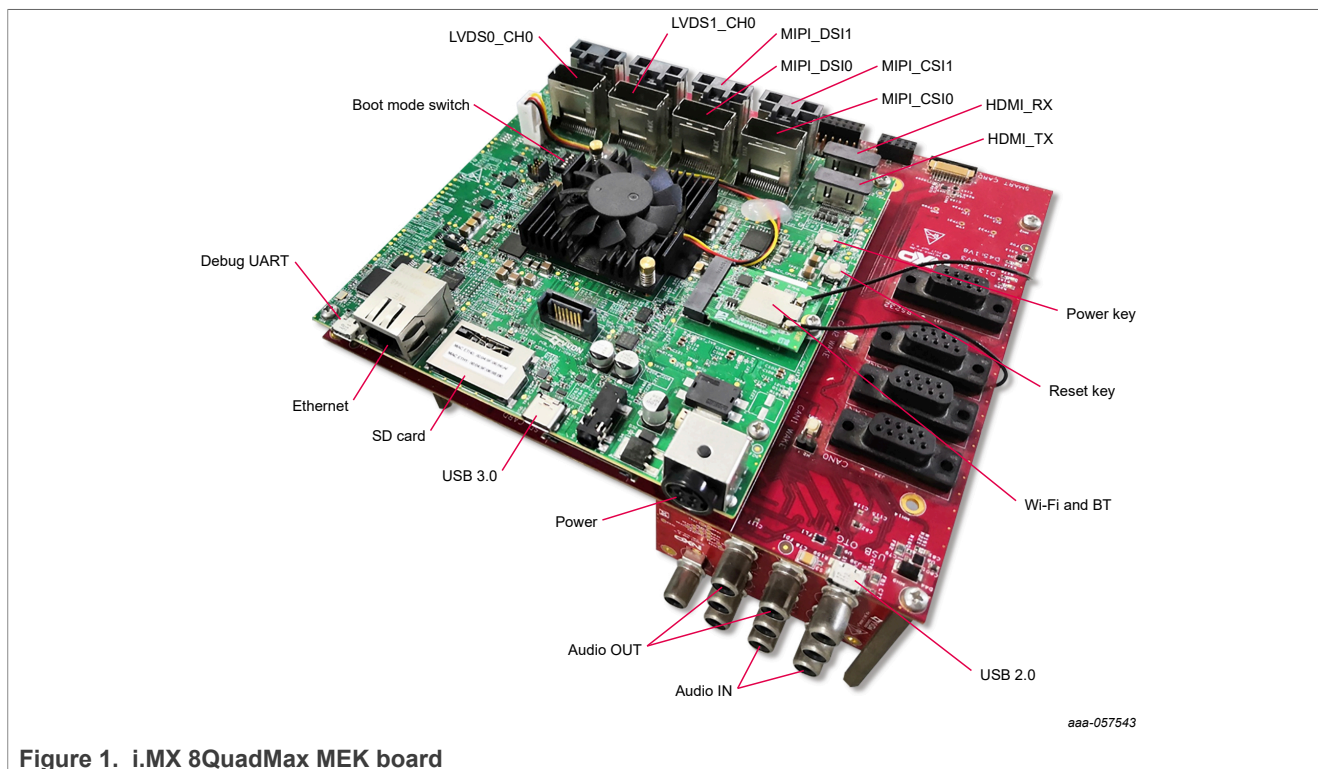


Figure 1. i.MX 8QuadMax MEK board

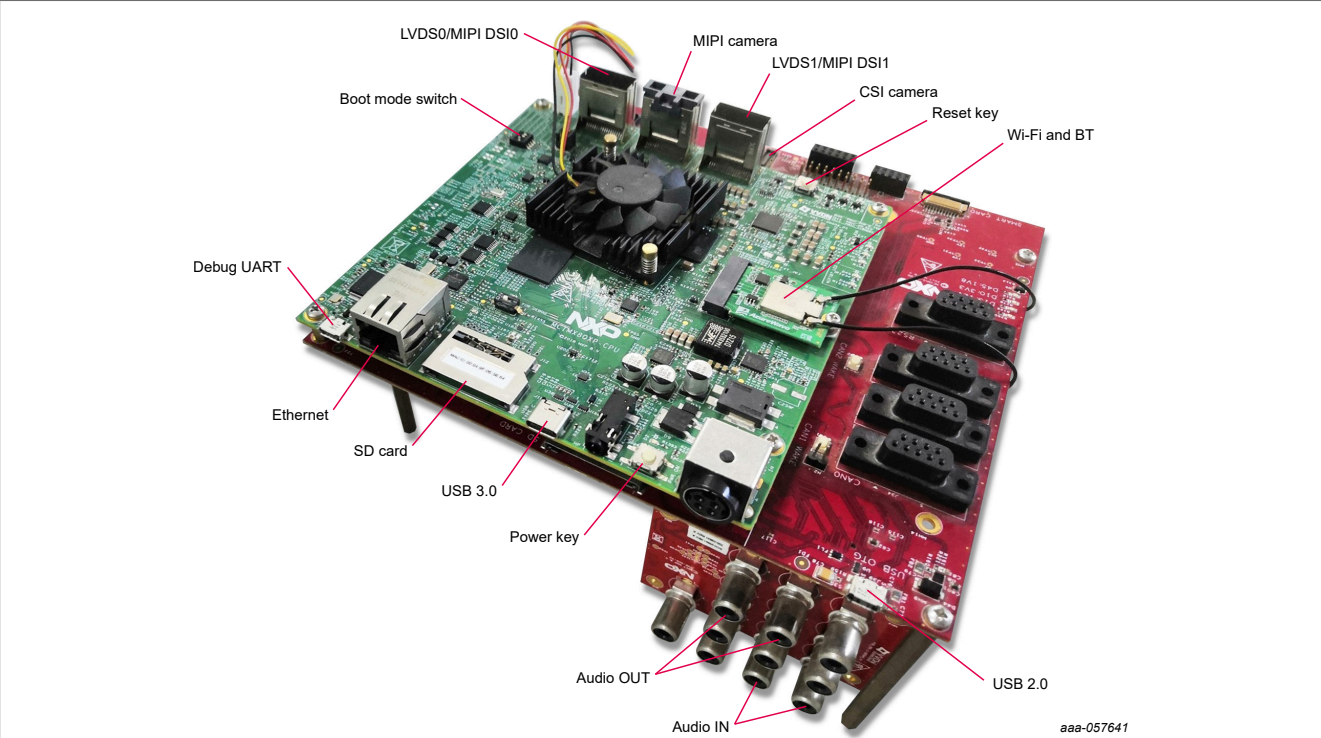


Figure 2. i.MX 8QuadXPlus MEK board



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



Figure 4. i.MX mini SAS cable with LVDS-to-HDMI adapter

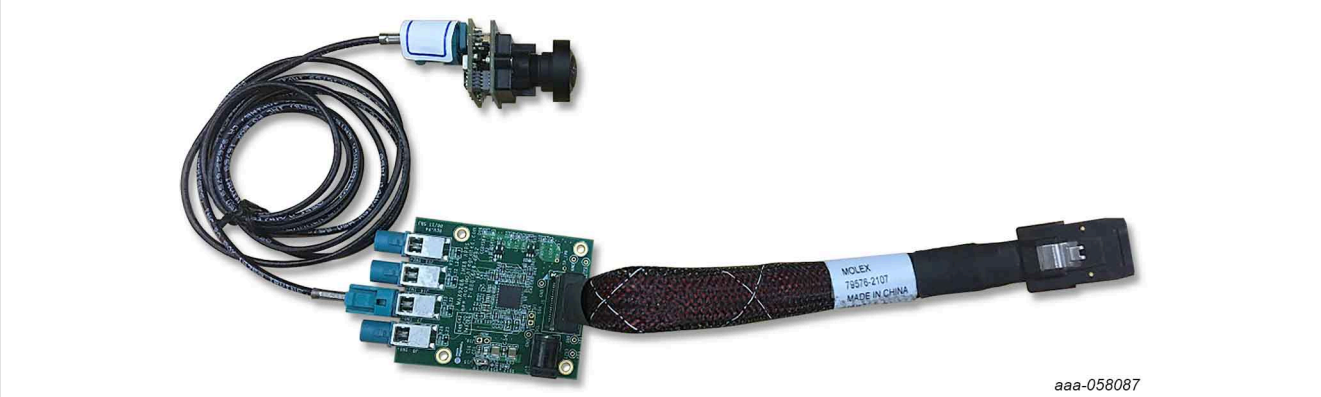


Figure 5. i.MX rearview camera (MAX9286)

Note:

- To use the i.MX rearview camera (MAX9286), ensure that the jumper J15 on the camera converter board is installed.
- i.MX 8QuadMax MEK
 - To test the display, connect the "LVDS1_CH0" port to the LVDS-to-HDMI adapter with the i.MX mini SAS cable.
 - To test the rearview camera, connect the "MIPI_CSI0" port with the i.MX MAX9286 MIPI camera.
 - To test the display, connect the "MIPI_DSI1" port to the MIPI-to-HDMI adapter with the i.MX mini SAS cable. Android images must be downloaded with multi-display (MD) support.
- i.MX 8QuadXPlus MEK
 - To test the display, connect the "LVDS0" port to the LVDS-to-HDMI adapter with the i.MX mini SAS cable.
 - To test the rearview camera, connect the "MIPI Camera" port with the i.MX MAX9286 MIPI camera.
 - To test the display, connect the "MIPI_DSI1" port to the MIPI-to-HDMI adapter with the i.MX mini SAS cable.
- "MCIMX8-8X-BB" is required for:
 - The *car* image for the UART interface to control the EVS function.
 - Audio media input/output.

3.2 Board images

There are two sets of images: `automotive-16.0.0_1.1.0_image_8qmek_car.tar.gz` and `automotive-16.0.0_1.1.0_image_8qmek_car2.tar.gz`. For more details related to EVS support and its differences between *car* and *car2* images and Dynamic partitions, see the updatable Apex described in the Android documentation (<https://source.android.com/docs/core/ota/apex>). The following table lists their differences with `8qmek_car` and `8qmek_car2` representing these two packages.

Table 1. Image differences

	8qmek_car	8qmek_car2
Exterior View System (EVS) function enabled in Cortex-M core	Y	N
EVS function enabled in Cortex-A core	Y	Y
Supports dynamic partition	Y	Y
Supports updatable apex	Y	Y

The table below describes the location in the board partitions of the software images in `automotive-16.0.0_1.1.0_image_8qmek_car.tar.gz`.

Table 2. Board images

Image name	Download target
<code>spl-imx8qm.bin</code>	0 kB offset of eMMC boot0 partition for i.MX 8QuadMax MEK.
<code>spl-imx8qm-secure-unlock.bin</code>	0 kB offset of eMMC boot0 partition for i.MX 8QuadMax MEK.
<code>spl-imx8qxp.bin</code>	32 kB offset of eMMC boot0 partition for i.MX 8QuadXPlus MEK with silicon revision B0 chip.
<code>spl-imx8qxp-secure-unlock.bin</code>	32 kB offset of eMMC boot0 partition for i.MX 8QuadXPlus MEK with silicon revision B0 chip.
<code>spl-imx8qxp-c0.bin</code>	32 kB offset of eMMC boot0 partition for i.MX 8QuadXPlus MEK with silicon revision C0 chip.
<code>bootloader-imx8qm.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8QuadMax.
<code>bootloader-imx8qm-secure-unlock.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 8QuadMax.

Table 2. Board images...continued

Image name	Download target
bootloader-imx8qxp.img	bootloader_a and bootloader_b partitions for i.MX 8QuadXPlus MEK with silicon revision B0 chip.
bootloader-imx8qxp-secure-unlock.img	bootloader_a and bootloader_b partitions for i.MX 8QuadXPlus MEK with silicon revision B0 chip.
bootloader-imx8qxp-c0.img	bootloader_a and bootloader_b partitions for i.MX 8QuadXPlus MEK with silicon revision C0 chip.
u-boot-imx8qm-mek-uuu.imx	Bootloader used by UUU for i.MX 8QuadMax MEK board. It is not flashed to MMC.
u-boot-imx8qxp-mek-uuu.imx	Bootloader used by UUU for i.MX 8QuadXPlus MEK board with silicon revision B0 chip. It is not flashed to MMC.
u-boot-imx8qxp-mek-c0-uuu.imx	Bootloader used by UUU for i.MX 8QuadXPlus MEK board with silicon revision C0 chip. It is not flashed to MMC.
boot.img	boot_a and boot_b partitions. This is the AOSP boot image.
vendor_boot.img	vendor_boot_a and vendor_boot_b partitions.
partition-table.img	Program to the first 17 kB, and then back up to the last 17 kB of the boot storage. GPT table image for 16 GB boot storage.
partition-table-28GB.img	Program to the first 17 kB, and then back up to the last 17 kB of the boot storage. GPT table image for 32 GB boot storage.
system.img	Logical partitions system_a and system_b in super partition.
system_ext.img	Logical partitions system_ext_a and system_ext_b in super partition.
system_dtkm.img	Logical partitions system_dtkm_a and system_dtkm_b in super partition.
vendor.img	Logical partitions vendor_a and vendor_b in super partition.
vendor_dtkm.img	Logical partitions vendor_dtkm_a and vendor_dtkm_b in super partition.
product.img	Logical partitions product_a and product_b in super partition.
super.img	Super partition.
vbmeta-imx8qm-md.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax to support multiple displays.
vbmeta-imx8qm-md-revd.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax MEK board rev.D/E to support multiple displays.
vbmeta-imx8qm.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax MEK to support LVDS-to-HDMI display.
vbmeta-imx8qm-revd.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax MEK board rev.D/E to support the LVDS-to-HDMI display.
vbmeta-imx8qxp.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadXPlus to support LVDS-to-HDMI display.
vbmeta-imx8qm-sof.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax to support SOF on DSP.
vbmeta-imx8qm-sof-revd.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax MEK board rev.D/E to support SOF on DSP.
vbmeta-imx8qxp-sof.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadXPlus to support SOF on DSP.
dtbo-imx8qm.img	dtbo_a and dtbo_b partitions for i.MX 8QuadMax MEK board.
dtbo-imx8qm-revd.img	dtbo_a and dtbo_b partitions for i.MX 8QuadMax MEK board revision D/E.

Table 2. Board images...continued

Image name	Download target
dtbo-imx8qm-md.img	dtbo_a and dtbo_b partitions for i.MX 8QuadMax to support multiple displays.
dtbo-imx8qm-md-revd.img	dtbo_a and dtbo_b partitions for i.MX 8QuadMax MEK board rev. D/E to support multiple displays.
dtbo-imx8qm-sof.img	dtbo_a and dtbo_b partitions for i.MX 8QuadMax to support SOF on DSP.
dtbo-imx8qm-sof-revd.img	dtbo_a and dtbo_b partitions for i.MX 8QuadMax MEK board rev. D/E to support SOF on DSP.
dtbo-imx8qxp.img	dtbo_a and dtbo_b partitions for i.MX 8QuadXPlus.
dtbo-imx8qxp-sof.img	dtbo_a and dtbo_b partitions for i.MX 8QuadXPlus to support SOF on DSP.
rpmb_key_test.bin	Prebuilt test RPMB key, which can be used to set the RPMB key as fixed 32 bytes 0x00.
testkey_public_rsa4096.bin	Prebuilt AVB public key. It is extracted from the default AVB private key.

3.3 Flashing board images

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

UUU source code and binary file are released on GitHub: [UUU release page on GitHub](#).

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

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

For this release, these two scripts are validated on UUU 1.5.201 version. Download the corresponding version from GitHub:

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

Because the two script files 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 (1-6 bit) to enter serial download mode for i.MX 8QuadMax.
 - Change the board's SW2 (boot mode) to 1000 (1-4 bit) to enter serial download mode for i.MX 8QuadXPlus.
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 i.MX 8QuadMax/8QuadXPlus MEK board: USB-to-UART, USB 2.0, and USB 3.0.
- The USB-to-UART is known as debug UART, which can be used to watch the log of hardware boot processing.
- USB 2.0 is USB Host and USB 3.0 is USB OTG.

4. Decompress `release_package/automotive-16.0.0_1.1.0_image_8qmek_car.tar.gz` or `release_package/automotive-16.0.0_1.1.0_image_8qmek_car2.tar.gz`, which contains the image files and UUU scripts.

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 Android Auto images on the i.MX 8QuadMax/8QuadXPlus MEK board, related options are described as follows.

Table 3. Options for `uuu_imx_android_flash` tool

Option	Description
<code>-h</code>	Displays the help information of this tool.
<code>-f soc_name</code>	Specifies the SoC information. For i.MX 8QuadMax, it should be <code>imx8qm</code> . For i.MX 8QuadXPlus, it should be <code>imx8qxp</code> . This option is mandatory.
<code>-a</code>	Only flashes slot A. If this option and <code>-b</code> option are not used, slots A and B are both flashed.
<code>-b</code>	Only flashes slot B. If this option and <code>-a</code> option are not used, slots A and B are both flashed.
<code>-m</code>	Flashes the MCU image.
<code>-u uboot_feature</code>	Flashes U-Boot or SPL&bootloader image with <code>uboot_feature</code> in their names. For i.MX 8QuadMax MEK, it can be <code>secure-unlock</code> . For i.MX 8QuadXPlus MEK, it can be <code>c0</code> or <code>secure-unlock</code> . If this option is not used, the default <code>spl-imx8qm.bin</code> and <code>bootloader-imx8qm.img</code> are flashed for i.MX 8QuadMax MEK, and the default <code>spl-imx8qxp.bin</code> and <code>bootloader-imx8qxp.img</code> are flashed for i.MX 8QuadXPlus MEK.
<code>-d dtb_feature</code>	Flashes dtbo and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 8QuadMax MEK, it can be <code>md</code> . For i.MX 8QuadXPlus MEK, do not use this option. If this option is not used, default <code>dtbo-imx8qm.img</code> and <code>vbmeta-imx8qm.img</code> are flashed for i.MX 8QuadMax MEK.
<code>-e</code>	Erases user data after images are flashed.
<code>-D directory</code>	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash</code> , it must be followed with an absolute path. If this option is not used, images in the current working directory are flashed.
<code>-daemon</code>	Runs UUU in Daemon mode. This option is used to flash multiple boards of the same type.
<code>-i</code>	If the script is executed with this option, no image is flashed. The script loads U-Boot to RAM and executes to fastboot mode. This option is used for development.
<code>-dryrun</code>	Only generates a UUU script but not executes UUU with this script.

- On the Linux system, open the shell terminal. For example, you can execute a command as follows:
 - For i.MX 8QuadMax, board rev. C:

```
> sudo ./uuu_imx_android_flash.sh -f imx8qm -e
```

- For i.MX 8QuadMax, board rev. D:

```
> sudo ./uuu_imx_android_flash.sh -f imx8qm -e -d revd
```

- For i.MX 8QuadXPlus (Silicon Revision C0):

```
> sudo ./uuu_imx_android_flash.sh -f imx8qxp -e -u c0
```

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

- For i.MX 8QuadMax, board rev. C:

```
> .\uuu_imx_android_flash.bat -f imx8qm -e
```

- For i.MX 8QuadMax, board rev. D:

```
> .\uuu_imx_android_flash.bat -f imx8qm -e -d revd
```

- For i.MX 8QuadXPlus (Silicon Revision C0):

```
> .\uuu_imx_android_flash.bat -f imx8qxp -e -u c0
```

When the command above is executed, the default images are flashed into eMMC both slot A and slot B for i.MX 8QuadMax and all user data are erased.

Note:

- 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.

- To test the demonstration implementation of secure unlock, execute the tool with `-u secure-unlock`. For secure unlocking details, see the *i.MX Android Security User's Guide (UG10158)*.
 - To enable physical HDMI output (J6) on i.MX 8QuadMax MEK with images in `automotive-16.0.0_1.1.0_image_8qmek_car2.tar.gz`, execute the tool with `-u md -d md`.
6. Wait for the `uuu_imx_android_flash` execution to complete. If there is no error, the command line interface shows that the images are already flashed.
 7. Power off the board.
 8. Change boot device as eMMC.
 - Change SW2 to switch the board back to 000100 (1-6 bit) to enter eMMC boot mode for i.MX 8QuadMax.
 - Change SW2 to switch the board back to 0100 (1-4 bit) to enter eMMC boot mode for i.MX 8QuadXPlus.

4 Working with the i.MX 95 EVK Board

4.1 Board hardware

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

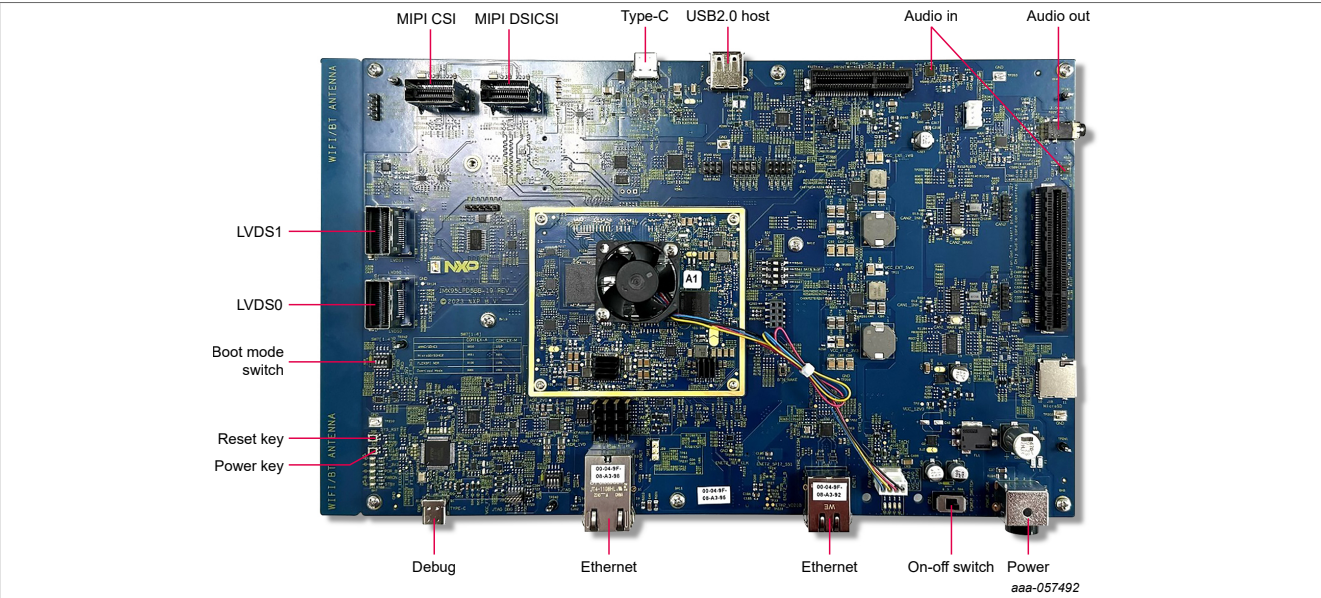


Figure 6. i.MX 95 EVK

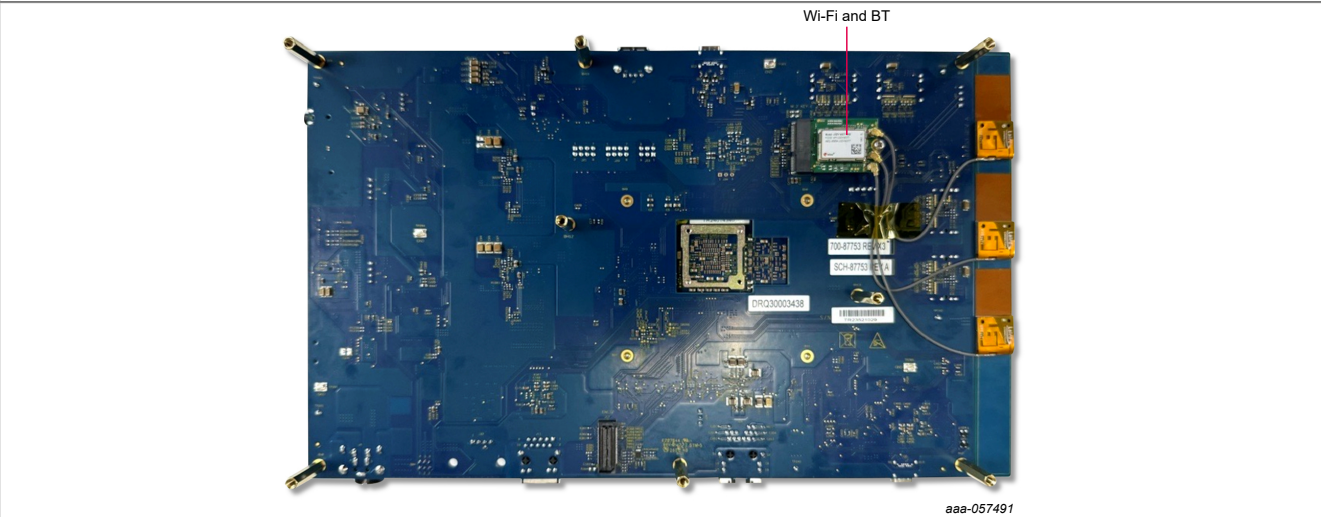


Figure 7. i.MX 95 EVK board back view

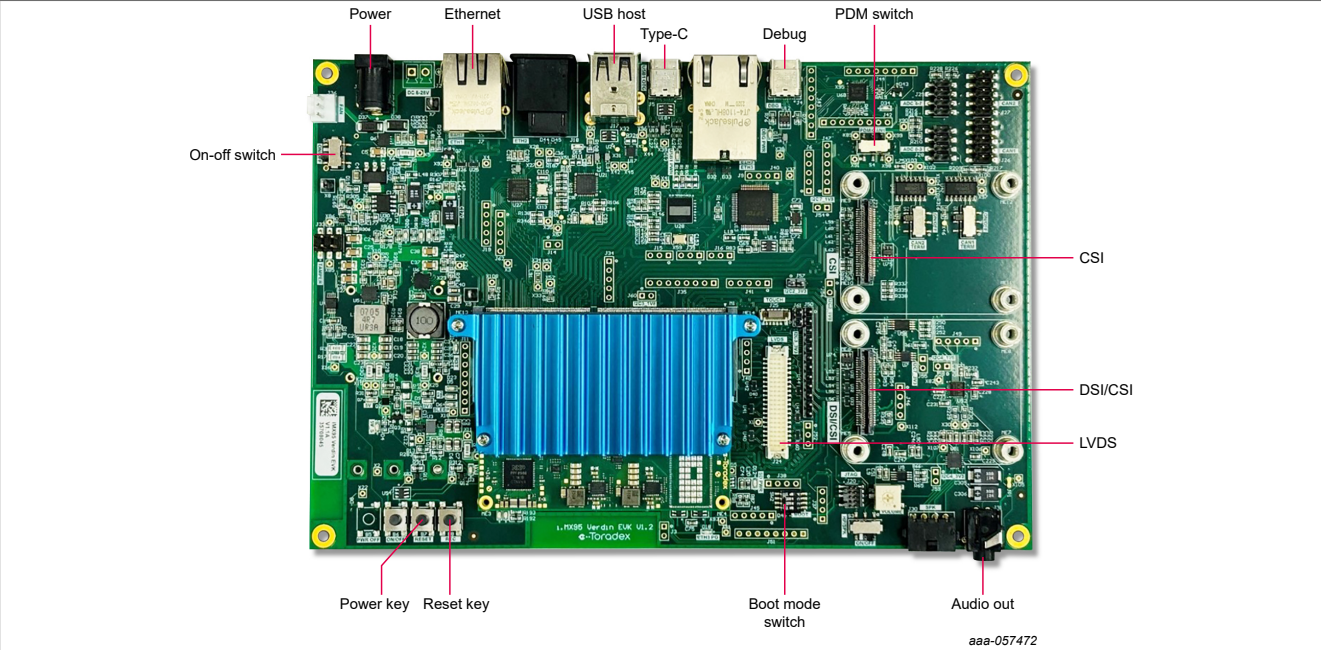


Figure 8. i.MX 95 Verdin

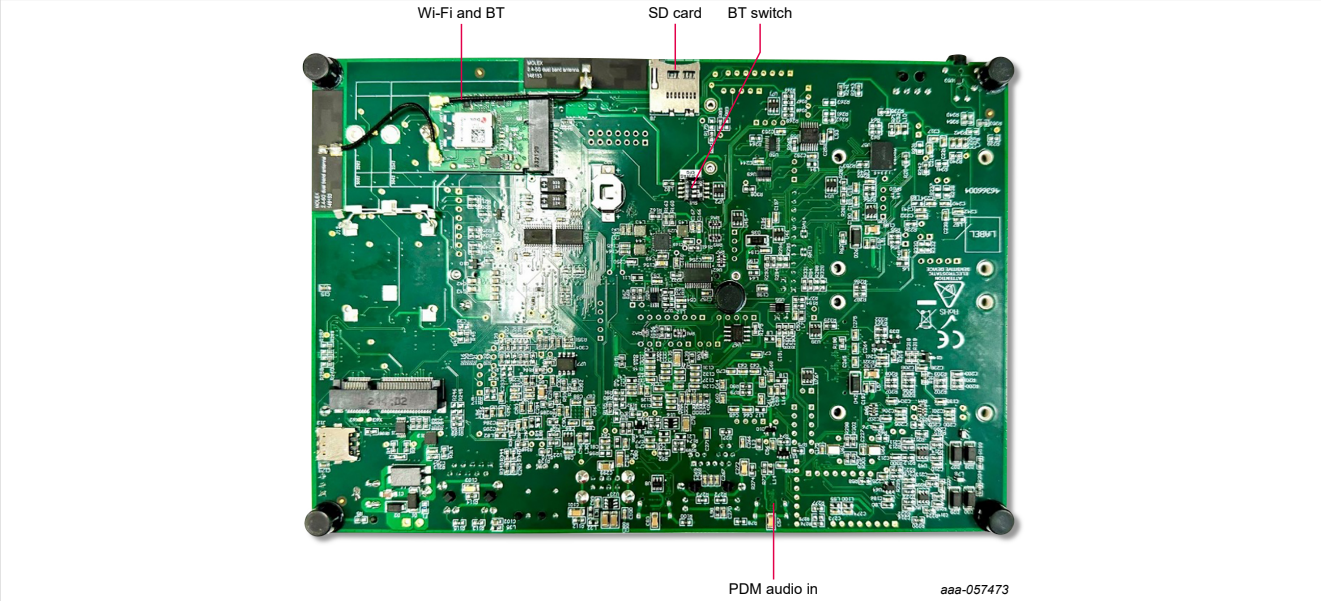


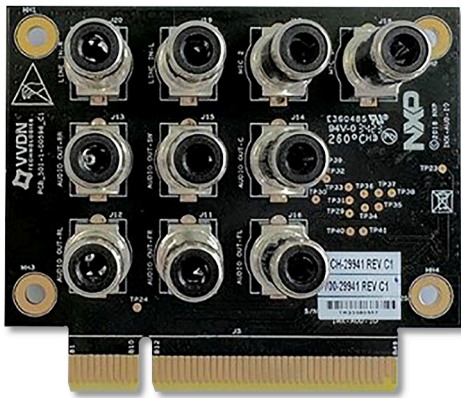
Figure 9. i.MX 95 Verdin board back view



Figure 10. i.MX mini SAS cable with DSI-to-HDMI adapter



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



aaa-057481

Figure 12. AUDIO-IO board

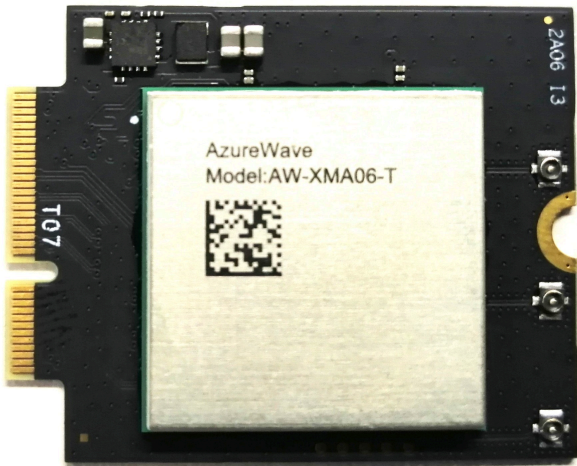


Figure 13. AW693 (AzureWave 693)



aaa-057501

Figure 14. PCIE9098 (U-Blox JODY-W3)

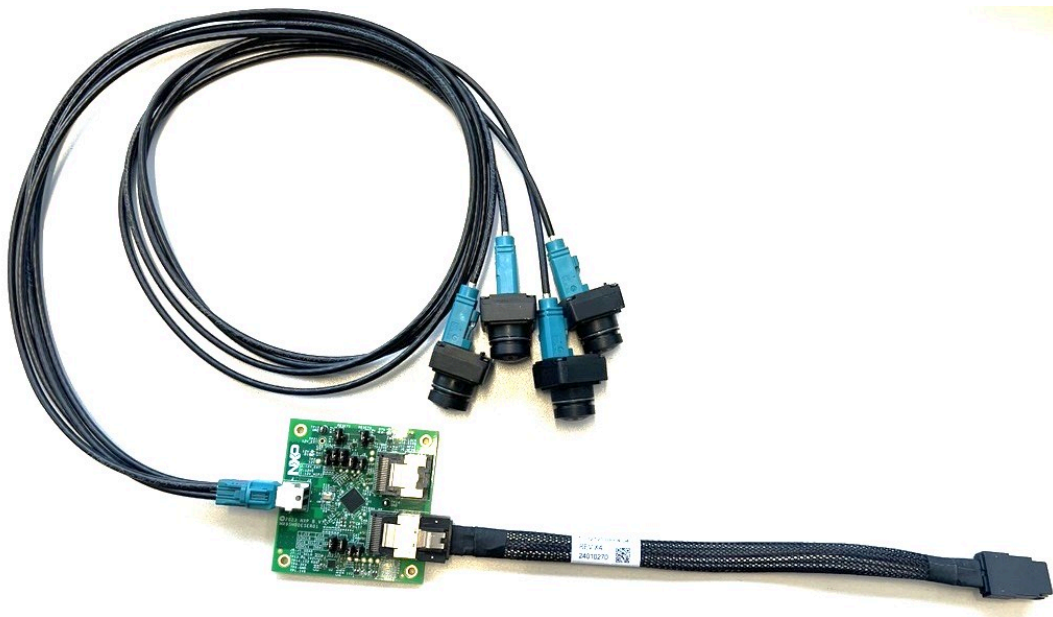


Figure 15. i.MX rearview camera (MAX96724, OX03C10 camera sensors)



Figure 16. i.MX CSI MIPI Camera AP1302

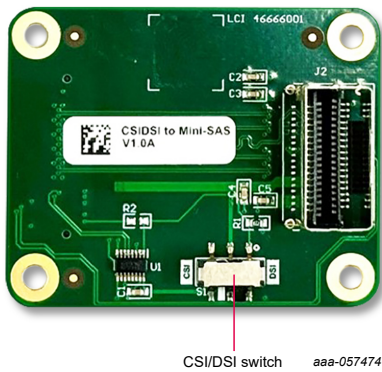
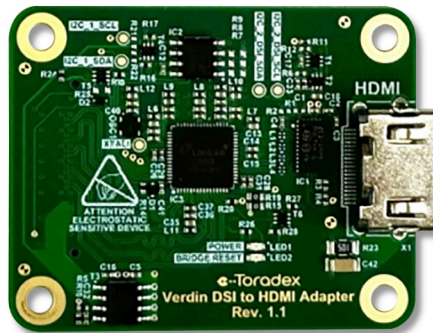


Figure 17. i.MX 95 Verdin CSI/DSI to Mini-SAS adapter



aaa-057461

Figure 18. i.MX 95 Verdin DSI-to-HDMI adapter

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_DSICSI" 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 AUDIO-IO board, connect the AUDIO-IO board to the J27 connector.
- Connect the PCIE9098 (U-Blox JODY-W3) Wi-Fi&BT M.2 expansion card to the J24 connector to enable the Wi-Fi and Bluetooth to work.
- Connect the PCIE AW693 Wi-Fi&BT M.2 expansion card to the J24 connector to enable the Wi-Fi and Bluetooth to work.
- To test the camera on i.MX 95 EVK, connect the deserializer MAX96724 with Camera sensors OX03C10 to the i.MX "MIPI_CSI" port. To use the i.MX rearview camera (MAX96724), ensure that the jumper J5 on the camera converter board is installed in position 1-2.
- To test the camera on i.MX 95 EVK, connect the i.MX CSI MIPI Camera AP1302 to the "MIPI CSI" port.
- To test the MIPI-to-HDMI (ADV7535) display on the i.MX 95 Verdin, 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 board. Then switch the "CSI/DSI Switch" to DSI.
- To test the MIPI-to-HDMI (LT8912) display on i.MX 95 Verdin, connect the Verdin DSI-to-HDMI adapter to the "DSI/CSI" port.

4.2 Board images

There are two sets of images: `automotive-16.0.0_1.1.0_image_95evk_car.tar.gz` and `automotive-16.0.0_1.1.0_image_95evk_car2.tar.gz`. For more details related to EVS support and its differences between `car` and `car2` images and Dynamic partitions, see the updatable Apex described in the Android documentation (<https://source.android.com/docs/core/ota/apex>). The following table lists their differences with `95evk_car` and `95evk_car2` representing these two packages.

Table 4. Image differences

	95evk_car	95evk_car2
Exterior View System (EVS) function enabled in Cortex-M core	Y	N
EVS function enabled in Cortex-A core	Y	Y
Supports updatable Apex	Y	Y

The table below describes the location in the board partitions of the software images in `automotive-16.0.0_1.1.0_image_95evk_car.tar.gz`.

Table 5. Board images

Image name	Download target
<code>spl-imx95.bin</code>	0 kB offset of eMMC boot0 partition for i.MX 95 EVK.
<code>spl-imx95-secure-unlock.bin</code>	0 kB offset of eMMC boot0 partition for i.MX 95 EVK.
<code>spl-imx95-verdin.bin</code>	0 kB offset of eMMC boot0 partition for i.MX 95 Verdin.
<code>bootloader-imx95.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 95 EVK.
<code>bootloader-imx95-secure-unlock.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 95 EVK.
<code>bootloader-imx95-verdin.img</code>	<code>bootloader_a</code> and <code>bootloader_b</code> partitions for i.MX 95 Verdin.
<code>u-boot-imx95-evk-uuu.imx</code>	Bootloader used by UUU for i.MX 95 EVK board. It is not flashed to MMC.
<code>u-boot-imx95-verdin-uuu.imx</code>	Bootloader used by UUU for i.MX 95 Verdin board. It is not flashed to MMC.
<code>boot.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions. This is the AOSP boot image.
<code>boot-imx.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions. Boot image built with the i.MX kernel tree for debugging.
<code>vendor_boot.img</code>	<code>vendor_boot_a</code> and <code>vendor_boot_b</code> partitions.
<code>partition-table.img</code>	Program to the first 17 kB, and then back up to the last 17 kB of the boot storage. GPT table image for 16 GB boot storage.
<code>partition-table-28GB.img</code>	Program to the first 17 kB, and then back up to the last 17 kB of the boot storage. GPT table image for 32 GB boot storage.
<code>system.img</code>	Logical partitions <code>system_a</code> and <code>system_b</code> in super partition.
<code>system_ext.img</code>	Logical partitions <code>system_ext_a</code> and <code>system_ext_b</code> in super partition.
<code>system_dtkm.img</code>	Logical partitions <code>system_dtkm_a</code> and <code>system_dtkm_b</code> in super partition.
<code>vendor.img</code>	Logical partitions <code>vendor_a</code> and <code>vendor_b</code> in super partition.
<code>vendor_dtkm.img</code>	Logical partitions <code>vendor_dtkm_a</code> and <code>vendor_dtkm_b</code> in super partition.
<code>product.img</code>	Logical partitions <code>product_a</code> and <code>product_b</code> in super partition.
<code>super.img</code>	Super partition.
<code>vbmeta-imx95-lvds0.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions for i.MX 95 EVK to support LVDS0 output.
<code>vbmeta-imx95.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions for i.MX 95 EVK to support the MIPI-to-HDMI display.
<code>vbmeta-imx95-ap1302.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions for i.MX 95 EVK to support the MIPI-to-HDMI display and AP1302 camera.
<code>vbmeta-imx95-lvds0.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions for i.MX 95 EVK to support LVDS0 output and four OX03C10 camera sensors.
<code>vbmeta-imx95-lvds0-ap1302.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions for i.MX 95 EVK to support LVDS0 output and AP1302 camera sensor.
<code>vbmeta-imx95-mipi-lvds1.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions for i.MX 95 EVK to support dual displays (MIPI and LVDS1) and four OX03C10 camera sensors.
<code>vbmeta-imx95-mipi-lvds1-ap1302.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions for i.MX 95 EVK to support dual displays (MIPI and LVDS1) and APL1302 camera sensor.

Table 5. Board images...continued

Image name	Download target
vbmeta-imx95-verdin.img	vbmeta_a and vbmeta_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display and four OX03C10 camera sensors.
vbmeta-imx95-verdin-adv7535.img	vbmeta_a and vbmeta_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display (ADV7535 converter) and four OX03C10 camera sensors.
vbmeta-imx95-verdin-adv7535-ap1302.img	vbmeta_a and vbmeta_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display (ADV7535 converter) and AP1302 camera sensor.
vbmeta-imx95-verdin-ap1302.img	vbmeta_a and vbmeta_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display and AP1302 camera sensor.
dtbo-imx95.img	dtbo_a and dtbo_b partitions for i.MX 95 EVK to support the MIPI-to-HDMI display and four OX03C10 camera sensors.
dtbo-imx95-ap1302.img	dtbo_a and dtbo_b partitions for i.MX 95 EVK to support the MIPI-to-HDMI display and AP1302 camera.
dtbo-imx95-lvds0.img	dtbo_a and dtbo_b partitions for i.MX 95 EVK to support LVDS0 output and four OX03C10 camera sensors.
dtbo-imx95-lvds0-ap1302.img	dtbo_a and dtbo_b partitions for i.MX 95 EVK to support LVDS0 output and AP1302 camera.
dtbo-imx95-mipi-lvds1.img	dtbo_a and dtbo_b partitions for i.MX 95 EVK to support dual displays (MIPI and LVDS1).
dtbo-imx95-mipi-lvds1-ap1302.img	dtbo_a and dtbo_b partitions for i.MX 95 EVK to support dual displays (MIPI and LVDS1) and AP1302 camera.
dtbo-imx95-verdin.img	dtbo_a and dtbo_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display and four OX03C10 camera sensors.
dtbo-imx95-verdin-adv7535.img	dtbo_a and dtbo_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display (ADV7535 converter) and four OX03C10 camera sensors.
dtbo-imx95-verdin-adv7535-ap1302.img	dtbo_a and dtbo_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display (ADV7535 converter) and AP1302 camera sensor.
dtbo-imx95-verdin-ap1302.img	dtbo_a and dtbo_b partitions for i.MX 95 Verdin to support the MIPI-to-HDMI display and AP1302 camera.
testkey_public_rsa4096.bin	Prebuilt AVB public key. It is extracted from the default AVB private key.

4.3 Flashing board images

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

UUU source code and binary file are released on GitHub: [UUU release page on GitHub](#).

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

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

For this release, these two scripts are validated on UUU 1.5.201 version. Download the corresponding version from GitHub:

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

Because the two script files 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 SW7 (boot mode) to 1001 (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.

Note: The debug PORT can be used to watch the logs of the hardware boot processing.

4. Decompress `release_package/automotive-16.0.0_1.1.0_image_95evk_car.tar.gz` or `release_package/automotive-16.0.0_1.1.0_image_95evk_car2.tar.gz`, which contains the image files and UUU scripts.
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 Android Automotive images on i.MX 95 EVK and Verdin boards, related options are described as follows.

Table 6. 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 95 EVK and Verdin boards, it should be <code>imx95</code> . This option is mandatory .
-a	Only flashes slot A. If this option and -b option are not used, slots A and B are both flashed.
-b	Only flashes slot B. If this option and -a option are not used, slots A and B are both flashed.
-m	Flashes the MCU image.
-u uboot_feature	Flashes U-Boot or SPL and bootloader image with <code>uboot_feature</code> in their names. It can be <code>secure-unlock</code> for EVK board, or <code>verdin</code> for Verdin board. If this option is not used, the default <code>spl-imx95.bin</code> and <code>bootloader-imx95.img</code> are flashed for i.MX 95 B0 EVK board.
-d dtb_feature	Flashes dtbo and vbmeta images with <code>dtb_feature</code> in their names. For i.MX 95 EVK, it can be <code>lvds0</code> or <code>mipi-lvds1</code> . For i.MX 95 Verdin, it can be <code>verdin</code> or <code>verdin-adv7535</code> . If this option is not used, the default <code>dtbo-imx95.img</code> and <code>vbmeta-imx95.img</code> are flashed for i.MX 95 EVK.
-e	Erases user data after the images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. For <code>uuu_imx_android_flash</code> , it must be followed with an absolute path. If this option is not used, the images in the 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 loads U-Boot to RAM and executes to fastboot mode. This option is used for development.

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

Option	Description
<code>-dryrun</code>	Only generates a UUU script but not executes UUU with this script.

- On the Linux system, open the shell terminal. For example, you can execute a command as follows:
 - For i.MX 95 EVK, MIPI-to-HDMI (ADV7353 converter) display output:

```
> sudo ./uuu_imx_android_flash.sh -f imx95 -e
```

- For i.MX 95 Verdin, MIPI-to-HDMI (LT8912 converter) display output:

```
> sudo ./uuu_imx_android_flash.sh -f imx95 -e -u verdin -d verdin
```

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

- For i.MX 95 EVK, MIPI-to-HDMI (ADV7353 converter) display output:

```
> .\uuu_imx_android_flash.bat -f imx95 -e
```

- For i.MX 95 Verdin, MIPI-to-HDMI (LT8912 converter) display output:

```
> .\uuu_imx_android_flash.bat -f imx95 -e -u verdin -d verdin
```

When the command above is executed, the default images are flashed into eMMC both slot A and slot B for i.MX 95 and all user data are erased.

Note:

- 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.

- To test the demonstration implementation of secure unlock, execute the tool with `-u secure-unlock`. For secure unlocking details, see the *i.MX Android Security User's Guide (UG10158)*.

- Wait for the `uuu_imx_android_flash` execution to complete. If there is no error, the command line interface shows that the images are already flashed.
- Power off the board.
- Change boot device as eMMC.
 - Change SW7 to switch the board back to 1010 (form 1-4 bit) to enter eMMC boot mode.

5 Acronyms

Table 7. Acronyms

Acronym	Description
AOSP	Android Open Source Project
BT	Bluetooth
CST	(NXP) Code Signing Tool
eMMC	Embedded Multi-Media Card
EVK	Evaluation Kit
EVS	Android Exterior View System
GAS	Google Automotive Services
GCC	GNU Compiler collection

Table 7. Acronyms...continued

Acronym	Description
GPT	GUID partition table
HVAC	Heating, ventilation, and air conditioning
MEK	Multisensory Enablement Kit
OS	Operating system
PC	Personal (host) computer
SoC	System on Chip
SPL	U-Boot Secondary Program Loader
OTA	Over-The-Air programming
SOF	Sound Open Firmware
U-Boot	Universal Boot Loader

6 Note About the Source Code in the Document

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

Copyright 2026 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.

7 Revision History

This table provides the revision history.

Table 8. Revision history

Document ID	Release date	Description
UG10177 v.automotive-16.0.0_1.1.0	2 March 2026	i.MX 8QuadMax MEK (Silicon Revision B0), i.MX 95 EVK (Silicon 19x19, Revision B0), and i.MX 95 Verdin (Silicon 19x19, Revision B0) GA release

Table 8. Revision history...continued

Document ID	Release date	Description
UG10177 v.automotive-15.0.0_2.1.0	14 October 2025	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release, i.MX 95 EVK (Silicon 19x19, Revision A1, B0) and i.MX 95 Verdin (Silicon 19x19, Revision A1, B0) Beta
UG10177 v.automotive-15.0.0_1.3.0	21 July 2025	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release, i.MX 95 EVK (Silicon Revision A1 19x19) and i.MX 95 Verdin (Silicon Revision A1 19x19) Beta
UG10177 v.automotive-15.0.0_1.1.0	15 May 2025	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release, i.MX 95 EVK (Silicon Revision A1 19x19) Beta, and i.MX 95 Verdin (Silicon Revision A1 19x19) Experimental
UG10177 v.automotive-14.0.0_2.3.0	23 January 2025	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release, i.MX 95 EVK (Silicon Revision A1 19x19) Beta, and i.MX 95 Verdin (Silicon Revision A1 19x19) Experimental
UG10177 v.automotive-14.0.0_2.1.0	7 November 2024	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release, i.MX95 EVK (Silicon Revision A1 19x19) Alpha (EAR)
AAQSG_14.0.0_1.1.0	20 June 2024	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-13.0.0_2.3.0	4 January 2024	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-13.0.0_2.1.0	10/2023	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-13.0.0_1.3.0	07/2023	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-13.0.0_1.1.0	05/2023	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-12.1.0_1.1.0	12/2022	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-12.0.0_2.1.0	09/2022	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-12.0.0_1.1.0	06/2022	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-11.0.0_2.5.0	03/2022	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-11.0.0_2.3.0	12/2021	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0, C0) GA release
automotive-11.0.0_2.1.0	11/2021	Added the examples for i.MX 8QuadXPlus and upgraded the tool version
android-11.0.0_1.1.0-AUTO	01/2021	i.MX 8QuadXPlus/8QuadMax MEK GA release
android-10.0.0_2.4.0	07/2020	i.MX 8QuadMax MEK GA release
android-10.0.0_2.2.0-AUTO	06/2020	i.MX 8QuadXPlus/8QuadMax MEK GA release
automotive-10.0.0_1.1.0	03/2020	Deleted the Android 10 image
automotive-10.0.0_1.1.0	03/2020	i.MX 8QuadXPlus/8QuadMax MEK (Silicon Revision B0) GA release

Table 8. Revision history...continued

Document ID	Release date	Description
P9.0.0_2.1.0-AUTO-ga	08/2019	Updated the location of the SCFW porting kit
P9.0.0_2.1.0-AUTO-ga	04/2019	i.MX 8QuadXPlus/8QuadMax Automotive GA release
P9.0.0_1.0.2-AUTO-beta	01/2019	i.MX 8QuadXPlus/8QuadMax Automotive Beta release
P9.0.0_1.0.2-AUTO-alpha	11/2018	i.MX 8QuadXPlus/8QuadMax Automotive Alpha release
O8.1.0_1.1.0_AUTO-beta	05/2018	i.MX 8QuadXPlus/8QuadMax Beta release
O8.1.0_1.1.0_AUTO-EAR	02/2018	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.

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.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

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.

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.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamiQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Contents

1 Overview 2

2 Hardware Requirements 2

3 Working with the i.MX 8QuadXPlus/8QuadMax MEK Board 2

3.1 Board hardware 2

3.2 Board images 4

3.3 Flashing board images 6

4 Working with the i.MX 95 EVK Board 8

4.1 Board hardware 8

4.2 Board images 13

4.3 Flashing board images 15

5 Acronyms 17

6 Note About the Source Code in the Document 18

7 Revision History 18

Legal information 21

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