

Android™ Quick Start Guide

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 AIY Board.....	9
5	Working with the i.MX 8QuadXPlus MEK Board.....	14
6	Working with the i.MX 8M Nano Board.....	19
7	Revision History.....	26

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 8QuadXPlus
- i.MX 8M Nano

Supported boards:

- EVK board and Platform
- MEK board and Platform
- AIY board and Platform



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

3.1 Board hardware

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

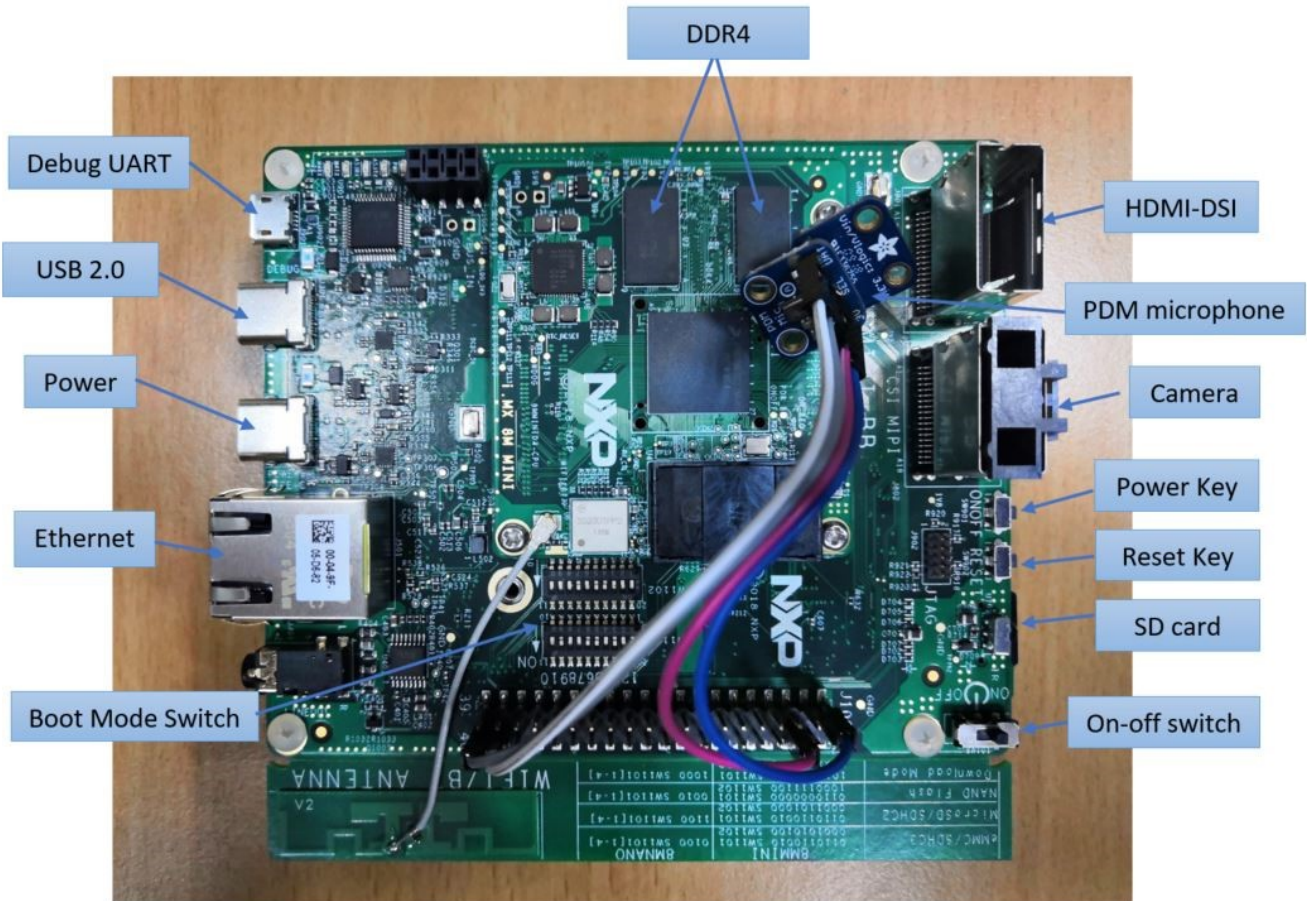


Figure 1. i.MX 8M Mini EVK board

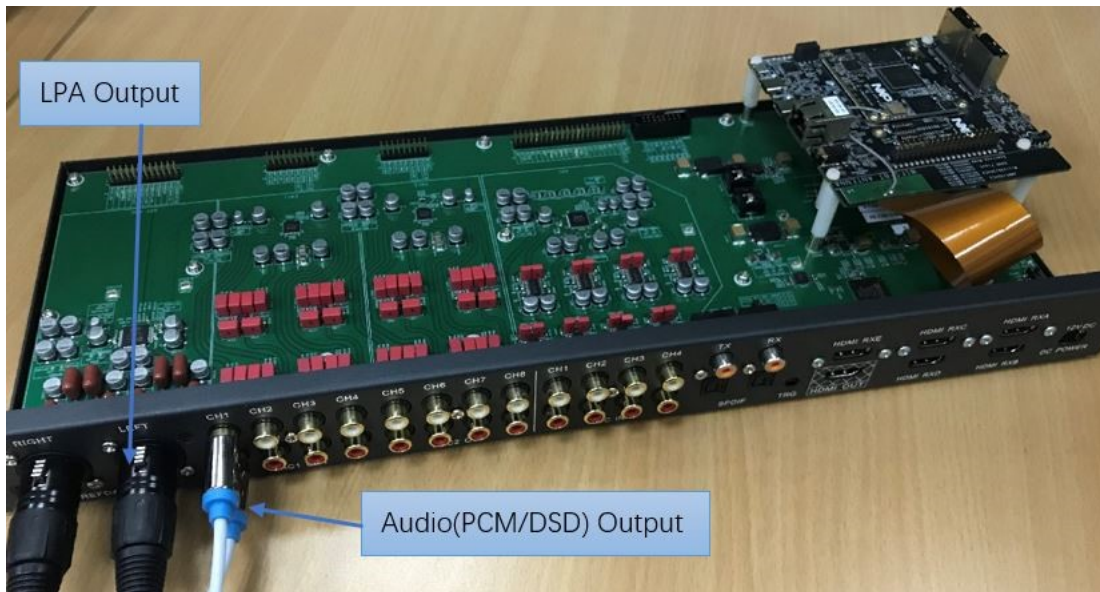


Figure 2. i.MX 8M Mini EVK with audio board



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_2.3.0_image_8mmevk.tar.gz`.

Table 1. Board images

Image name	Download target
<code>u-boot-imx8mm.img</code>	33 KB offset of MMC for a board with LPDDR4 on it.
<code>u-boot-imx8mm-ddr4.img</code>	33 KB offset of MMC for a board with DDR4 on it.
<code>u-boot-imx8mm-trusty.img</code>	33 KB offset of MMC for a board with LPDDR4 on it.
<code>u-boot-imx8mm-evk-uuu.img</code>	Bootloader used by UUU for the i.MX 8M Mini board with LPDDR4 on it. It is not flashed to MMC.
<code>u-boot-imx8mm-ddr4-evk-uuu.img</code>	Bootloader used by UUU for the i.MX 8M Mini board with DDR4 on it. It is not flashed to MMC.
<code>imx8mm_mcu_demo.img</code>	5120 KB offset of MMC.
<code>partition-table.img</code>	0 offset of MMC. If the actually size of the SD card is larger than 13 GB, use the default <code>partition-table.img</code> .
<code>partition-table-7GB.img</code>	0 offset of MMC. If the actually size of the SD card is larger than 7 GB, use this image as <code>partition-table.img</code> .
<code>partition-table-28GB.img</code>	0 offset of MMC. If the actually size of the SD card is larger than 28 GB, use this image as <code>partition-table.img</code> .
<code>boot.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions.
<code>vbmeta-imx8mm.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support LPDDR4 and MIPI-to-HDMI output and Direct Stream Digital (DSD) playback.
<code>vbmeta-imx8mm-ddr4.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support DDR4 and MIPI-to-HDMI output.
<code>vbmeta-imx8mm-m4.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support LPDDR4, MIPI-to-HDMI output, and audio playback based on Cortex-M4 FreeRTOS.
<code>vbmeta-imx8mm-mipi-panel.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support LPDDR4 and MIPI panel output.
<code>system.img</code>	<code>system_a</code> and <code>system_b</code> partitions.
<code>vendor.img</code>	<code>vendor_a</code> and <code>vendor_b</code> partitions.
<code>dtbo-imx8mm.img</code>	<code>dtbo_a</code> and <code>dtbo_b</code> partitions to support LPDDR4, MIPI-to-HDMI output, and DSD playback.
<code>dtbo-imx8mm-ddr4.img</code>	<code>dtbo_a</code> and <code>dtbo_b</code> partitions to support DDR4 and MIPI-to-HDMI output.
<code>dtbo-imx8mm-m4.img</code>	<code>dtbo_a</code> and <code>dtbo_b</code> partitions to support LPDDR4, MIPI-to-HDMI output, and audio playback based Cortex-M4 FreeRTOS.
<code>dtbo-imx8mm-mipi-panel.img</code>	<code>dtbo_a</code> and <code>dtbo_b</code> 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).

Working with the i.MX 8M Mini EVK Board

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 1.2.135 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 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_2.3.0_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 MCU image. If this option is not used, MCU 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.
-tos	Flashes the U-Boot image with Trusty OS integrated into it.
-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 -tos
```

- 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 -tos
```

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 the image with Trusty OS, execute the tool with "-tos".
- 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.
- The -tos mode only works for eMMC boot.
- 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 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
loop.max_part=7
```

```
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
loop.max_part=7
```

```
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
loop.max_part=7
```

```
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
loop.max_part=7
U-Boot > setenv bootcmd "bootmcu && boota mmc0" # for SD 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.

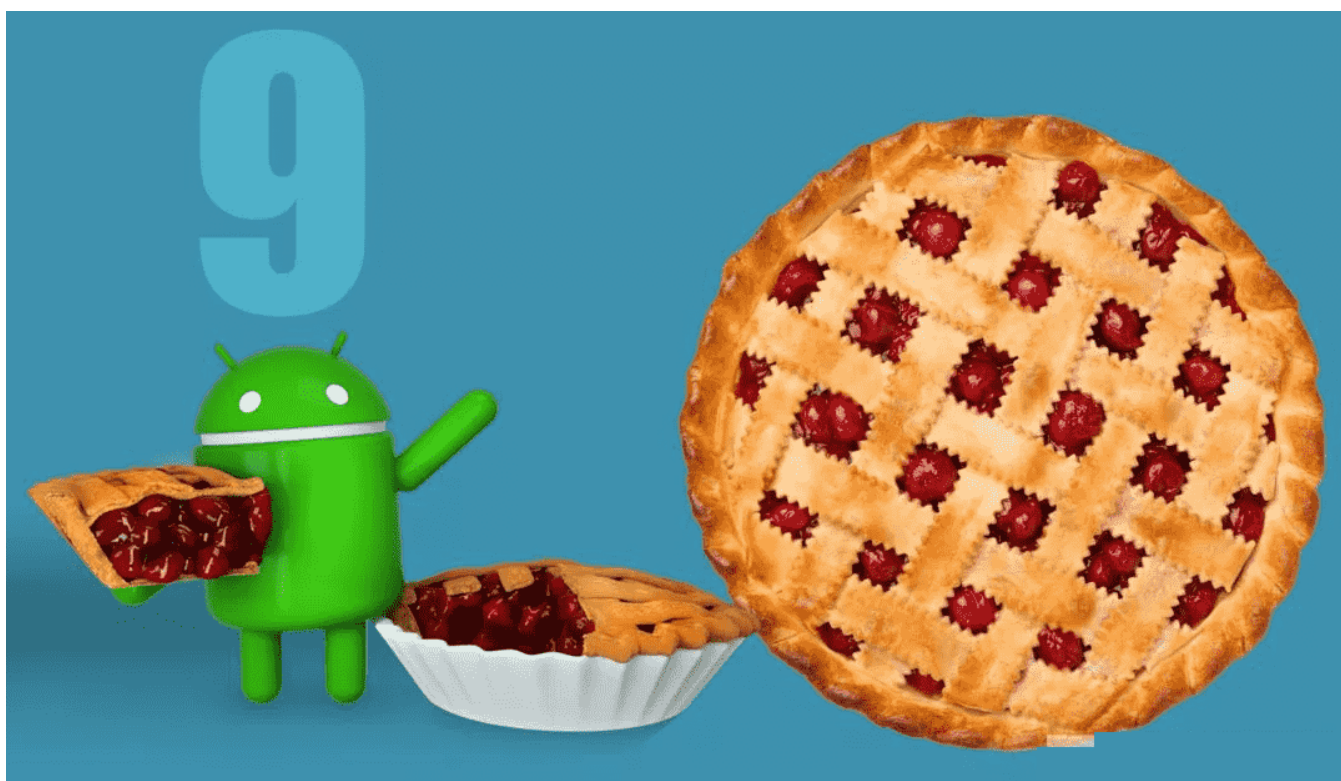


Figure 6. Android Pie image

4 Working with the i.MX 8M Quad AIY Board

4.1 Board hardware

This i.MX 8MQuad AIY board is identical with the Google Coral Dev Board for local AI. Mendel Linux system is maintained by Google based on this board. For more info about the Google Coral Dev Board, visit coral.withgoogle.com/.

NXP has enabled Android system on the i.MX 8M Quad AIY board. This document provides the software features of the Android system running on this board and its basic hardware information.

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

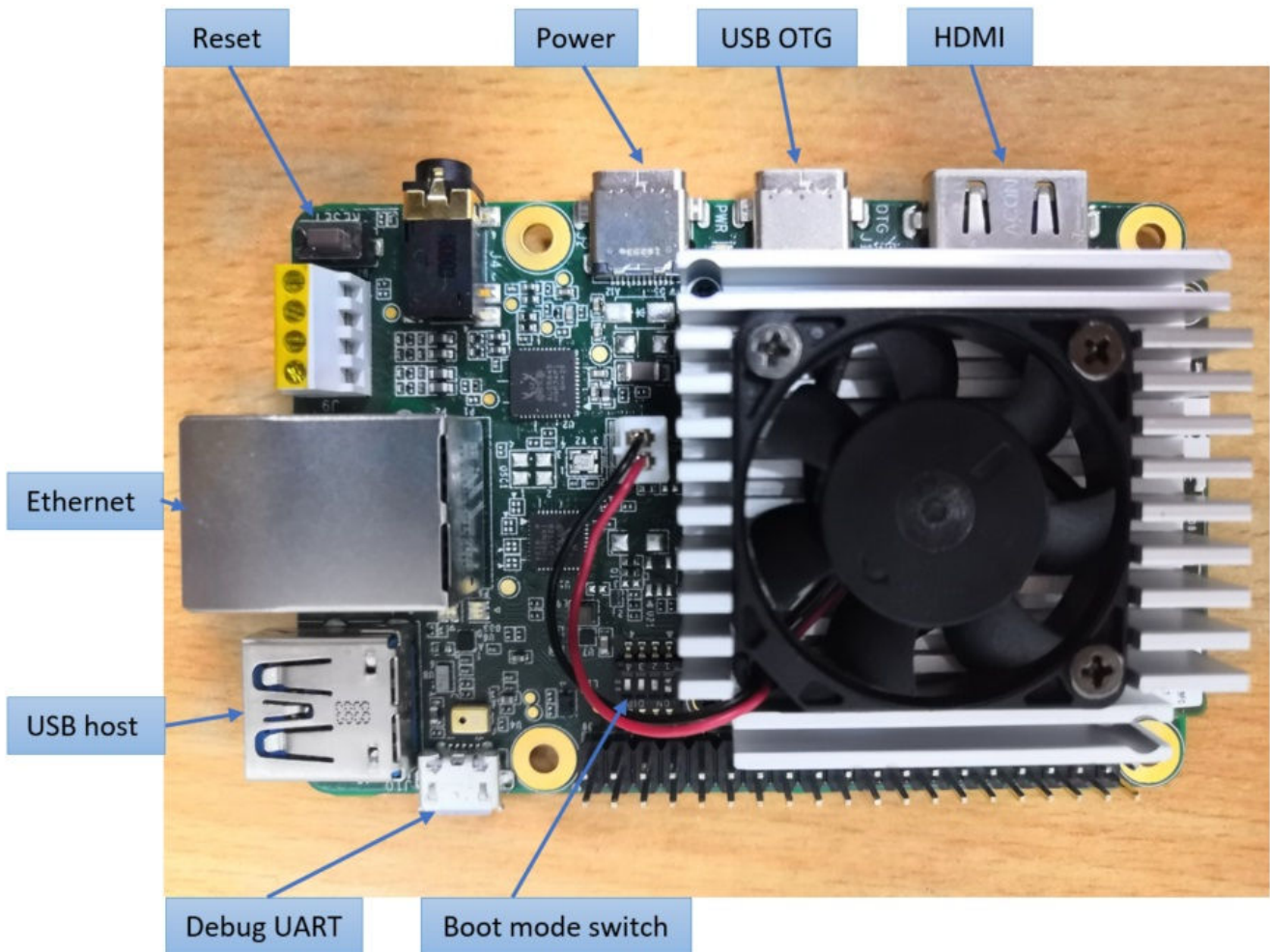


Figure 7. i.MX 8M Quad AIY board

4.2 Board images

The table below describes the location in the board partitions of the software images in `android_p9.0.0_2.3.0_image_8mqaiy.tar.gz`.

Table 3. Board images

Image name	Download target
u-boot-imx8mq.imx	33 KB offset of MMC.
u-boot-imx8mq-evk-uuu.imx	Bootloader used by UUU for the i.MX 8M Quad AIY board. It is not flashed to MMC.
boot.img	boot_a and 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 for storage with an actual size larger than 13 GB.
partition-table-7GB.img	Program to the first 17 KB, and then back up to the last 17 KB of the boot storage. GPT for storage with an actual size larger than 7 GB.
vbmata-imx8mq.img	vbmata_a and vbmata_b partitions for i.MX 8M Quad AIY to support HDMI display.
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 for i.MX 8M Quad AIY.

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 1.2.135 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 SW1 (boot mode) to 0100 (from 1-4 bit) to enter serial download mode for i.MX 8M Quad AIY.

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 AIY board: USB-to-UART, USB 3.0 host and USB 3.0 OTG.
- The USB-to-UART port can be referenced as debug UART to output log information of software running on the board.

4. Decompress `release_package/android_p9.0.0_2.3.0_image_8mqaiy.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 AIY 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 AIY, it should be "imx8mq". This option is mandatory.
-p platform	Specifies the platform information. For i.MX 8M Quad AIY, it should be "aiy". This option is mandatory for i.MX 8M Quad AIY.
-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	Specifies which partition table image file to flash. For 8M Quad AIY, it can be followed with "7". 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.
-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 -p aiy -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 -p aiy -e
```

When the command above is executed, the default images will be flashed into eMMC both slot a and slot b for i.MX 8M Quad AIY and all user data will be erased.

NOTE

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

```
> net use z: \\192.168.1.1\daily_images
```

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

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.
 - Change SW1 to switch the board back to 1000 (from 1-4 bit) to enter eMMC boot mode for i.MX 8M Quad AIY.

4.4 Booting

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

4.4.1 Booting with HDMI display

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

```
u-boot=> setenv bootargs console=ttymxc0,115200 earlycon=ec_imx6q,0x30860000,115200 init=/
init androidboot.gui_resolution=1080p androidboot.console=ttymxc0
androidboot.hardware=freescale androidboot.fbTileSupport=enable
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never androidboot.wificountrycode=CN loop.max_part=7
```

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
```

Save the environment variable set above.

```
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.

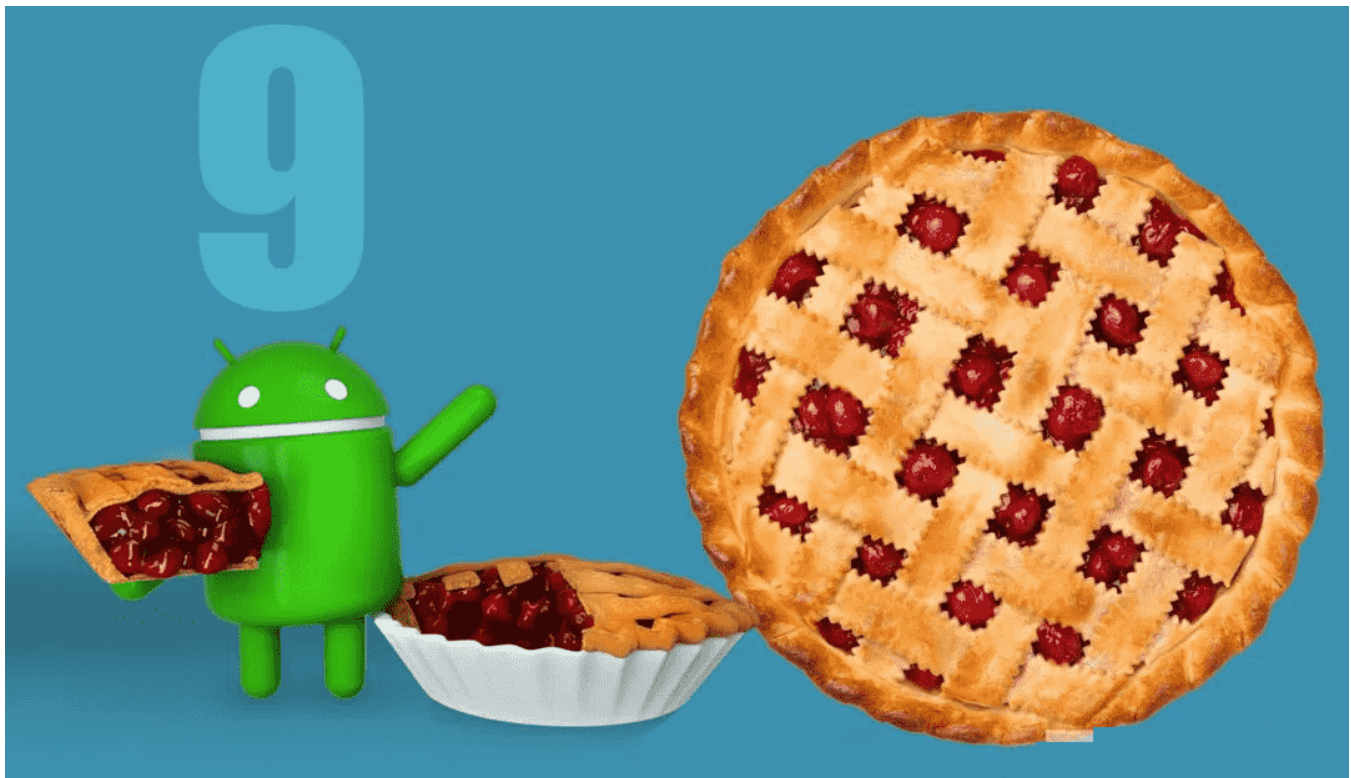


Figure 8. Android Pie image

5 Working with the i.MX 8QuadXPlus MEK Board

5.1 Board hardware

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

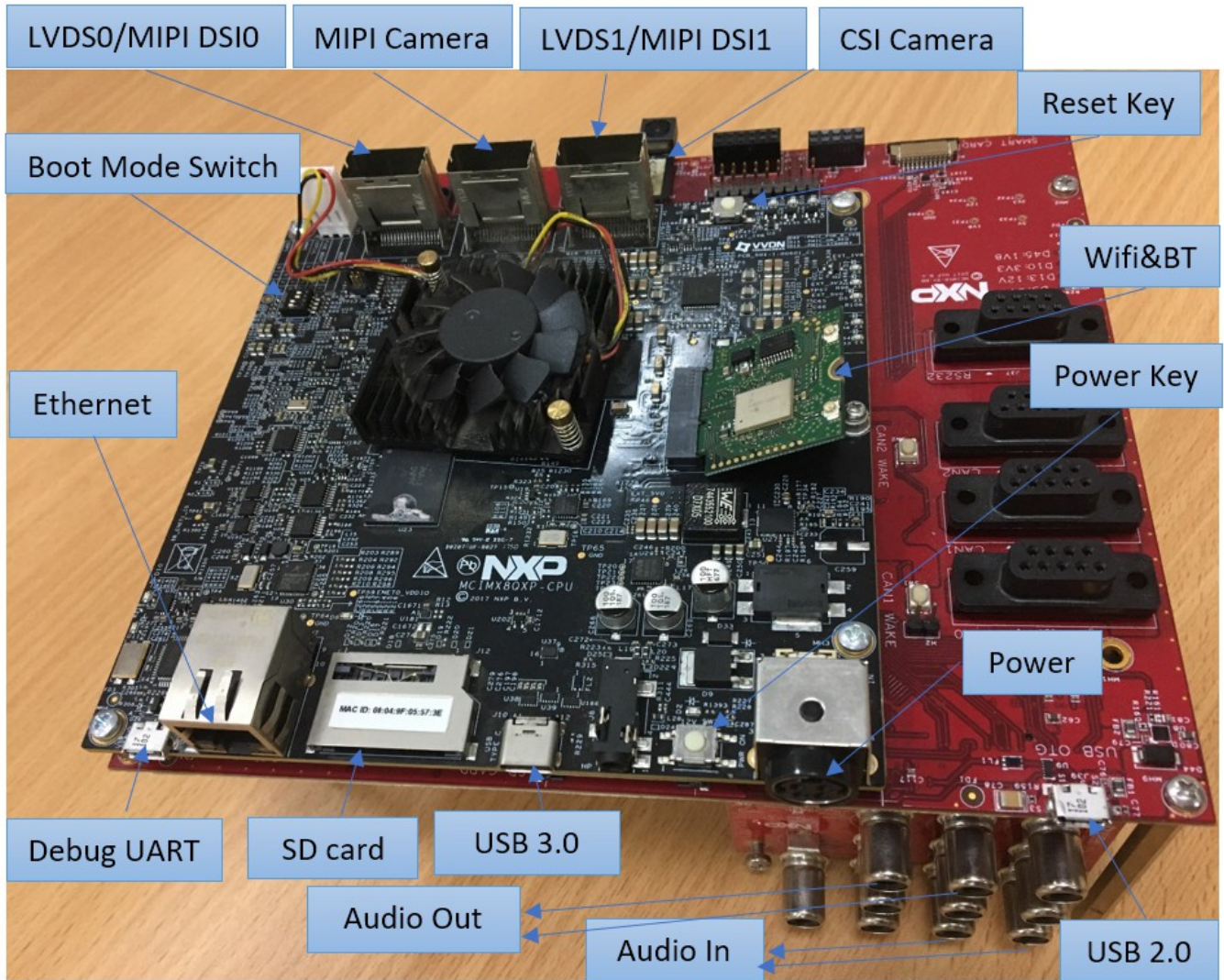


Figure 9. i.MX 8QuadXPlus MEK board



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

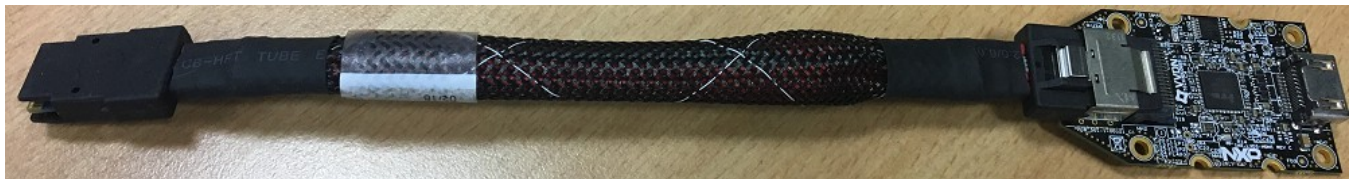


Figure 11. i.MX mini SAS cable with LVDS-to-HDMI adapter



Figure 12. 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.

5.2 Board images

The table below describes the location in the board partitions of the software images in `android_p9.0.0_2.3.0_image_8qmek.tar.gz`.

Table 5. Board images

Image name	Download target
u-boot-imx8qxp.img	32 KB offset of MMC.
u-boot-imx8qxp-mek-uuu.img	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
vmeta-imx8qxp.img	vmeta_a and vmeta_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
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.

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 1.2.135 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_2.3.0_image_8qmek.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
 5. Execute the `uuu_imx_android_flash` tool to flash images.
The `uuu_imx_android_flash` tool can be executed with options to get help information and specify the images to be flashed. For 8QuadXPlus board, related options are described as follows

Table 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 8QuadXPlus, it should be "imx8qxp". This option is mandatory.
-a	Only flashes slot a. If this option and "-b" option are not used, slots a and b are both flashed.
-b	Only flashes slot b. If this option and "-a" option are not used, slots a and b are both flashed.
-c card_size	Specifies which partition table image file to flash. For 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.

Table continues on the next page...

Table 6. Options for uuu_imx_android_flash tool (continued)

Option	Description
-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..

5.4 Booting

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

5.4.1 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 loop.max_part=7
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
```

5.5 Board reboot

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

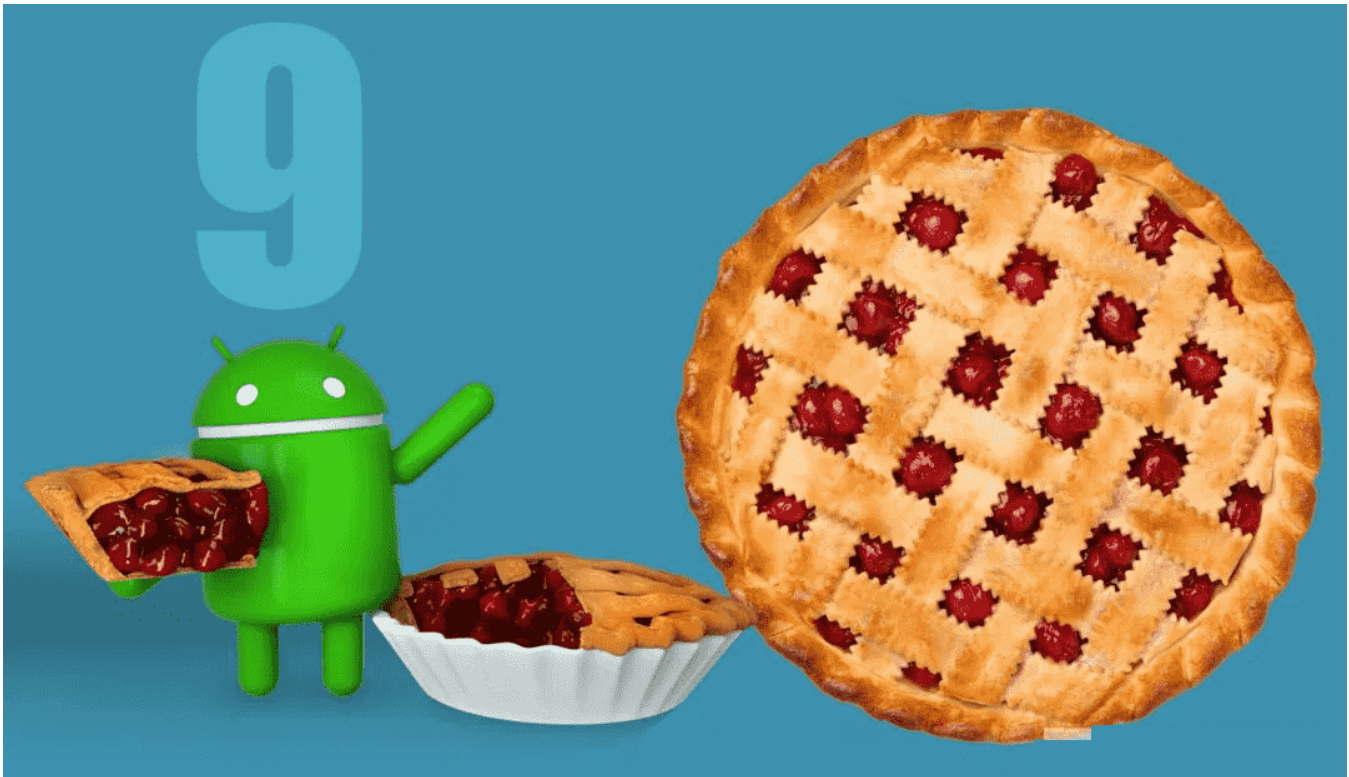


Figure 13. Android Pie image

6 Working with the i.MX 8M Nano Board

6.1 Board hardware

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

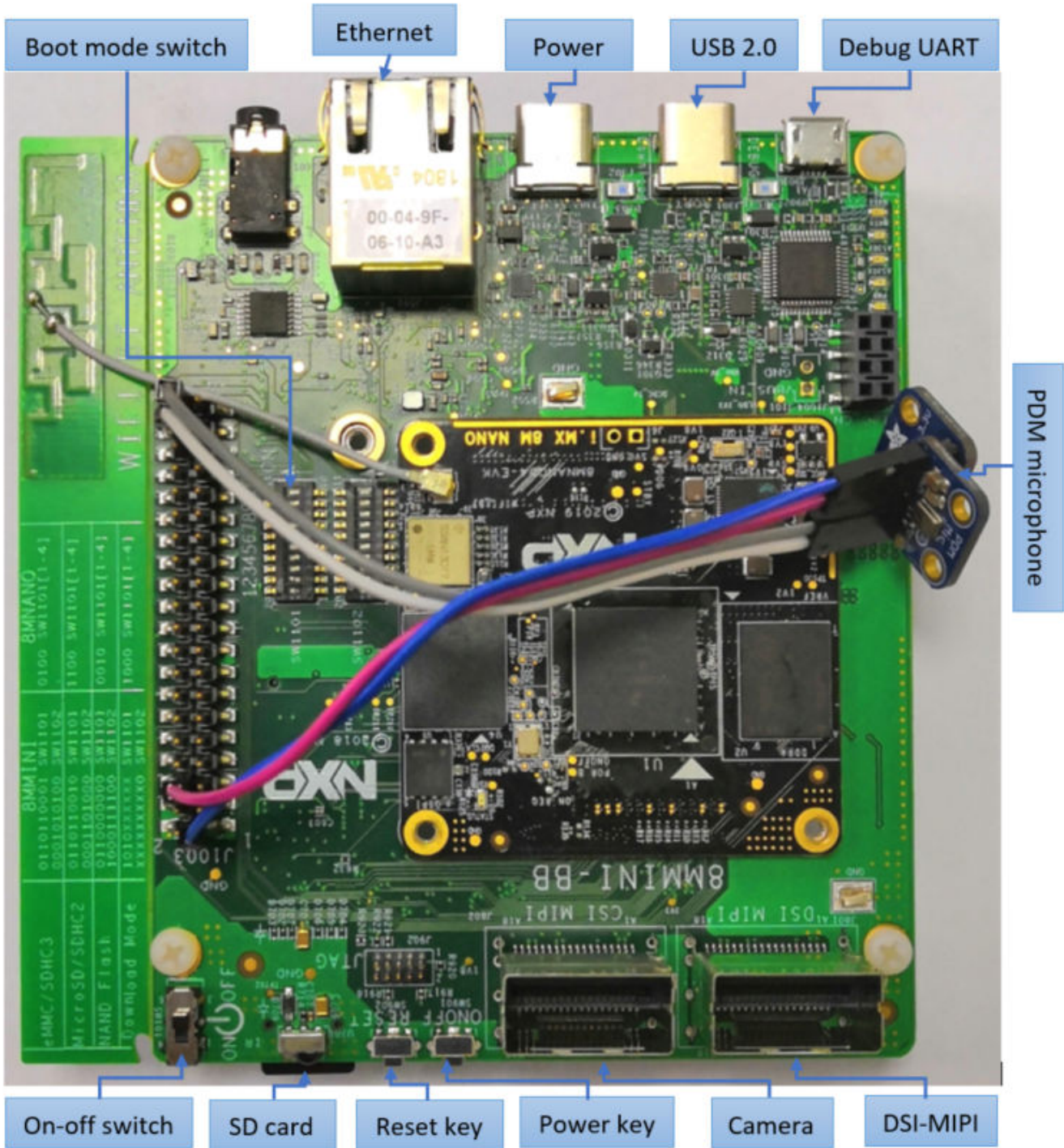


Figure 14. i.MX 8M Nano board



Figure 15. i.MX mini SAS cable with DSI-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 MIPI panel display, connect the i.MX MIPI panel to the "MIPI DSI" port.
- To test the camera, connect the i.MX CSI MIPI Camera to the "Camera" port.
- The BCM 1MW Wi-Fi/bluetooth module is used on the i.MX 8M Nano EVK DDR4 Board.

6.2 Board images

The table below describes the location in the board partitions of the software images in `android_p9.0.0_2.3.0_image_8mnev.k.tar.gz`.

Table 7. Board images

Image name	Download target
<code>u-boot-imx8mn.img</code>	32 KB offset of MMC for a board with DDR4 on it.
<code>u-boot-imx8mn-trusty.img</code>	32 KB offset of MMC for a board with DDR4 on it.
<code>u-boot-imx8mn-evk-uuu.img</code>	Bootloader used by UUU for i.MX 8M Nano board with DDR4 on it. It is not flashed to MMC.
<code>imx8mn_mcu_demo.img</code>	5120 KB offset of MMC.
<code>partition-table.img</code>	0 offset of MMC. If the actually size of your SD card is larger than 13 GB, use the default <code>partition-table.img</code> .
<code>partition-table-7GB.img</code>	0 offset of MMC. If the actually size of your SD card is larger than 7 GB, use this image as <code>partition-table.img</code> .
<code>partition-table-28GB.img</code>	0 offset of MMC. If the actually size of your SD card is larger than 28 GB, use this image as <code>partition-table.img</code> .
<code>boot.img</code>	<code>boot_a</code> and <code>boot_b</code> partitions
<code>vbmeta-imx8mn.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support DDR4 and MIPI-to-HDMI output.
<code>vbmeta-imx8mm-rpmsg.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support DDR4, MIPI-to-HDMI output and MCU image.
<code>vbmeta-imx8mn-mipi-panel.img</code>	<code>vbmeta_a</code> and <code>vbmeta_b</code> partitions to support DDR4 and MIPI panel output.
<code>system.img</code>	<code>system_a</code> and <code>system_b</code> partitions.
<code>vendor.img</code>	<code>vendor_a</code> and <code>vendor_b</code> partitions.
<code>dtbo-imx8mn.img</code>	<code>dtbo_a</code> and <code>dtbo_b</code> partitions to support DDR4 and MIPI-to-HDMI output.
<code>dtbo-imx8mn-rpmsg.img</code>	<code>dtbo_a</code> and <code>dtbo_b</code> partitions to support DDR4, MIPI-to-HDMI output and MCU image.
<code>dtbo-imx8mn-mipi-panel.img</code>	<code>dtbo_a</code> and <code>dtbo_b</code> partitions to support DDR4 and MIPI panel output.

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 1.2.135 version. Download corresponding version from github:

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

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

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described before. Install UUU into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
For Rev. C2 boards, change the first four bits of board's sw1101 to 1000 (from 1-4bit) 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_2.3.0_image_8mnev.k.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 Nano 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 i.MX 8M Nano, it should be "imx8mn". 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 8M Nano, it can be followed with "7" or "28". If this option is not used, default "partition-table.img" is flashed.
-m	Flashes the MCU image. If this option is not used, the MCU image is not flashed.
-d dev	Specifies some images with "dev" in its name. For i.MX 8M Nano, it can be "rpmmsg", "mipi-panel". If this option is not used, default dtbo and vbmeta images are flashed.
-tos	Flashes the U-Boot image with Trusty OS integrated into it.
-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 i.MX 8M Nano, 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 rpmmsg" 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 imx8mn -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 imx8mn -a -e
```

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

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 the image with Trusty OS, execute the tool with "-tos".
- To test MIPI-DSI to HDMI output with DDR4 on board, no need to use -d option.
- To test MIPI-DSI to HDMI output with DDR4 on board and MCU image, execute the tool with "-d rpmsg".
- To test MIPI panel output with DDR4 on board, execute the tool with "-d mipi-panel".
- 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. C2 boards:
 - To boot from eMMC, change sw1101 to 0100000000 and change sw1102 to 0000000000.
 - To boot from the SD card, change sw1101 to 1100000000 and change sw1102 to 0000000000.

6.4 Booting

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

6.4.1 Booting with single MIPI-to-HDMI/MIPI panel display

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


```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200 init=/
init androidboot.console=ttyMXC1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never loop.max_part=7 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
```

6.4.2 Booting with single MIPI-to-HDMI display with MCU image

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

```
U-Boot > setenv bootargs console=ttyMXC1,115200 earlycon=ec_imx6q,0x30890000,115200 init=/
init androidboot.console=ttyMXC1 androidboot.hardware=freescale cma=800M@0x400M-0xb80M
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
transparent_hugepage=never loop.max_part=7 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

If other dtbo image is used, do not add "bootmcu" to "bootcmd". Use the following command to recover bootcmd:

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

For how to use the Wi-Fi Country Code, see <https://community.nxp.com/docs/DOC-342877>.

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
```

6.5 Board reboot

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

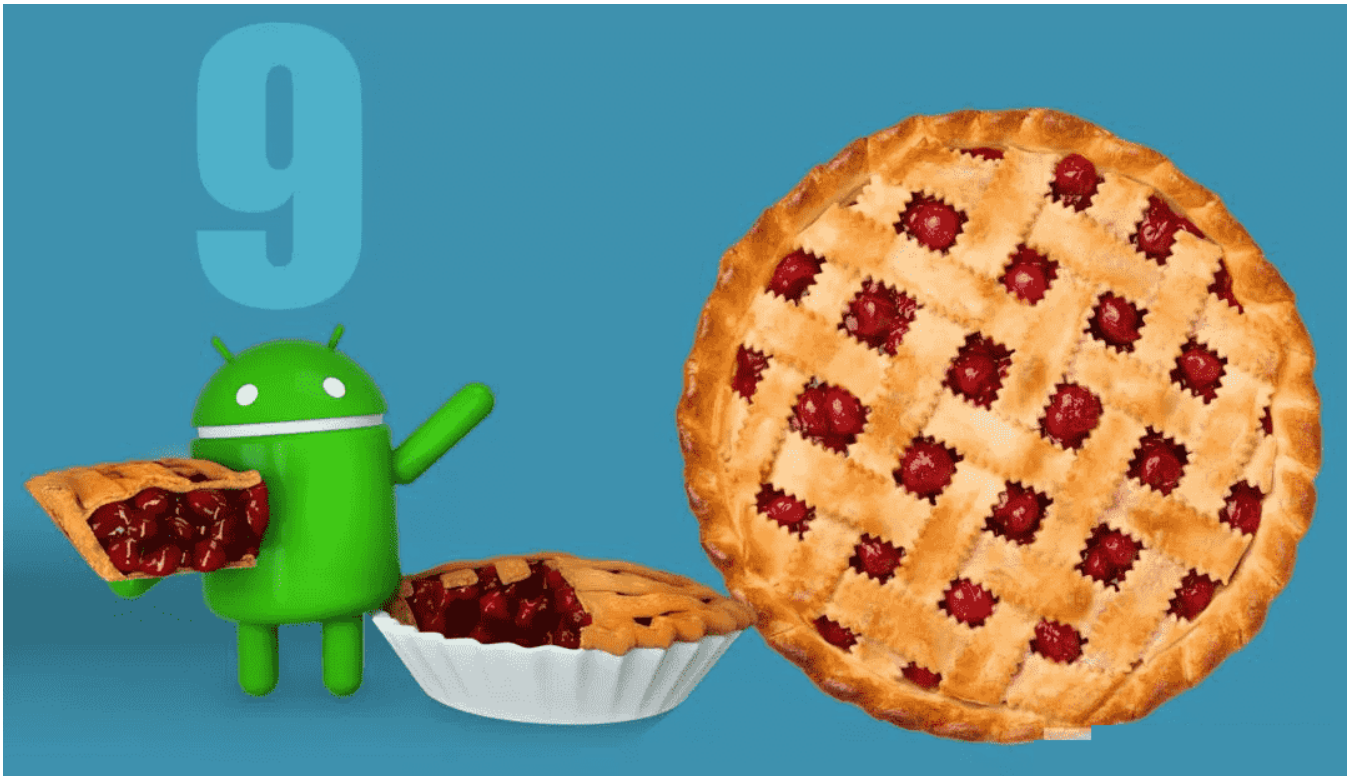


Figure 18. Android Pie image

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, and i.MX 8QuadXPlus GA release.
P9.0.0_2.0.0-ga	04/2019	i.MX 8M, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.
P9.0.0_2.3.0	08/2019	i.MX 8M Mini, i.MX 8M Quad, i.MX 8M Nano, and i.MX 8QuadXPlus Alpha release.

How to Reach Us:**Home Page:**nxp.com**Web Support:**nxp.com/support

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

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

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

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, 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, μ 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.

