

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 Nano
- i.MX 8QuadXPlus

Supported boards:

- EVK board and Platform
- MEK board and Platform

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

3.1 Board hardware

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

Contents

1 Overview	1
2 Hardware Requirements	1
3 Working with the i.MX 8M Nano EVK Board	1
4 Working with the i.MX 8QuadXPlus MEK Board	8
5 Revision History	14



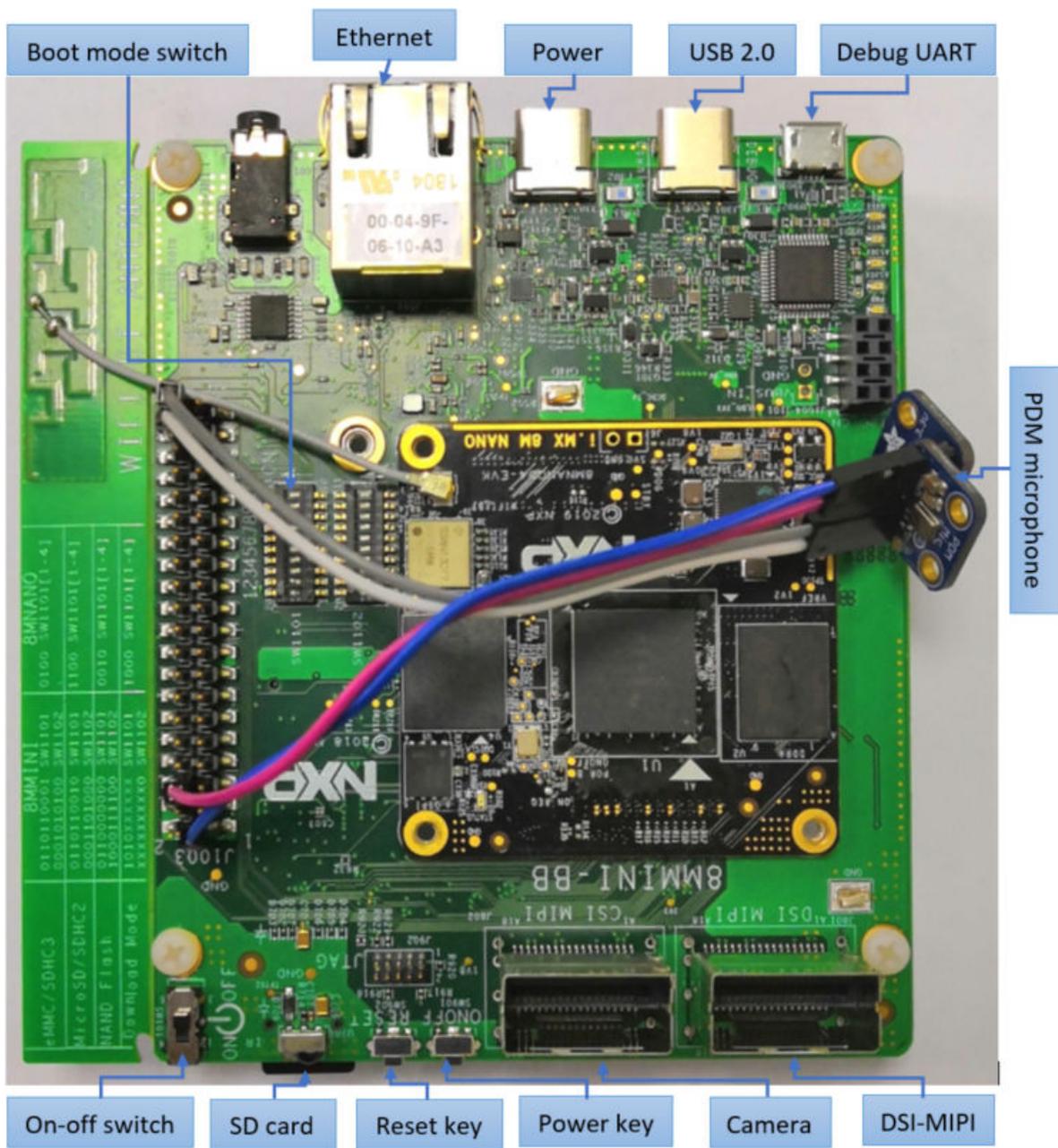


Figure 1. i.MX 8M Nano board



Figure 2. i.MX mini SAS cable with DSI-to-HDMI adapter



Figure 3. i.MX MIPI panel



Figure 4. 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 both on the i.MX 8M Nano EVK DDR4 board and i.MX 8M Nano EVK LPDDR4 board.

3.2 Board images

The table below describes the location in the board partitions of the software images in `android_p9.0.0_2.3.4_image_8mnevk.tar.gz`.

Image name	Download target
u-boot-imx8mn.imx	32 KB offset of MMC for i.MX 8M Nano EVK DDR4 board.
u-boot-imx8mn-lpddr4.imx	32 KB offset of MMC for for i.MX 8M Nano EVK LPDDR4 board.
u-boot-imx8mn-trusty.imx	32 KB offset of MMC for i.MX 8M Nano EVK DDR4 board.
u-boot-imx8mn-evk-uuu.imx	Bootloader used by UUU for i.MX 8M Nano EVK DDR4 board. It is not flashed to MMC.
u-boot-imx8mn-lpddr4-evk-uuu.imx	Bootloader used by UUU for i.MX 8M Nano EVK LPDDR4 board. It is not flashed to MMC.
imx8mn_mcu_demo.img	5120K offset of MMC.
partition-table.img	0 offset of MMC. If the actually size of your SD card is larger than 13 GB, you can use the default partition-table.img.
partition-table-7GB.img	0 offset of MMC. If the actually size of your SD card is larger than 7 GB, you can use this image as partition-table.img.
partition-table-28GB.img	0 offset of MMC. If the actually size of your SD card is larger than 28 GB, you can use this image as partition-table.img.
boot.img	boot_a and boot_b partitions.
vbmeta-imx8mn.img	vbmeta_a and vbmeta_b partitions to support MIPI-to-HDMI output.
vbmeta-imx8mm-rpmsg.img	vbmeta_a and vbmeta_b partitions to support MIPI-to-HDMI output and MCU image.
vbmeta-imx8mn-mipi-panel.img	vbmeta_a and vbmeta_b partitions to support DDR4 and MIPI panel output.
system.img	system_a and system_b partitions.
vendor.img	vendor_a and vendor_b partitions.
dtbo-imx8mn.img	dtbo_a and dtbo_b paritions to support MIPI-to-HDMI output.
dtbo-imx8mn-rpmsg.img	dtbo_a and dtbo_b paritions to support MIPI-to-HDMI output and MCU image.
dtbo-imx8mn-mipi-panel.img	dtbo_a and dtbo_b paritions to support 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](#).

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 1.3.102 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:

- Download the UUU binary file from github as described before. Place uuu in a directory contained by the system environment variable of "PATH". For example,
 - Linux OS users can put it in the directory of "/usr/sbin/", which is already contained by "PATH" with the system default configuration for all users. If uuu is put into a directory and the "PATH" environment is modified to include that directory, with "sudo" command to invoke UUU, the modified "PATH" may not take effect, since different users have different "PATH" values.
 - Windows OS users can put `uuu.exe` in any directory and then change the system variable of "PATH". "User variable" and "System variable" both need to be modified. After the modification, `cmd/powershell` needs to be restarted to make the modification take effect.

- 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.

- Power on the board. Use the USB cable on the board OTG port to connect your PC with the board.
- Decompress `release_package/android_p9.0.0_2.3.4_image_8mnevk.tar.gz`, which contains the image files and `uuu_imx_android_flash` tool.
- 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 1. Options for `uuu_imx_android_flash` tool

Option	Description
-h	Displays the help information of this tool.
-f soc_name	Specifies SoC information. For i.MX 8M Nano EVK, it should be "imx8mn". This option is mandatory .
-a	Only flashes slot a. If this option and "-b" are not used, a and b slot are both flashed.
-b	Only flashes slot b. If this option and "-a" are not used, a and b slot are both flashed.
-c card_size	Specifies which partition table image file to flash. For i.MX 8M Nano EVK, it can be followed with "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, the MCU image is not flashed.
-u uboot_feature	Flashes U-Boot image with "uboot_feature" in their names. For i.MX 8M Nano EVK, it can be "lpmddr4" or "trusty". If this option is not used, default "u-boot-imxmn.imx" is flashed.
-d dtbo_feature	Flashes dtbo and vbmeta images with "dtbo_feature" in their names. For i.MX 8M Nano EVK, it can be "rpsmsg" or "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 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 image is flashed. The script just loads U-Boot to RAM and executes to fastboot mode. This option is used for development.

Obviously, "-m" and "-d rpsmsg" 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 -u trusty
```

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

```
> .\uuu_imx_android_flash.bat -f imx8mn -a -e -u trusty
```

When the command above is executed, `u-boot-imx8mn-trusty.imx` with other default images are flashed into eMMC slot a for i.MX 8M Nano EVK DDR4 board.

NOTE

- "-u" followed with a parameter and containing "trusty" cannot be used together with "-t sd", because Trusty OS cannot boot from the SD card.
- To flash the SD card, execute the tool with "-t sd". To flash eMMC, it does not need to use -t option.
- If your SD card is 16 GB or the on-board eMMC is used as the boot device, it does not need to use -c option.
- If your SD card is 32 GB, execute the tool with "-c 28".
- If your SD card is 8 GB, execute the tool with "-c 7".
- To test the image on the i.MX 8M Nano EVK LPDDR4 board, execute the tool with "-u lpddr4".
- To test the image with Trusty OS on the i.MX 8M Nano EVK DDR4 board, execute the tool with "-u trusty".
- To test MIPI-DSI to HDMI output, it does not need to use -d option.
- To test MIPI-DSI to HDMI output and MCU image, execute the tool with "-m -d rpmsg".
- To test MIPI panel output, 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: \\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 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`.

3.4 Booting

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

3.4.1 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=ttyMX1,115200 earlycon=ec_imx6q,0x30890000,115200 init=/init
androidboot.console=ttyMX1 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
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended to the U-Boot's bootargs as follows:

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

3.4.2 Booting with single MIPI-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"
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"
U-Boot > saveenv
```

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

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended to the U-Boot's bootargs as follows:

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

3.5 Board reboot

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

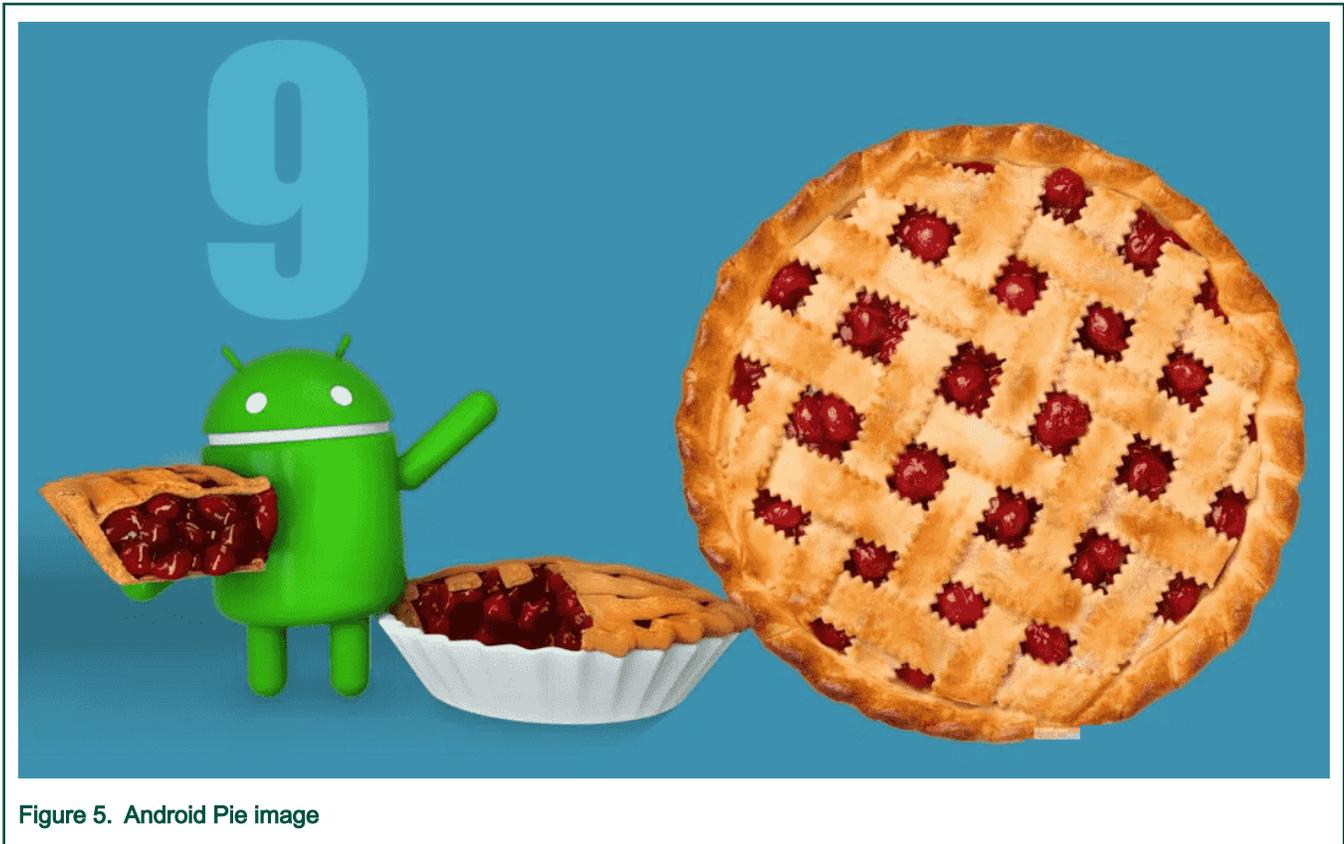


Figure 5. Android Pie image

4 Working with the i.MX 8QuadXPlus MEK Board

4.1 Board hardware

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

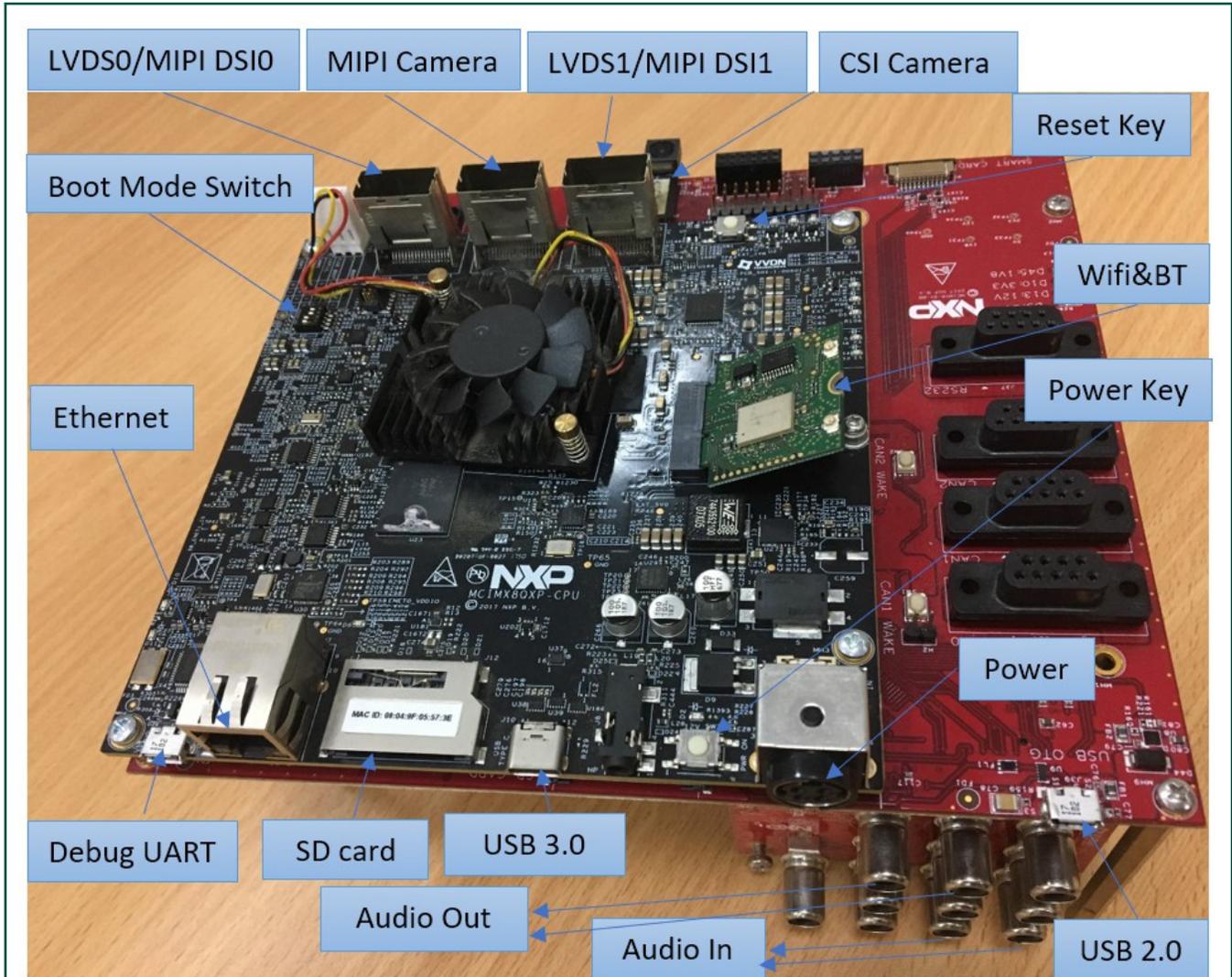


Figure 6. i.MX 8QuadXPlus MEK board



Figure 7. i.MX mini SAS cable with DSI-to-HDMI adapter



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



Figure 9. 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.

4.2 Board images

The table below describes the location in the board partitions of the software images in `android_p9.0.0_2.3.4_image_8qmek.tar.gz`.

Table 2. Board images

Image name	Download target
u-boot-imx8qxp.imx	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with b0 chip.
u-boot-imx8qxp-trusty.imx	32 KB offset of mmc for i.MX 8QuadXPlus MEK board with b0 chip.
u-boot-imx8qxp-mek-uuu.imx	Bootloader used by UUU for i.MX 8QuadXPlus MEK board with b0 chip. It is not flashed to MMC.
u-boot-imx8qxp-c0.imx	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with c0 chip.
u-boot-imx8qxp-trusty-c0.imx	32 KB offset of MMC for i.MX 8QuadXPlus MEK board with c0 chip.
u-boot-imx8qxp-mek-c0-uuu.imx	Bootloader used by UUU for i.MX 8QuadXPlus MEK board with c0 chip. 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 the last 17 KB of the boot storage. GPT table image for 8 GB boot storage.
partition-table-28GB.img	Program to the first 17 KB, and then back up to the last 17 KB of the boot storage. GPT table image for 32 GB boot storage.
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 displays with dual cameras support.
system.img	system_a and system_b partitions.

Table continues on the next page...

Table 2. Board images (continued)

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 cameras support.

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 1.3.102 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. Place uuu in a directory contained by the system environment variable of "PATH". For example,
 - Linux OS users can put it in the directory of "/usr/sbin/", which is already contained by "PATH" with the system default configuration for all users. If uuu is put into a directory and the "PATH" environment is modified to include that directory, with "sudo" command to invoke UUU, the modified "PATH" may not take effect, since different users have different "PATH" values.
 - Windows OS users can put `uuu.exe` in any directory and then change the system variable of "PATH". "User variable" and "System variable" both need to be modified. After the modification, `cmd/powershell` needs to be restarted to make the modification take effect.
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_2.3.4_image_8qxpek.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 3. Options for uuu_imx_android_flash tool

Option	Description
-h	Displays the help information of this tool.
-f soc_name	Specifies SoC information. For i.MX 8QuadXPlus MEK, it should be "imx8qxp". This option is mandatory .
-a	Only flashes slot a. If this option and "-b" are not used, a and b slot are both flashed.
-b	Only flashes slot b. If this option and "-a" are not used, a and b slot are both flashed.
-c card_size	Specifies which partition table image file to flash. For i.MX 8QuadXPlus, it can be followed with "7" or "28". If this option is not used, default "partition-table.img" is flashed.
-u uboot_feature	Flashes U-Boot image with "uboot_feature" in their names. For i.MX 8QuadXPlus MEK, it can be "trusty", "c0", or "trusty-c0". If this option is not used, default "u-boot-imx8qxp.imx" 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 i.MX 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 image is flashed. The script just 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 -u trusty
```

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

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

When the command above is executed, u-boot-imx8qxp-trusty.imx and other default images will be flashed into eMMC slot a for i.MX 8QuadXPlus MEK with b0 chip.

NOTE

- "-u" followed with a parameter and containing "trusty" cannot be used together with "-t sd", because Trusty OS cannot boot from the SD card.
- To flash the SD card, execute the tool with "-t sd". To flash eMMC, it does not need to use -t option.
- If your SD card is 16 GB or the on-board eMMC is used as the boot device, it does not need to use -c option.
- If your SD card is 32 GB, execute the tool with "-c 28".
- 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: \\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 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..

4.4 Booting

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

4.4.1 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
```

To disable selinux for userdebug/eng build mode images, "androidboot.selinux=permissive" needs to be appended to the U-Boot's bootargs as follows:

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

4.5 Board reboot

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

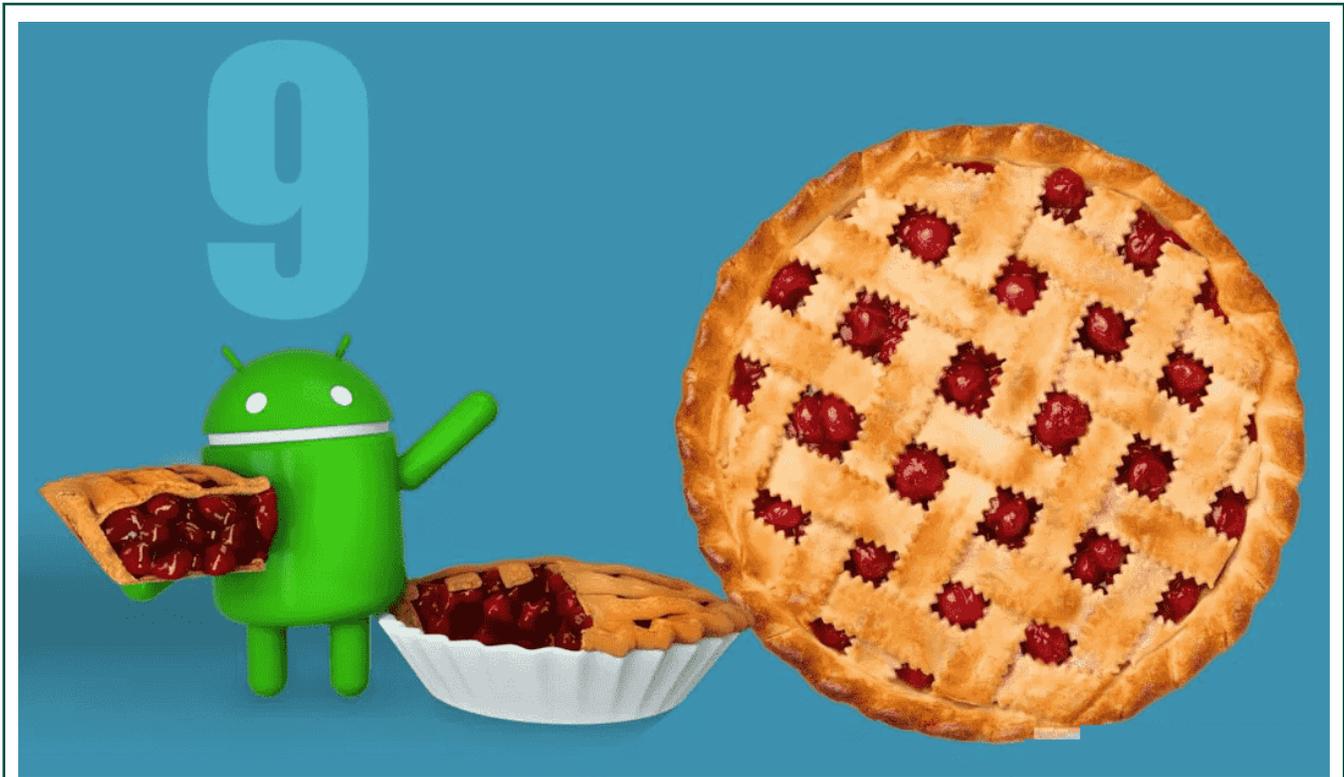


Figure 10. Android Pie image

5 Revision History

Table 4. 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.
P9.0.0_2.3.2	02/2020	i.MX 8M Nano, i.MX 8QuadMax, and i.MX 8QuadXPlus GA release.
P9.0.0_2.3.2	03/2020	Removed the i.MX 8QuadMax related content.
P9.0.0_2.3.4	03/2020	Fixed the communication issue between Cortex-A core and Cortex-M core for i.MX 8QuadXPlus MEK.

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, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D 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, µVision, 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.

© NXP B.V. 2018-2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 6 March 2020

Document identifier: AQSUG

