

# Android™ Quick Start Guide

## 1 Overview

This document guides you through the processes of downloading and running this release package. It only explains how to download and run the default release image with default configuration. For details on using the release package, see the *Android™ User's Guide (AUG)* included in this release package.

## 2 Hardware Requirements

The hardware requirements for using this release package are as follows:

Supported system-on-chips (SoCs):

- i.MX 8M Mini
- i.MX 8M Quad
- i.MX 8QuadMax
- i.MX 8QuadXPlus

Supported boards:

- EVK board and Platform
- MEK board and Platform

### Contents

1	Overview.....	1
2	Hardware Requirements.....	1
3	Working with the i.MX 8M Mini EVK Board.....	2
4	Working with the i.MX 8M Quad EVK Board.....	9
5	Working with the i.MX 8QuadMax MEK Board.....	18
6	Working with the i.MX 8QuadXPlus MEK Board.....	25
7	Revision History.....	30



## 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 board.

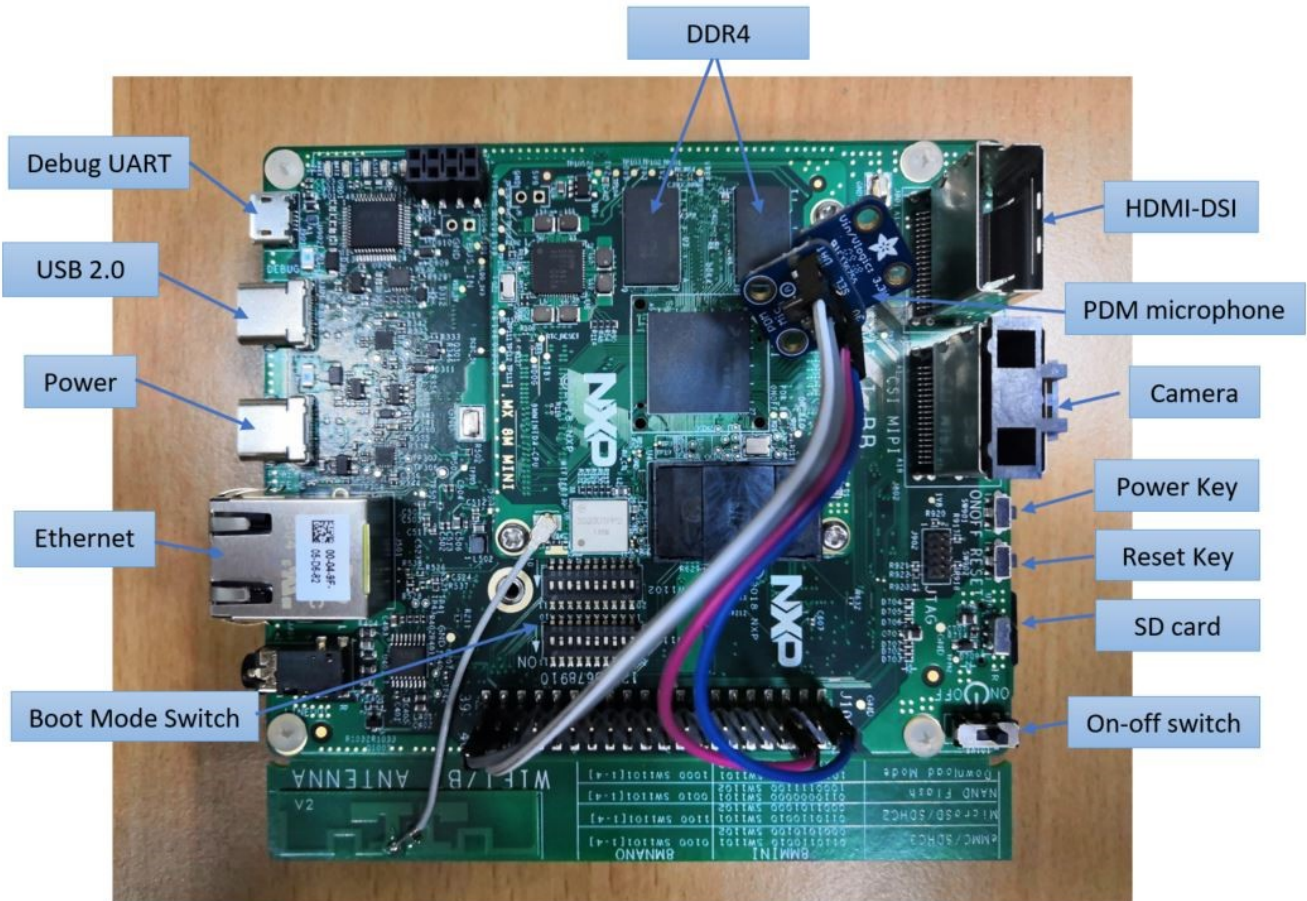


Figure 1. i.MX 8M Mini EVK board

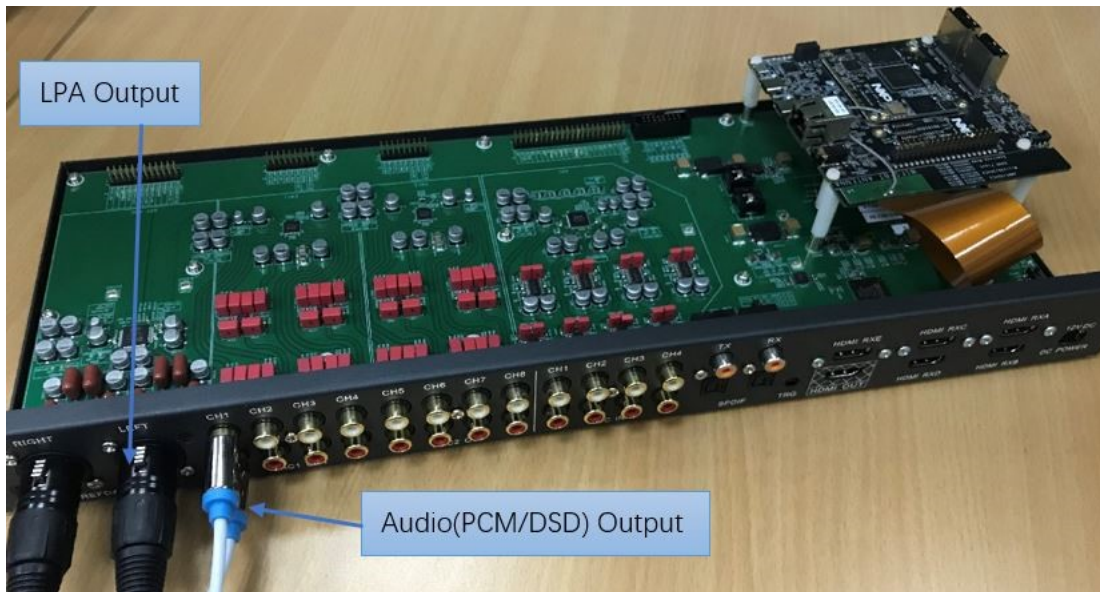


Figure 2. i.MX 8M Mini EVK with audio board



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



Figure 4. i.MX MIPI panel



Figure 5. i.MX MIPI camera

**NOTE**

- 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 MIPI panel display, connect the i.MX MIPI panel to the "HDMI DSI" port.
- To test the camera, connect the i.MX CSI MIPI Camera to the "Camera" port.
- The QCOM 1PJ Wi-Fi/Bluetooth module is used on the i.MX 8M Mini EVK LPDDR4 Board.
- The BCM 1MW Wi-Fi/Bluetooth module is used on the i.MX 8M Mini EVK DDR4 Board.

## 3.2 Board images

The table below describes the location in the board partitions of the software images in android\_p9.0.0\_2.0.0-ga\_image\_8mmevk.tar.gz.

**Table 1. Board images**

Image name	Download target
/u-boot-imx8mm.img	33 KB offset of MMC for a board with LPDDR4 on it.
/u-boot-imx8mm-ddr4.img	33 KB offset of MMC for a board with DDR4 on it.
/u-boot-imx8mm-evk-uuu.img	Bootloader used by UUU for the i.MX 8M Mini board with LPDDR4 on it. It is not flashed to MMC.
/u-boot-imx8mm-ddr4-evk-uuu.img	Bootloader used by UUU for the i.MX 8M Mini board with DDR4 on it. It is not flashed to MMC.
/imx8mm_m4_demo.img	5120 KB offset of MMC.
/partition-table.img	0 offset of MMC. If the actually size of the SD card is larger than 13 GB, use the default partition-table.img.
/partition-table-7GB.img	0 offset of MMC. If the actually size of the SD card is larger than 7 GB, use this image as partition-table.img.
/partition-table-28GB.img	0 offset of MMC. If the actually size of the SD card is larger than 28 GB, use this image as partition-table.img.
/boot.img	boot_a and boot_b partitions.
/vbmeta-imx8mm.img	vbmeta_a and vbmeta_b partitions to support LPDDR4 and MIPI-to-HDMI output and Direct Stream Digital (DSD) playback.
/vbmeta-imx8mm-ddr4.img	vbmeta_a and vbmeta_b partitions to support DDR4 and MIPI-to-HDMI output.
/vbmeta-imx8mm-m4.img	vbmeta_a and vbmeta_b partitions to support LPDDR4, MIPI-to-HDMI output, and audio playback based on Cortex-M4 FreeRTOS.
/vbmeta-imx8mm-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support LPDDR4 and MIPI panel output.
/system.img	system_a and system_b partitions.
/vendor.img	vendor_a and vendor_b partitions.
/dtbo-imx8mm.img	dtbo_a and dtbo_b partitions to support LPDDR4, MIPI-to-HDMI output, and DSD playback.
/dtbo-imx8mm-ddr4.img	dtbo_a and dtbo_b partitions to support DDR4 and MIPI-to-HDMI output.
/dtbo-imx8mm-m4.img	dtbo_a and dtbo_b partitions to support LPDDR4, MIPI-to-HDMI output, and audio playback based Cortex-M4 FreeRTOS.
/dtbo-imx8mm-mipi-panel.img	dtbo_a and dtbo_b partitions to support LPDDR4 and MIPI panel output.

## 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](#).

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

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 1.2.91 version. Download corresponding version from github:

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

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

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
  - For Rev. B boards, change the first two bits of board's sw1101 to 10 (from 1-2 bit) to enter serial download mode.
  - For Rev. C boards, change the first four bits of board's sw1101 to 1010 (from 1-4 bit) to enter serial download mode.
3. Power on the board. Use the USB cable on the board OTG port to connect your PC with the board.
4. Decompress `release_package/android_p9.0.0_2.0.0-ga_image_8mmevk.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 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 the SoC information. For i.MX 8M Mini, it should be "imx8mm". This option is mandatory.
-a	Only flashes slot a. If this option and "-b" option are not used, slots a and b are both flashed.
-b	Only flashes slot b. If this option and "-a" option are not used, slots a and b are both flashed.
-c card_size	Specifies which partition table image file to flash. For i.MX 8M Mini, it can be followed with "7" or "28". If this option is not used, default "partition-table.img" is flashed.
-m	Flashes Cortex-M4 image. If this option is not used, Cortex-M4 image is not flashed.
-d dev	Specifies some images with "dev" in its name. For i.MX 8M Mini, it can be "m4", "mipi-panel", "ddr4". If this option is not used, default dtbo and vbmeta images are flashed.
-e	Erases user data after images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. If this option is not used, images in the current working directory are flashed.

*Table continues on the next page...*

**Table 2. Options for uuu\_imx\_android\_flash tool (continued)**

Option	Description
-t target_dev	Specifies the target device. For i.MX 8M Mini, it can be "emmc" and "sd". If this option is not used, images are flashed to eMMC.
-daemon	Run UUU in Daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images will be flashed. The script loads U-Boot to RAM and executes to fastboot mode. this option is used for development.

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

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

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

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

```
> .\uuu_imx_android_flash.bat -f imx8mm -a -e
```

When the command above is executed, the default images will be flashed into eMMC slot a for i.MX 8M Mini.

#### NOTE

- 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".
- If your SD card is 8 GB, execute the tool with "-c 7".
- To test MIPI-DSI to HDMI output with LPDDR4 on board, it does not need to use -d option.
- To test MIPI-DSI to HDMI output with DDR4 on board, execute the tool with "-d ddr4".
- To test MIPI panel output with LPDDR4 on board, execute the tool with "-d mipi-panel".
- To test MIPI-to-HDMI output and audio playback based on Cortex-M4 FreeRTOS with LPDDR4 on board, execute the tool with "-m" and "-d m4".
- uuu\_imx\_android\_flash.bat generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- If uuu\_imx\_android\_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

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

6. Wait for the uuu\_imx\_android\_flash execution to complete. If there is not any error, you will get information on the command window indicating that images are already flashed.
7. Power off the board.

### 8. Change boot device as eMMC or SD card.

For Rev. B boards:

- Change sw1101 to 01110010 and change sw1102 to 00101010 if you want to boot from eMMC.
- Change sw1101 to 01000110 and change sw1102 to 00110100 if you want to boot from SD card.

For Rev. C boards:

- Change sw1101 to 0110110010 and change sw1102 to 0001010100 if you want to boot from eMMC.
- Change sw1101 to 0110110010 and change sw1102 to 0001101000 if you want to boot from SD card.

## 3.4 Booting

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

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

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

- i.MX 8M Mini EVK LPDDR4 Board:

```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMXC1 consoleblank=0 androidboot.hardware=freescale
cma=800M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wifivendor=qca androidboot.wificountrycode=CN
U-Boot > saveenv
```

- i.MX 8M Mini EVK DDR4 Board:

```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMXC1 consoleblank=0 androidboot.hardware=freescale
cma=800M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wifivendor=bcm androidboot.wificountrycode=CN
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

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

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

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

- i.MX 8M Mini EVK LPDDR4 Board:

```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMXC1 consoleblank=0 androidboot.hardware=freescale
cma=800M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wifivendor=qca androidboot.wificountrycode=CN
U-Boot > setenv bootcmd "bootmcu && boota mmc0" # for SD boot
U-Boot > setenv bootcmd "bootmcu && boota mmc1" # for eMMC boot
U-Boot > saveenv
```

- i.MX 8M Mini EVK DDR4 Board:

```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200
init=/init androidboot.console=ttyMXC1 consoleblank=0 androidboot.hardware=freescale
cma=800M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wifivendor=bcm androidboot.wificountrycode=CN
```



```

U-Boot > setenv bootcmd "bootmcu && boota mmc0" # for SD boot
U-Boot > setenv bootcmd "bootmcu && boota mmc1" # for emmc boot
U-Boot > saveenv

```

**NOTE**

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

```

U-Boot > setenv bootcmd "boota mmc0" # for SD boot
U-Boot > setenv bootcmd "boota mmc1" # for emmc boot
U-Boot > saveenv

```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

```

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

```

### 3.5 Board reboot

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

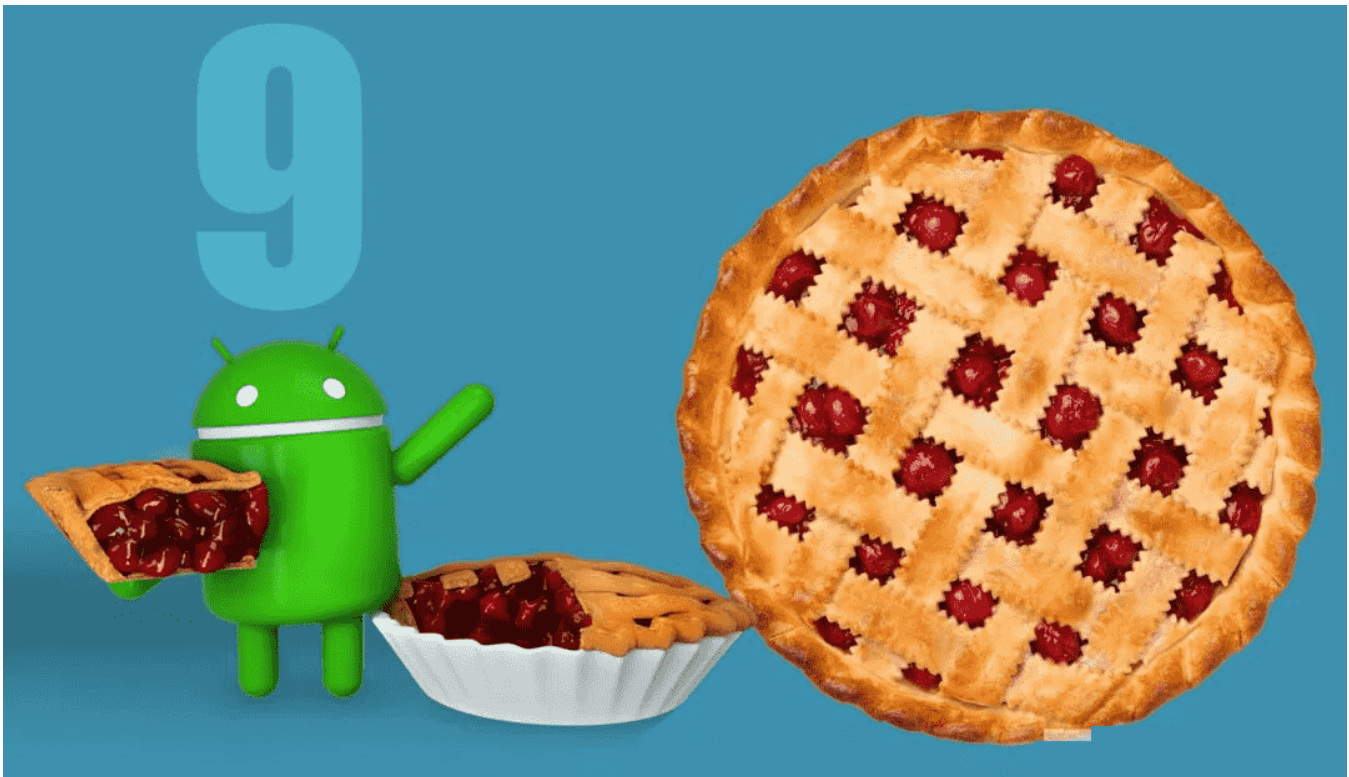


Figure 6. Android Pie image

## 4 Working with the i.MX 8M Quad EVK Board

## 4.1 Board hardware

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

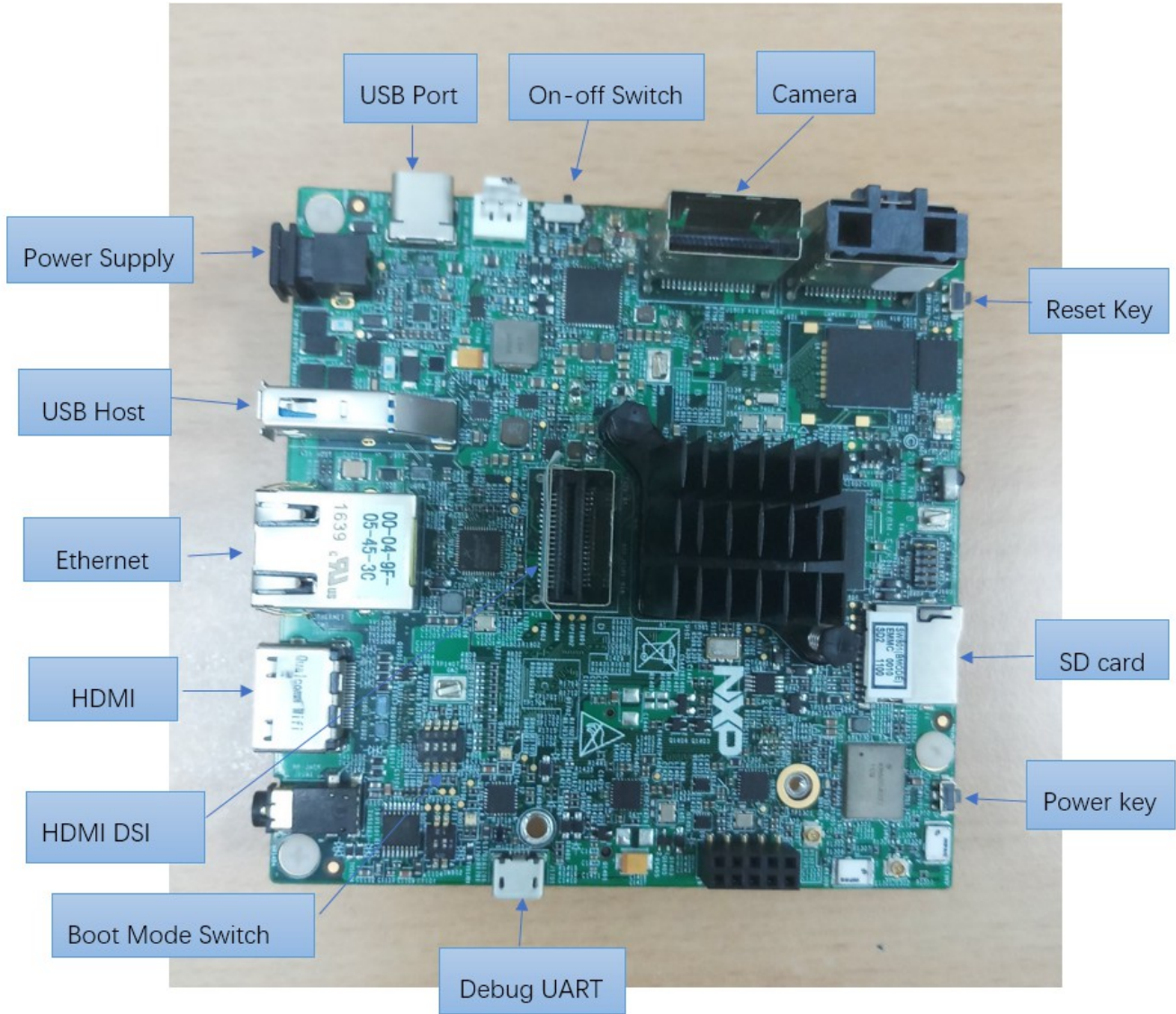


Figure 7. i.MX 8M Quad EVK board

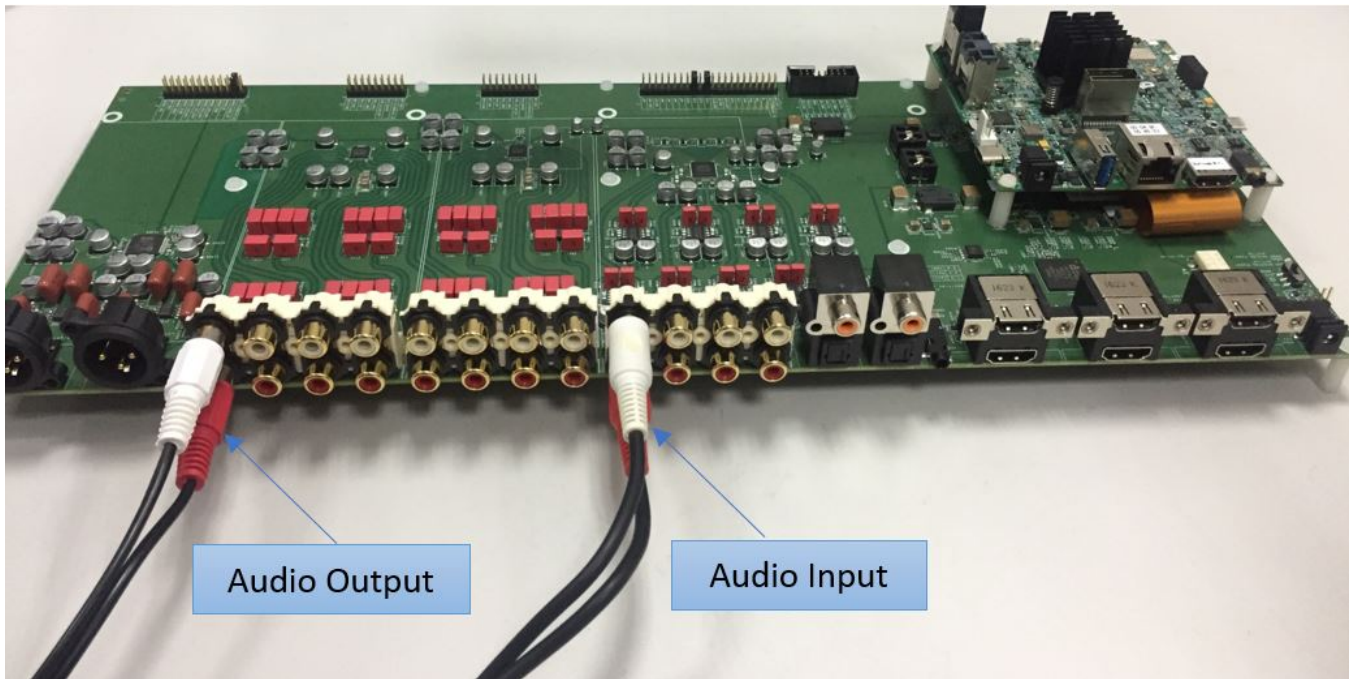


Figure 8. i.MX 8M Quad EVK with audio board



Figure 9. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 10. i.MX MIPI panel



Figure 11. i.MX MIPI camera

**NOTE**

- To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "HDMI DSI" port.
- To test the MIPI panel display, connect the i.MX MIPI panel to the "HDMI DSI" port.
- To test the camera, connect the i.MX CSI MIPI Camera to the "Camera" port.
- The QCOM 1CQ Wi-Fi/Bluetooth module is used on the i.MX 8MQuad Rev. B3/B4 Board.
- The BCM 1CX Wi-Fi/Bluetooth module is used on the i.MX 8MQuad Rev. A Board.

## 4.2 Board images

The table below describes the location in the board partitions of the software images in android\_p9.0.0\_2.0.0-ga\_image\_8mqevk.tar.gz.

**Table 3. Board images**

Image name	Download target
/u-boot-imx8mq.imgx	33 KB offset of MMC
/u-boot-imx8mq-evk-uuu.imgx	Bootloader used by UUU for the i.MX 8MQuad board. It is not flashed to MMC
/partition-table.img	0 offset of MMC. If the actually size of the SD card is larger than 13 GB, use the default partition-table.img
/partition-table-7GB.img	0 offset of MMC. If the actually size of the SD card is larger than 7 GB, use this image as partition-table.img
/partition-table-28GB.img	0 offset of MMC. If the actually size of the SD card is larger than 28 GB, use this image as partition-table.img
/boot.img	boot_a and boot_b partitions
/vbmeta-imx8mq.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad B4 board HDMI output and DSD playback
/vbmeta-imx8mq-mipi.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad B4 board MIPI-to-HDMI output
/vbmeta-imx8mq-dual.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad B4 board HDMI and MIPI-to-HDMI dual output
/vbmeta-imx8mq-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad B4 board MIPI panel output
/vbmeta-imx8mq-b3.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad B3 board HDMI output and DSD playback
/vbmeta-imx8mq-mipi-b3.img	vbmeta_a and vbmeta_b partitions to support i.MX 8M Quad B3 board MIPI-to-HDMI output
/vbmeta-imx8mq-mipi-panel-b3.img	vbmeta_a and vbmeta_b partitions to support i.MX 8MQuad B3 board MIPI panel output
/system.img	system_a and system_b partitions
/vendor.img	vendor_a and vendor_b partitions
/dtbo-imx8mq.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad B4 board HDMI output
/dtbo-imx8mq-mipi.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad B4 board MIPI-to-HDMI output
/dtbo-imx8mq-dual.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad B4 board HDMI and MIPI-to-HDMI dual output
/dtbo-imx8mq-mipi-panel.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad B4 board MIPI panel output
/dtbo-imx8mq-b3.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad B3 board HDMI output
/dtbo-imx8mq-mipi-b3.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad B3 board MIPI-to-HDMI output
/dtbo-imx8mq-mipi-panel-b3.img	dtbo_a and dtbo_b partitions to support i.MX 8M Quad B3 board MIPI panel output

## 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 1.2.91 version. Download corresponding version from github:

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

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

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.

Change the board's sw802 (boot mode) to 01 (from 1-2 bit) to enter serial download mode.

3. Power on the board. Use the USB cable on the board USB 3.0 port to connect your PC with the board.

### NOTE

- There are two USB ports on the i.MX 8M Quad EVK board: USB-to-UART, USB 3.0. The USB-to-UART port can be referenced as debug UART, and the USB 3.0 port can be referenced as USB 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_p9.0.0_2.0.0-ga_image_8mqevk.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
  5. Execute the `uuu_imx_android_flash` tool to flash images.  
The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For i.MX 8M Quad board, related options are described as follows

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

Option	Description
-h	Displays the help information of this tool.
-f soc_name	Specifies the SoC information. For 8M Quad, it should be "imx8mq". This option is mandatory.
-a	Only flashes slot a. If this option and "-b" option are not used, slots a and b are both flashed.
-b	Only flashes slot b. If this option and "-a" option are not used, slots a and b are both flashed.

*Table continues on the next page...*

**Table 4. Options for uuu\_imx\_android\_flash tool (continued)**

Option	Description
-c card_size	Specifies which partition table image file to flash. For 8M Quad, it can be followed with "7" or "28". If this option is not used, default "partition-table.img" is flashed.
-d dev	Specifies some images with "dev" in its name. For 8M Quad, it can be "b3", "dual", "mipi-b3", "mipi-panel-b3", "mipi-panel", "mipi". If this option is not used, default dtbo and vbmeta images are flashed.
-e	Erases user data after images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. If this option is not used, images in the current working directory are flashed.
-t target_dev	Specifies the target device. For 8M Quad, it can be "emmc" and "sd". If this option is not used, images are flashed to eMMC.
-daemon	Run UUU in Daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images will be flashed. The script loads U-Boot to RAM and executes to fastboot mode. this option is used for development.

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

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

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

```
> .\uuu_imx_android_flash.bat -f imx8mq -a -e
```

When the command above is executed, the default images will be flashed into eMMC slot a for 8M Quad.

#### NOTE

- 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".
- If your SD card is 8 GB, execute the tool with "-c 7".
- To test feature on i.MX 8MQuad B4 and Rev. A board:
  - To test HDMI output, it does not need to use -d option.
  - To test MIPI-to-HDMI output, execute the tool with "-d mipi".
  - To test MIPI panel output, execute the tool with "-d mipi-panel".
  - To test HDMI and MIPI-to-HDMI dual output, execute the tool with "-d dual".
- To test feature on i.MX 8M Quad B3 board:
  - To test HDMI output, execute the tool with "-d b3".
  - To test MIPI-to-HDMI output, execute the tool with "-d mipi-b3".
  - To test MIPI panel output, execute the tool with "-d mipi-panel-b3".

- `uuu_imx_android_flash.bat` generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- If `uuu_imx_android_flash.bat` is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

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

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, you will get information on the command window indicating that images are already flashed.
7. Power off the board.
8. Change boot device as eMMC or SD card. Change the board's `sw802` (boot mode) to 10 (from 1-2 bit) to leave serial download mode.
  - Change `SW801` to switch the board back to 1100 (SD boot mode).
  - Change `SW801` to switch the board back to 0010 (eMMC boot mode).

## 4.4 Booting

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

### 4.4.1 Booting with single display: HDMI display

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

- i.MX 8M Quad EVK B3/B4 Board:

```
U-Boot > setenv bootargs console=ttyMxc0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.gui_resolution=1080p androidboot.console=ttyMxc0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.fbTileSupport=enable androidboot.wifivendor=qca
androidboot.wificountrycode=CN
U-Boot > saveenv
```

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

```
U-Boot > setenv bootargs console=ttyMxc0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.gui_resolution=1080p androidboot.console=ttyMxc0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.fbTileSupport=enable androidboot.wifivendor=bcm
androidboot.wificountrycode=CN
U-Boot > saveenv
```

With above settings, the Android platform does not start the shell console. To disable selinux, append "`androidboot.selinux=permissive`" to the U-Boot's `bootargs`. Boot environment variables are as follows:

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

### 4.4.2 Booting with single display: MIPI-to-HDMI display

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



- i.MX 8M Quad EVK B3/B4 Board:

```
U-Boot > setenv bootargs console=ttyMXC0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.lcd_density=160 androidboot.console=ttyMXC0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=mxsfb-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wifivendor=qca androidboot.displaymode=720p androidboot.wificountrycode=CN
U-Boot > saveenv
```

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

```
U-Boot > setenv bootargs console=ttyMXC0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.lcd_density=160 androidboot.console=ttyMXC0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=mxsfb-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wifivendor=bcm androidboot.displaymode=720p androidboot.wificountrycode=CN
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

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

### 4.4.3 Booting with dual displays: HDMI and MIPI-to-HDMI displays

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

- i.MX 8M Quad EVK B4 Board:

```
U-Boot > setenv bootargs console=ttyMXC0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.gui_resolution=1080p androidboot.console=ttyMXC0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wifivendor=qca androidboot.wificountrycode=CN
U-Boot > saveenv
```

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

```
U-Boot > setenv bootargs console=ttyMXC0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.gui_resolution=1080p androidboot.console=ttyMXC0 consoleblank=0
androidboot.hardware=freescale cma=1280M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wifivendor=bcm androidboot.wificountrycode=CN
U-Boot > saveenv
```

With above settings, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

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

### 4.4.4 Booting with single display: MIPI panel

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

- i.MX 8M Quad EVK B3/B4 Board:

```
U-Boot > setenv bootargs console=ttyMXC0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.console=ttyMXC0 consoleblank=0 androidboot.hardware=freescale
cma=1280M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wifivendor=qca androidboot.wificountrycode=CN
U-Boot > saveenv
```

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

## Working with the i.MX 8QuadMax MEK Board

```
U-Boot > setenv bootargs console=ttyMXC0,115200 earlycon=imxuart,0x30860000,115200
init=/init androidboot.console=ttyMXC0 consoleblank=0 androidboot.hardware=freescale
cma=1280M androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wifi vendor=bcm androidboot.wificountrycode=CN
androidboot.wificountrycode=CN
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

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

## 4.5 Board reboot

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

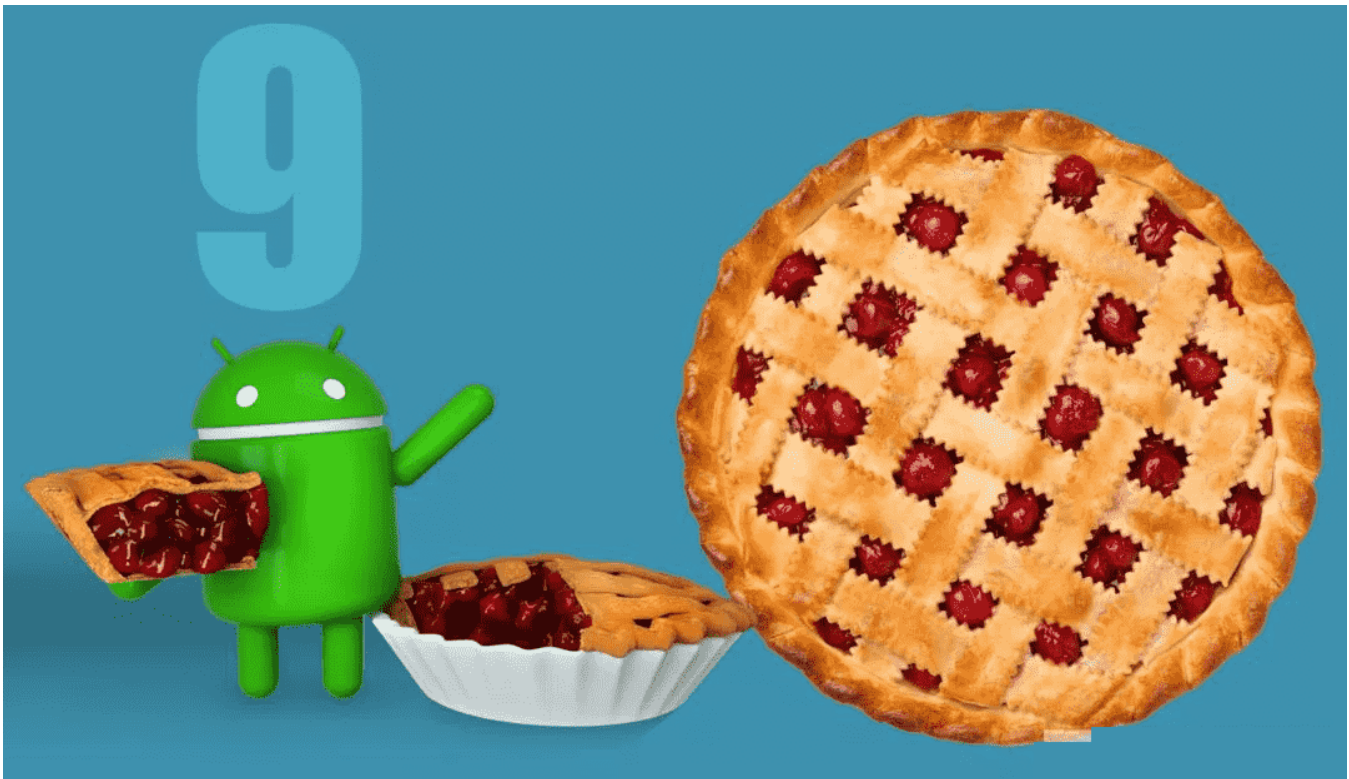


Figure 12. Android Pie image

## 5 Working with the i.MX 8QuadMax MEK Board

### 5.1 Board hardware

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

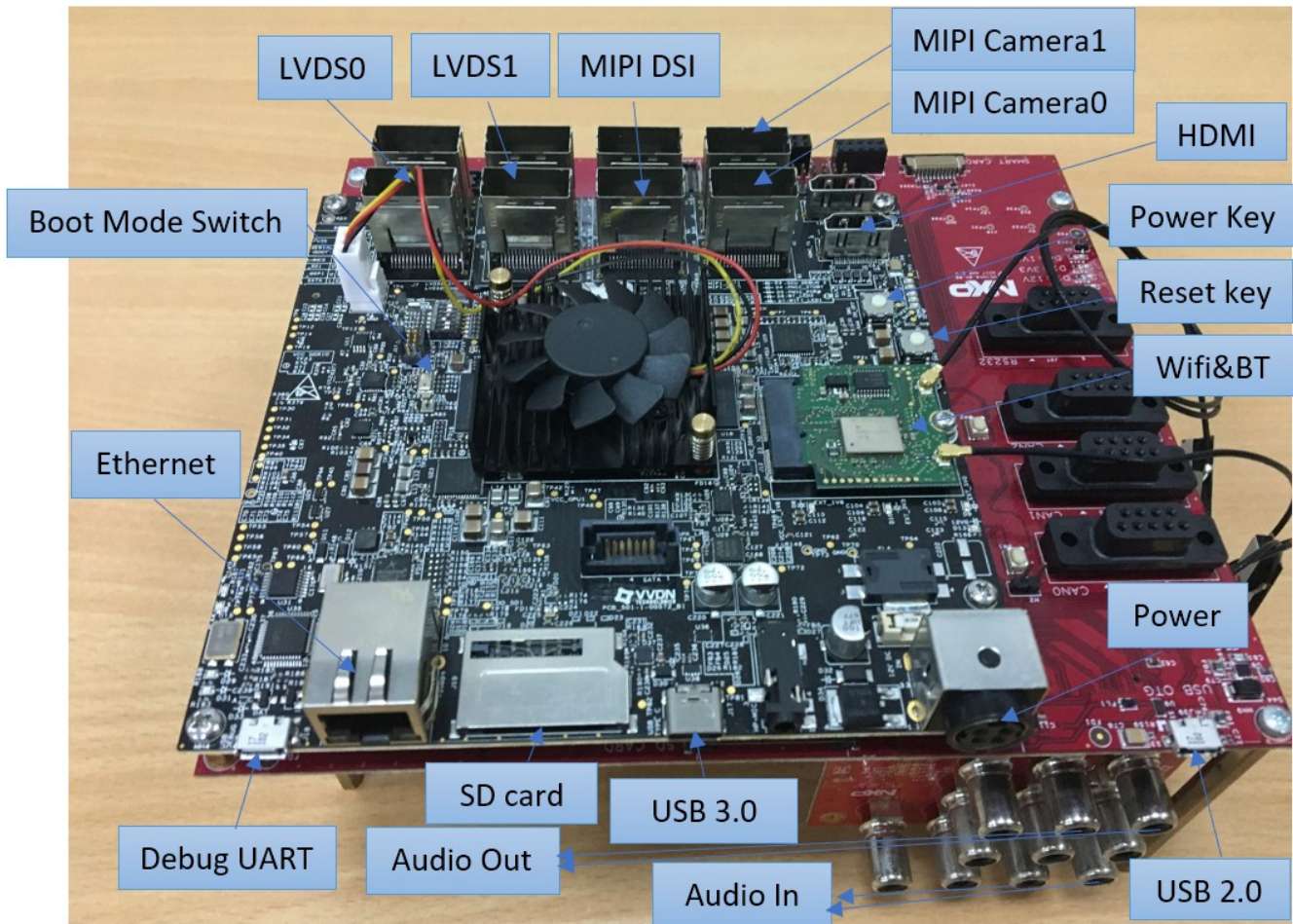


Figure 13. i.MX 8QuadMax MEK board



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

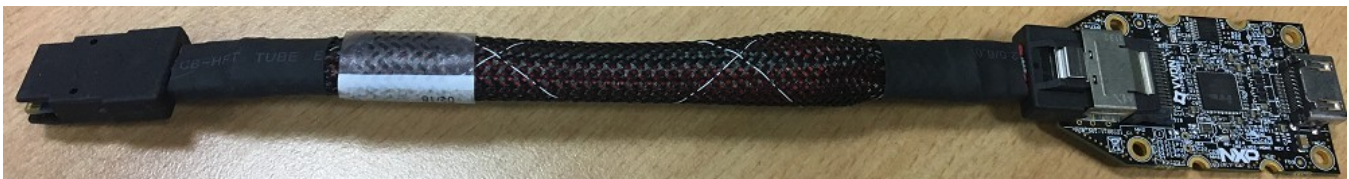


Figure 15. i.MX mini SAS cable with LVDS-to-HDMI adapter



Figure 16. i.MX MIPI panel



Figure 17. i.MX MIPI camera

**NOTE**

- To test the MIPI-DSI to HDMI display, use the i.MX mini SAS cable to connect the DSI-to-HDMI adapter to the "HDMI DSI" port.
- To test the LVDS-to-HDMI display, use the i.MX mini SAS cable to connect the LVDS-to-HDMI adapter to the "LVDS0/LVDS1" port.
- To test the camera, connect two i.MX MIPI cameras to the "MIPI Camera0" and "MIPI Camera1" ports.
- To test the MIPI panel display, connect the i.MX MIPI panel to the "MIPI DSI" port.
- To test single camera, connect i.MX MIPI Camera to the "MIPI Camera0" or "MIPI Camera1" port.
- To test dual cameras, connect both.

## 5.2 Board images

The table below describes the location in the board partitions of the software images in android\_p9.0.0\_2.0.0-ga\_image\_8qmek.tar.gz.

**Table 5. Board images**

Image name	Download target
/u-boot-imx8qm.img	0 KB offset of eMMC and 32 KB offset of SD card.
u-boot-imx8qm-mek-uuu.img	Bootloader used by UUU for i.MX 8QuadMax MEK board. It is not flashed to MMC.
/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-7GB.img	Program to the first 17 KB, and then back up to last 17 KB of the boot storage. GPT table image for 8 GB boot storage.
/partition-table-28GB.img	Program to first 17 KB, and then back up to last 17 KB of the boot storage. GPT table image for 32 GB boot storage.
/boot.img	boot_a and boot_b partitions
/vbmeta-imx8qm.img	vbmeta_a and vbmeta_b partitions to support LVDS-to-HDMI/MIPI-to-HDMI display.
/vbmeta-imx8qm-hdmi.img	vbmeta_a and vbmeta_b partitions to support physical HDMI display.
/vbmeta-imx8qm-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support MIPI panel display.
/system.img	system_a and system_b partitions.
/vendor.img	vendor_a and vendor_b partitions.
/dtbo-imx8qm.img	dtbo_a and dtbo_b partitions to support LVDS-to-HDMI/MIPI-to-HDMI display.
/dtbo-imx8qm-hdmi.img	dtbo_a and dtbo_b partitions to support physical HDMI display.
dtbo-imx8qm-mipi-panel.img	dtbo_a and dtbo_b partitions to support MIPI panel display.

## 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 1.2.91 version. Download corresponding version from github:

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

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

## Working with the i.MX 8QuadMax MEK Board

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\_p9.0.0\_2.0.0-ga\_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 6. Options for uuu\_imx\_android\_flash tool**

Option	Description
-h	Displays the help information of this tool.
-f soc_name	Specifies the SoC information. For 8QuadMax, it should be "imx8qm". This option is mandatory.
-a	Only flashes slot a. If this option and "-b" option are not used, slots a and b are both flashed.
-b	Only flashes slot b. If this option and "-a" option are not used, slots a and b are both flashed.
-c card_size	Specifies which partition table image file to flash. For 8QuadMax, it can be followed with "7" or "28". If this option is not used, default "partition-table.img" is flashed.
-d dev	Specifies some images with "dev" in its name. For 8QuadMax, it can be "hdmi", "mipi-panel". If this option is not used, default dtbo and vbmeta images are flashed.
-e	Erases user data after images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. If this option is not used, images in the current working directory are flashed.
-t target_dev	Specifies the target device. For 8QuadMax, it can be "emmc" and "sd". If this option is not used, images are flashed to eMMC.
-daemon	Run UUU in Daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images will be flashed. The script loads U-Boot to RAM and executes to fastboot mode. this option is used for development.

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

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

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

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

When the command above is executed, the default images will be flashed into eMMC slot a for 8QuadMax.

#### NOTE

- 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".
- If your SD card is 8 GB, execute the tool with "-c 7".
- To test LVDS-to-HDMI/MIPI-to-HDMI display, it does not need to use -d option.
- To test MIPI panel output, execute the tool with "-d mipi-panel".
- To test physical HDMI display, execute the tool with "-d hdmi".
- uuu\_imx\_android\_flash.bat generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- If uuu\_imx\_android\_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

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

6. Wait for the uuu\_imx\_android\_flash execution to complete. If there is not any error, you will get information on the command window indicating that images are already flashed.
7. Power off the board.
8. Change boot device as eMMC or SD card.
  - Change SW2 to switch the board back to 000100 (from 1-6 bit) to enter eMMC boot mode.
  - Change SW2 to switch the board back to 001100 (from 1-6 bit) to enter SD boot mode.

## 5.4 Booting

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

### 5.4.1 Booting with LVDS-to-HDMI/MIPI-to-HDMI display

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon=lpuart32,0x5a060000,115200 init=/
init androidboot.console=ttyLP0 consoleblank=0 androidboot.hardware=freescale
androidboot.fbTileSupport=enable cma=800M@0x960M-0xe00M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN
U-Boot > saveenv
```

## Working with the i.MX 8QuadMax MEK Board

With above settings, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

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

### 5.4.2 Booting with physical HDMI display

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon=lpuart32,0x5a060000,115200 init=/
init androidboot.console=ttyLP0 consoleblank=0 androidboot.hardware=freescale
androidboot.fbTileSupport=enable cma=1184M@0x960M-0xe00M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN
U-Boot > saveenv
```

With the settings above, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

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

## 5.5 Board reboot

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

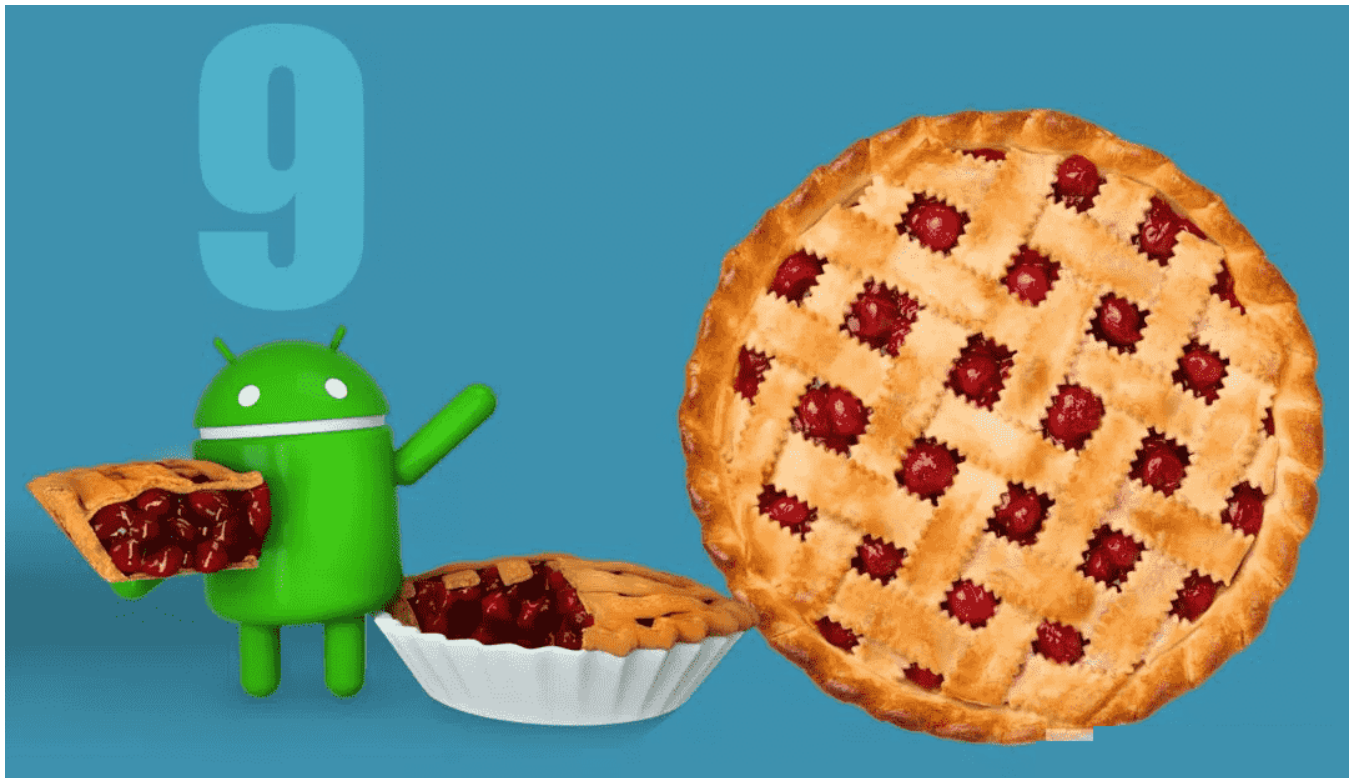


Figure 18. Android Pie image



## 6 Working with the i.MX 8QuadXPlus MEK Board

### 6.1 Board hardware

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

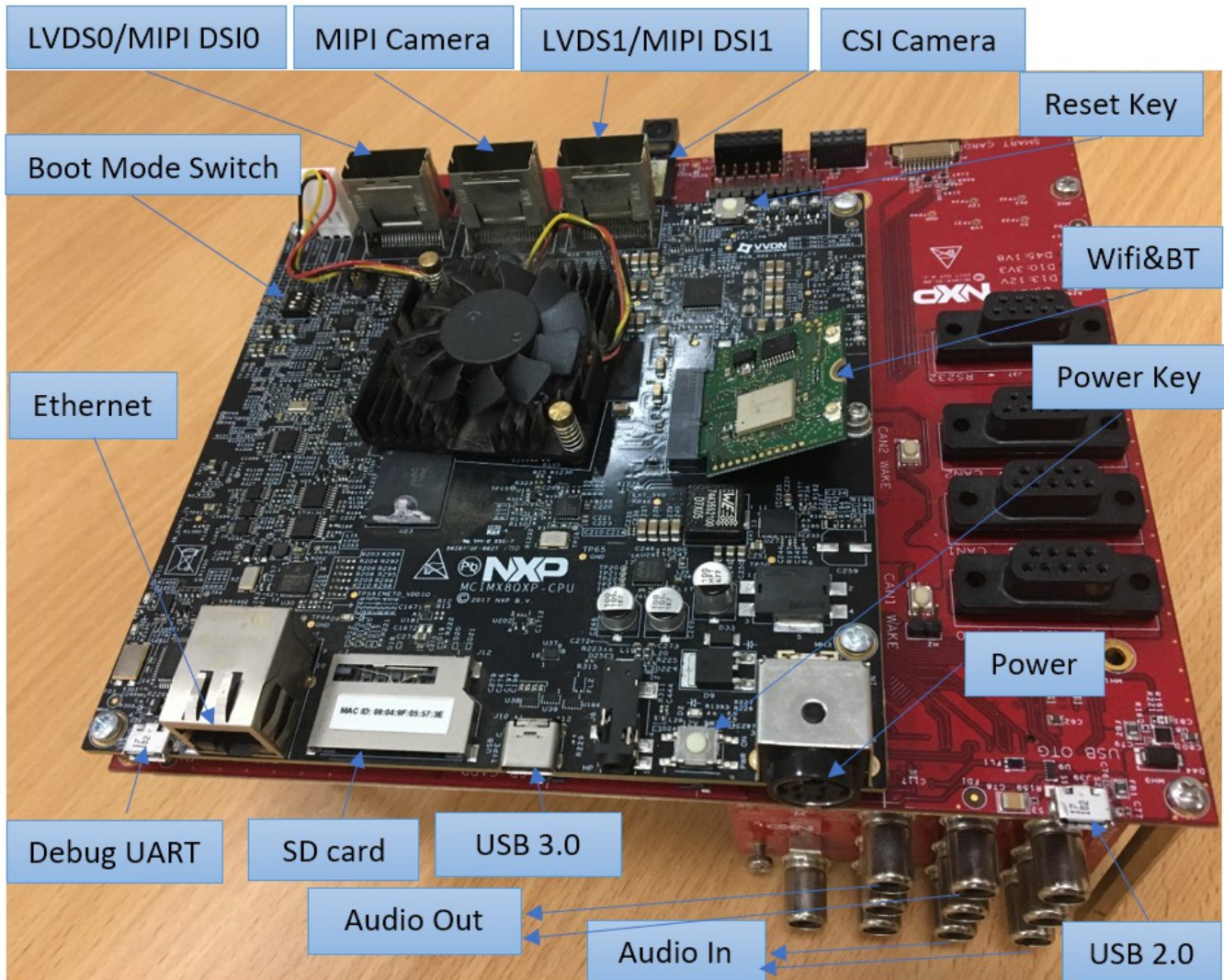


Figure 19. i.MX 8QuadXPlus MEK board



Figure 20. i.MX mini SAS cable with DSI-to-HDMI adapter



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



Figure 22. i.MX MIPI camera

**NOTE**

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

## 6.2 Board images

The table below describes the location in the board partitions of the software images in android\_p9.0.0\_2.0.0-ga\_image\_8qmek.tar.gz.

**Table 7. Board images**

Image name	Download target
/u-boot-imx8qxp.imx	32 KB offset of MMC.
/u-boot-imx8qxp-mek-uuu.imx	Bootloader used by UUU for the i.MX 8QuadMax MEK board. It is not flashed to MMC.
/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-7GB.img	Program to the first 17 KB, and then back up to last 17 KB of the boot storage. GPT table image for 8 GB boot storage.
/partition-table-28GB.img	Program to first 17 KB, and then back up to last 17 KB of the boot storage. GPT table image for 32 GB boot storage.
/boot.img	boot_a and boot_b partitions
/vbmeta-imx8qxp.img	vbmeta_a and vbmeta_b partitions to support single LVDS-to-HDMI/MIPI-to-HDMI or dual LVDS-to-HDMI display with dual-camera support.
/system.img	system_a and system_b partitions
/vendor.img	vendor_a and vendor_b partitions

Table continues on the next page...

**Table 7. Board images (continued)**

/dtbo-imx8qxp.img	dtbo_a and dtbo_b partitions to support single LVDS-to-HDMI/MIPI-to-HDMI or dual LVDS-to-HDMI displays with dual-camera support.
-------------------	--

## 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 1.2.91 version. Download corresponding version from github:

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

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

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.

Change the board's SW2 (boot mode) to 1000 (from 1-4 bit) to enter serial download mode.

3. Power on the board. Use the USB cable on the board USB 3.0 Type-C port to connect your PC with the board.

### NOTE

- There are three USB ports on the 8QuadXPlus MEK board: USB-to-UART, USB 2.0, and USB 3.0.
- The USB-to-UART port can be referenced as debug UART, which can be used to watch the log of the hardware boot processing.
- USB 2.0 is USB Host and USB 3.0 is USB OTG.

4. Decompress release\_package/android\_p9.0.0\_2.0.0-ga\_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 8. Options for uuu\_imx\_android\_flash tool**

Option	Description
-h	Displays the help information of this tool.
-f soc_name	Specifies the SoC information. For 8QuadXPlus, it should be "imx8qxp". This option is mandatory.

Table continues on the next page...

**Table 8. Options for uuu\_imx\_android\_flash tool (continued)**

Option	Description
-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 8QuadXPlus, it can be followed with "7" or "28". If this option is not used, default "partition-table.img" is flashed.
-e	Erases user data after images are flashed.
-D directory	Specifies the directory in which there are the images to be flashed. If this option is not used, images in the current working directory are flashed.
-t target_dev	Specifies the target device. For 8QuadXPlus, it can be "emmc" and "sd". If this option is not used, images are flashed to eMMC.
-daemon	Run UUU in Daemon mode. This option is used to flash multiple boards of the same type.
-i	If the script is executed with this option, no images will be flashed. The script loads U-Boot to RAM and executes to fastboot mode. this option is used for development.

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

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

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

```
> .\uuu_imx_android_flash.bat -f imx8qxp -a -e
```

When the command above is executed, the default images will be flashed into eMMC slot a for 8QuadXPlus.

#### NOTE

- 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".
- If your SD card is 8 GB, execute the tool with "-c 7".
- uuu\_imx\_android\_flash.bat generates a temporary file under the current working directory. Make sure you have Write permission under the current working directory.
- If uuu\_imx\_android\_flash.bat is used to flash images on a remote server through samba, you need to map the remote resource to the local environment first. Take the following command as an example:

```
> net use z: \\10.193.108.179\daily_images
```

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

6. Wait for the `uuu_imx_android_flash` execution to complete. If there is not any error, you will get information on the command window indicating that images are already flashed.
7. Power off the board.
8. Change boot device as eMMC or SD card.
  - Change SW2 to switch the board back to 0100 (from 1-4 bit) to enter eMMC boot mode.
  - Change SW2 to switch the board back to 1100 (from 1-4 bit) to enter SD boot mode..

## 6.4 Booting

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

### 6.4.1 Booting with single LVDS-to-HDMI/MIPI-to-HDMI or dual LVDS-to-HDMI displays

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon=lpuart32,0x5a060000,115200 init=/
init androidboot.console=ttyLP0 consoleblank=0 androidboot.hardware=freescale
androidboot.fbTileSupport=enable cma=800M@0x960M-0xe00M androidboot.primary_display=imx-drm
firmware_class.path=/vendor/firmware transparent_hugepage=never
androidboot.wificountrycode=CN
U-Boot > saveenv
```

With above settings, the Android platform does not start the shell console. To disable selinux, append "androidboot.selinux=permissive" to the U-Boot's bootargs. Boot environment variables are as follows:

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

## 6.5 Board reboot

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

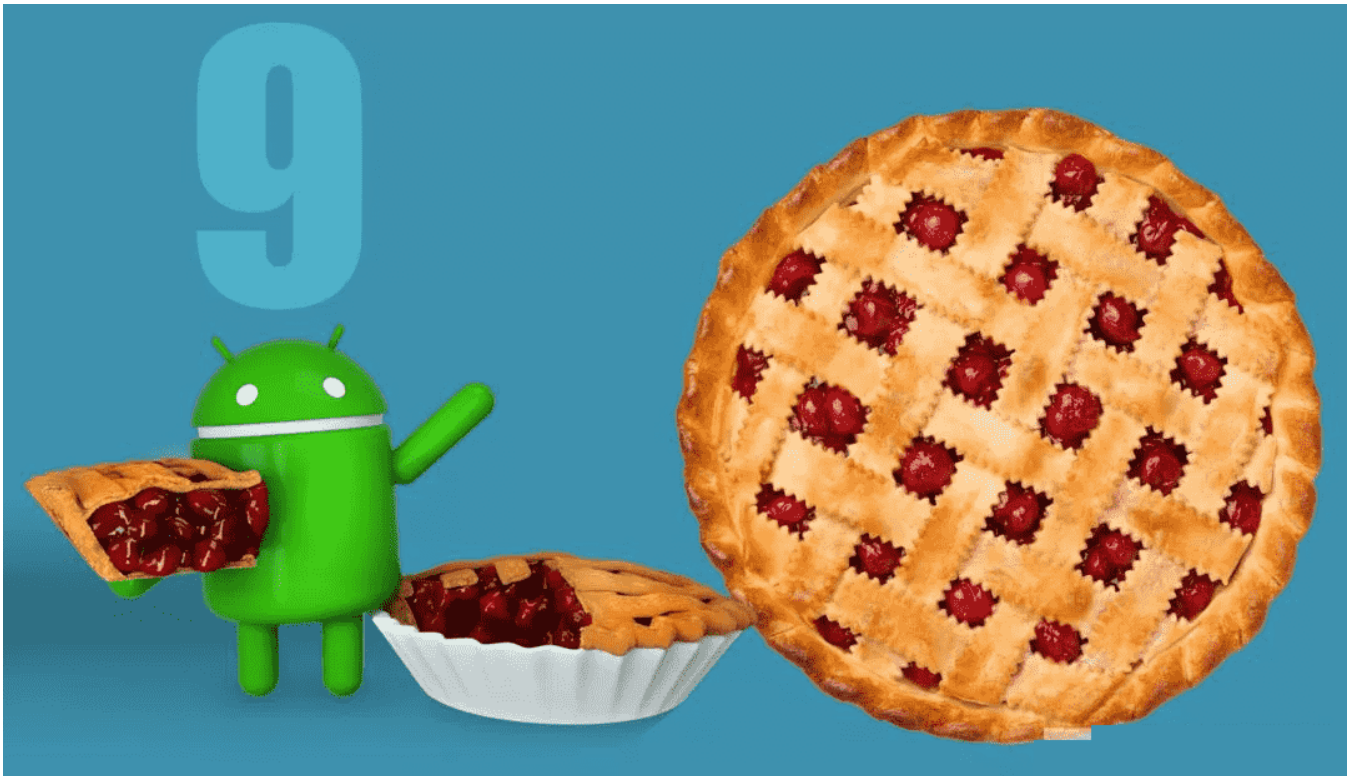


Figure 23. Android Pie image

## 7 Revision History

Table 9. Revision history

Revision number	Date	Substantive changes
P9.0.0_1.0.0-beta	11/2018	Initial release
P9.0.0_1.0.0-ga	01/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.
P9.0.0_2.0.0-ga	04/2019	i.MX 8M, i.MX 8QuadMax, i.MX 8QuadXPlus GA release.

**How to Reach Us:****Home Page:**[nxp.com](http://nxp.com)**Web Support:**[nxp.com/support](http://nxp.com/support)

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

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

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

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

© 2019 NXP B.V.

