Android™ Quick Start Guide

Contents

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 8QuadXPlus/8QuadMax

Supported boards:

- i.MX 8QuadXPlus/8QuadMax MEK Board and Platform

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

3.1 Board hardware

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

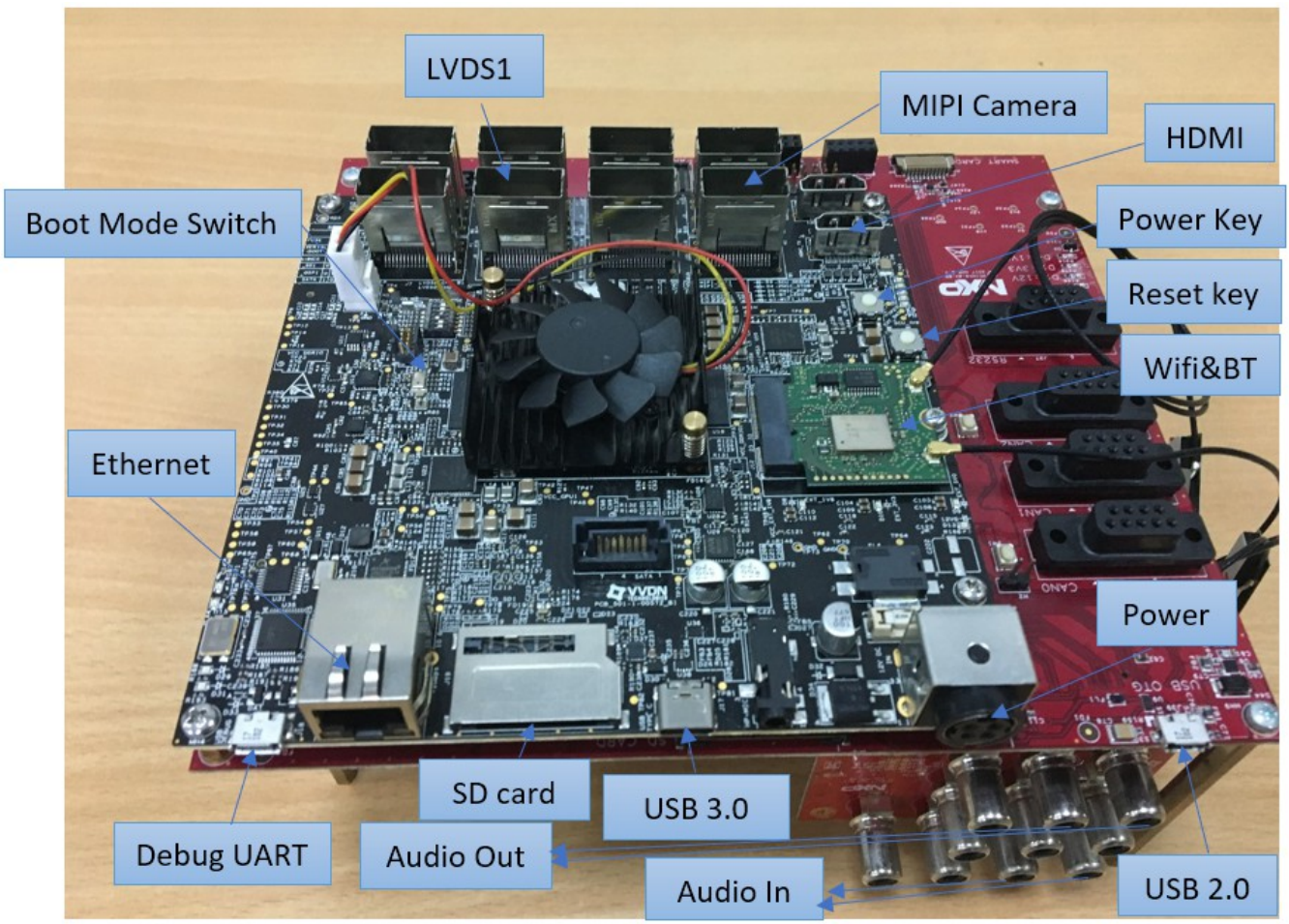

Figure 1. i.MX 8QuadMax MEK board

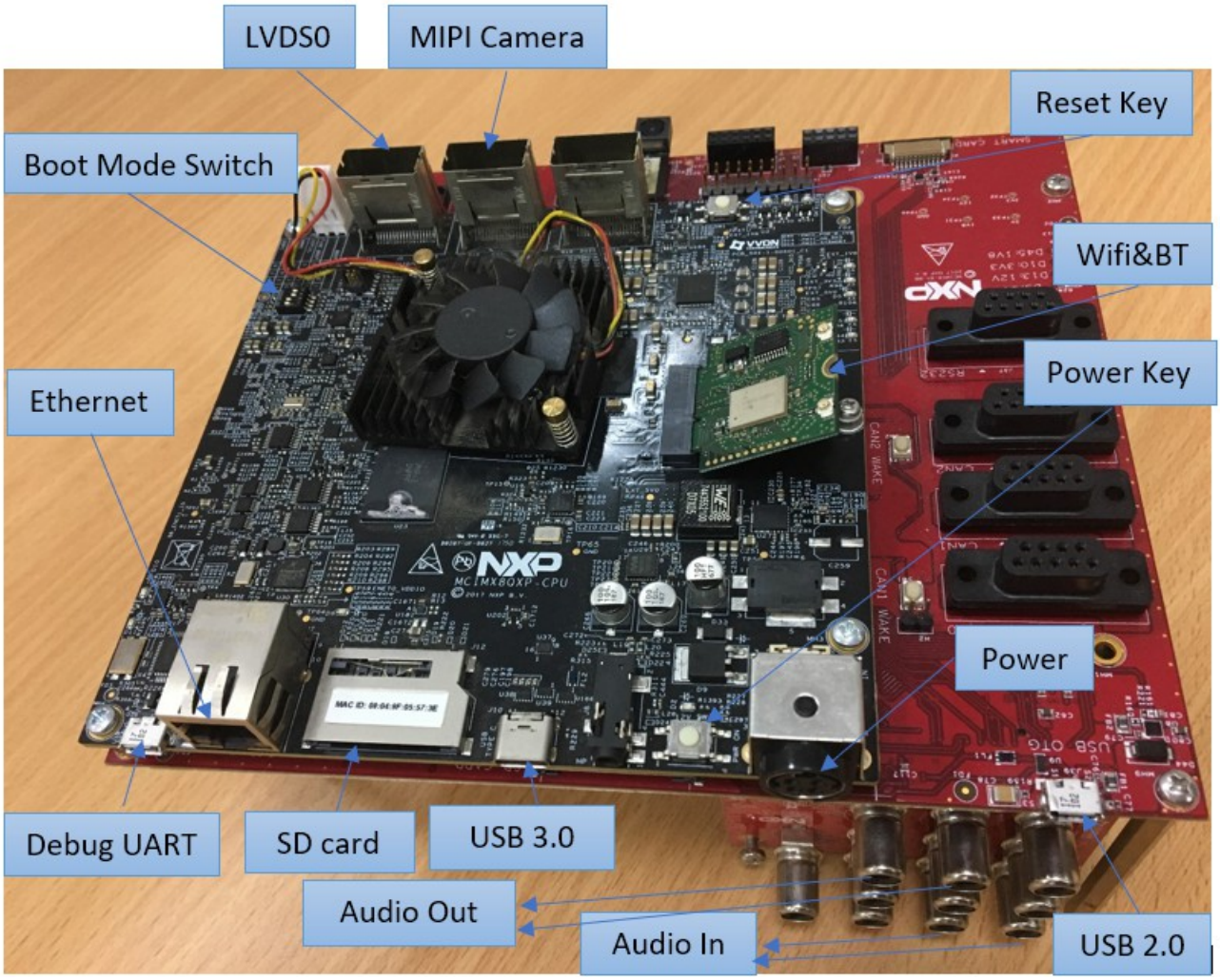

Figure 2. i.MX 8QuadXPlus MEK board

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

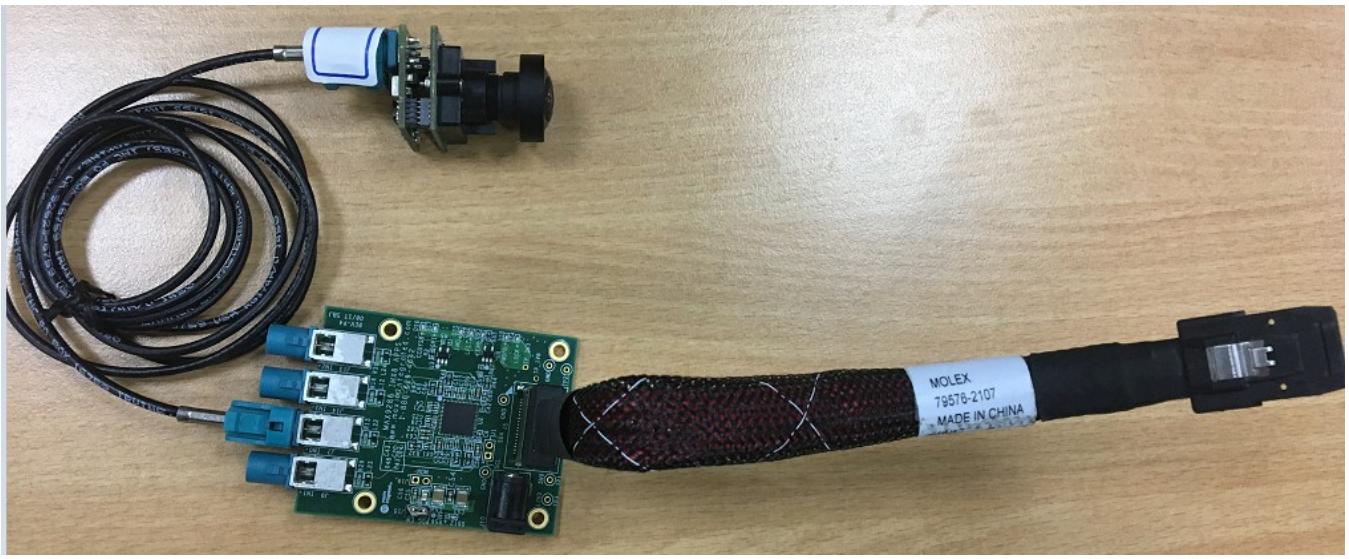

Figure 4. i.MX rearview camera (MAX9286)

NOTE

- To use i.MX rearview camera (MAX9286), connect the two pads of J15.
- i.MX 8QuadMax MEK
  - To test the display, connect the "LVDS1" port to the LVDS-to-HDMI adapter with the i.MX mini SAS cable.
  - To test the rearview camera, connect the "MIPI Camera" port with the i.MX MAX9286 MIPI camera.
- i.MX 8QuadXPlus MEK
  - To test the display, connect the "LVDS0" port to the LVDS-to-HDMI adapter with the i.MX mini SAS cable.
  - To test the rearview camera, connect the "MIPI Camera" port with the i.MX MAX9286 MIPI camera.
  - To use i.MX rearview camera (MAX9286), connect the two pads of J15.

3.2 Board images

To test prebuilt images with the EVS function enabled in the Arm Cortex-M4 CPU core, use android_o8.1.0_2.0.0-auto-beta_image_8qmek.tar.gz. To test prebuilt images without the EVS function enabled in the Arm Cortex-M4 CPU core, use android_o8.1.0_2.0.0-auto-beta_image_8qmek2.tar.gz.

The table below describes the location in the board partitions of the software images in android_o8.1.0_2.0.0-auto-beta_image_8qmek.tar.gz.

Table 1. Board images

Image name	Download target
/u-boot-imx8qm.imx	0K offset of MMC for i.MX 8QuadMax.
/u-boot-imx8qm-xen.imx	FAT partition on the SD card.
/u-boot-imx8qxp.imx	32K offset of MMC for i.MX 8QuadXPlus.
/u-boot-imx8qm-mek-uuu.imx	Bootloader used by UUU for i.MX 8QuadMax MEK board. It is not flashed to MMC.

Table continues on the next page...

Table 1. Board images (continued)

/u-boot-imx8qxp-mek-uuu.imx	Bootloader used by UUU for i.MX 8QuadXPlus MEK board. It is not flashed to MMC.
/boot.img	boot_a and boot_b partitions to support LVDS-to-HDMI display.
/partition-table.img	Program to first 17 KB, and then back up to last 17 KB of the boot storage. GPT table image for 16 GB boot storage.
/partition-table-7GB.img	Program to 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.
/vbmeta-imx8qm.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax to support LVDS-to-HDMI display.
/vbmeta-imx8qm-xen.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadMax to support LVDS-to-HDMI display on Xen.
/vbmeta-imx8qxp.img	vbmeta_a and vbmeta_b partitions for i.MX 8QuadXPlus to support LVDS-to-HDMI 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 for i.MX 8QuadMax.
/dtbo-imx8qm-xen.img	dtbo_a and dtbo_b partitions for i.MX 8QuadMax on Xen.
/dtbo-imx8qxp.img	dtbo_a and dtbo_b partitions for i.MX 8QuadXPlus.
/rpmb_key_test.bin	Prebuilt test RPMB key, which can be used to set the RPMB key as fixed 32 bytes 0x00.
/testkey_public_rsa4096.bin	Prebuilt AVB public key. It is extracted from the default AVB private key.

The table below describes UUU scripts in android_o8.1.0_2.0.0-auto-beta_image_8qmek and android_o8.1.0_2.0.0-auto-beta_image_8qmek2. They are used with the UUU binary file to download the images above into the board. For detailed information on how to download images with UUU, see Section 3.3 "Downloading Board Images".

Table 2. UUU scripts in android_o8.1.0_2.0.0-auto-beta_image_8qmek

UUU script name	Function
uuu-android-mx8qm-mek-emmc.lst	Used with the UUU binary file to download image files into eMMC on i.MX 8QuadMax.
uuu-android-mx8qxp-mek-emmc.lst	Used with the UUU binary file to download image files into eMMC on i.MX 8QuadXPlus.

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. You can download the latest version (1.2.31 for now).

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

You can put these files in a path containing the system environment variable "PATH", and then directly call uuu in command line or shell terminal.

There are two ways as follows to use UUU to flash images:

- Directly invoke UUU with the lst scripts in the command line to flash images.
- Use the uuu_imx_android_flash shell script or Windows batch file to invoke UUU and fastboot tool to flash images.

uuu_imx_android_flash is a new tool, which is more flexible. If the users are not familiar with this tool, the way to use UUU with the lst scripts is still maintained, and it will be removed in the future.

The two ways are described as follows. Users can choose either of it to flash images.

3.3.1 Directly invoking UUU with the lst scripts in command line to flash images

For detailed information on the UUU lst scripts used in this way, see Section 3.2 "Board images".

NOTE

UUU uses the integrated fastboot tool to flash images. Make sure you have fastboot driver software installed on your computer.

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described before.
2. Make the board enter serial download mode.
   - Change the board's SW2 (boot mode) to 001000 (1-6 bit) to enter serial download mode for i.MX 8QuadMax.
   - Change the board's SW2 (boot mode) to 1000 (1-4 bit) to enter serial download mode for i.MX 8QuadXPlus.
3. Power on the board. Use the USB cable on the board USB 3.0 type-c port to connect your PC with the board.

NOTE

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

4. Decompress release_package/android_o8.1.0_2.0.0-auto-beta_image_8qmek.tar.gz, which contains the image files and UUU scripts.

   Choose the correct UUU script file as shown in the following table.

Table 3. MFGTool VBS file

Target device and boot storage	UUU script file
i.MX 8QuadMax MEK eMMC	uuu-android-mx8qm-mek-emmc.lst
i.MX 8QuadXPlus MEK eMMC	uuu-android-mx8qxp-mek-emmc.lst

5. Use UUU and proper script file to flash image files.
   Execute the following command to invoke the UUU binary file and UUU script to flash the image files.
   - On the Linux system, open the shell terminal, and execute the following command. `${uuu_script_path}` is the file path (including the name of the UUU script) of the UUU script that is used. It can be a relative path or an absolute path.

```
> sudo uuu ${uuu_script_path}
```

- On the Windows system, open the cmd terminal, and execute the following command. `${uuu_script_path}` is the file path (including the name of the UUU script) of UUU script.

```
> uuu.exe ${uuu_script_path}
```

6. Wait for the script file execution to complete. If there is no error, the command line window displays the following information:

```
    PS C:\Users\user_01\tools\uuu> uuu.exe  C:\Users\user_01\images
\android_o8.1.0_2.0.0-auto-beta_image_8qmek\uuu-android-mx8qm-mek-emmc.lst
    uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.2.24-0-g0d63ca3

    Powershell: Enjoy auto [tab] command complete by run below command or put into
Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
        Register-ArgumentCompleter -CommandName uuu -ScriptBlock {param($commandName,
$parameterName,$wordToComplete,$commandAst,$fakeBoundParameter); C:\Users\user_01\tools
\uuu>uuu.exe -autocomplete $parameterName }

    Succuess 1    Failure 0

           1/ 0   [                              ]
    2:2   22/22   [Done                           ] FB: done
```

   As you can see, it is on the Windows system, and the absolute file path of the UUU script is used. For the output information on the command complete feature, it is recommendations given by UUU itself. This feature is not used here.
7. Power down the board.
8. Change the boot device as eMMC.
   - Change SW2 to switch the board back to 000100 (1-6 bit) to enther eMMC boot mode for i.MX 8QuadMax.
   - Change SW2 to switch the board back to 0100 (1-4 bit) to enter eMMC boot mode for i.MX 8QuadXPlus.

The following problems may be encountered when using UUU:

If the data speed of the target device is too slow, you may get the following prompts on the command line window when flashing system.img. In this situation, modify the UUU script file, and change the number after "-t" to a lager value. Currently, it is 100000, as shown below.

```
2:2    15/21   [Bulk read failure           ] FB[-t 100000]: flash system_a system.img
```

3.3.2 Using the uuu_imx_android_flash tool to invoke UUU and fastboot tool to flash images

The uuu_imx_android_flash shell script and windows batch file are provided to flash Android images with much more flexibility.

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described above. Install fastboot into a directory contained by the system environment variable of "PATH".
2. Make the board enter serial download mode.
   - Change the board's SW2 (boot mode) to 001000 (1-6 bit) to enter serial download mode for i.MX 8QuadMax.
   - Change the board's SW2 (boot mode) to 1000 (1-4 bit) to enter serial download mode for i.MX 8QuadXPlus.
3. Power on the board. Use the USB cable on the board USB 3.0 type-c port to connect your PC with the board.

NOTE

- There are three USB ports on the i.MX 8QuadMax/8QuadXPlus MEK board: USB-to-UART, USB 2.0, and USB 3.0.

- The USB-to-UART is known as debug UART, which can be used to watch the log of hardware boot processing.
- USB 2.0 is USB Host and USB 3.0 is USB OTG.

4. Decompress release_package/android_o8.1.0_2.0.0-auto-beta_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 Android Auto images on i.MX 8QuadMax/8QuadXPlus MEK 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 i.MX 8QuadMax, it should be "imx8qm". For i.MX 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.
-d dev	Specifies some images with "dev" in its name. For i.MX 8QuadMax, it can be "xen". For i.MX 8QuadXPlus, do not use this option. 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.

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

     ```
     > sudo ./uuu_imx_android_flash.sh -f imx8qm -e
     ```
   - On Windows system, open the command line interface, the corresponding command is as follows:

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

   When the command above is executed, the default images are flashed into the eMMC slot a for i.MX 8QuadMax.

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 (1-6 bit) to enter eMMC boot mode for i.MX 8QuadMax.
   - Change SW2 to switch the board back to 0100 (1-4 bit) to enter eMMC boot mode for i.MX 8QuadXPlus.

3.4 Booting with LVDS-to-HDMI display

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

```
U-Boot > setenv bootargs console=ttyLP0,115200 earlycon=lpuart32,0x5a060000,115200
androidboot.console=ttyLP0 androidboot.xen_boot=default init=/init consoleblank=0
androidboot.hardware=freescale androidboot.fbTileSupport=enable cma=800M@0x960M-0xe00M
galcore.contiguousSize=33554432 androidboot.primary_display=imx-drm firmware_class.path=/
```

```
vendor/firmware transparent_hugepage=never swiotlb=49152
U-Boot > saveenv
```

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

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

NOTE

i.MX 8QuadXPlus/8QuadMax MEK supports LVDS-to-HDMI display. They share the same bootargs.

3.5 Board reboot

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

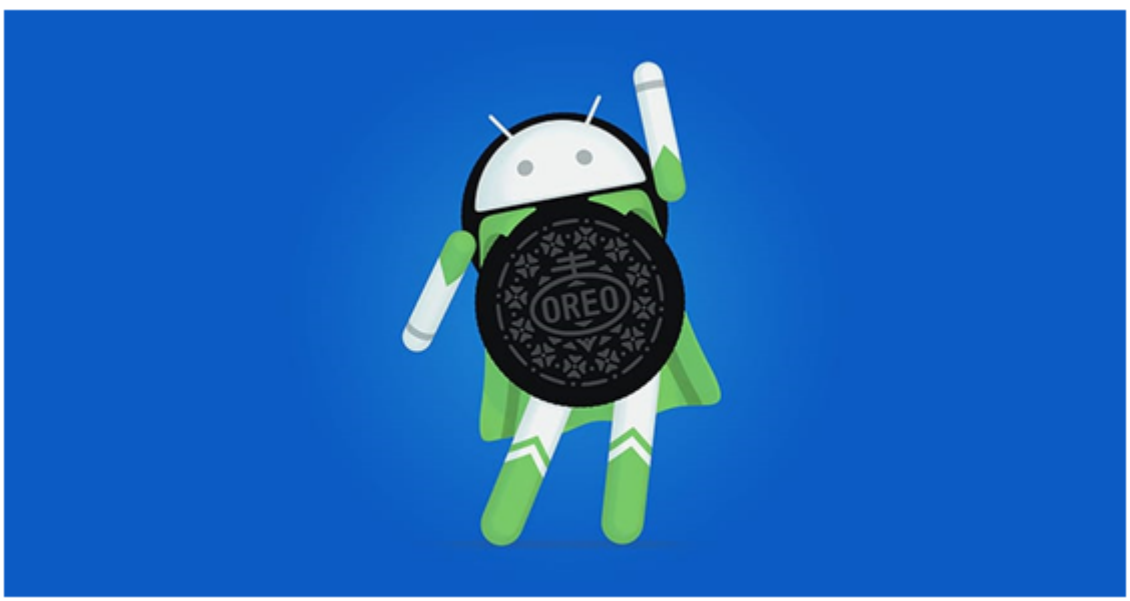

Figure 5. Android Oreo image

4 Revision History

Table 5. Revision history

Revision number	Date	Substantive changes
O8.1.0_1.1.0_AUTO-EAR	02/2018	Initial release
O8.1.0_1.1.0_AUTO-beta	05/2018	i.MX 8QuadXPlus/8QuadMax Beta release
O8.1.0_2.0.0-AUTO-beta	01/2019	i.MX 8QuadXPlus/8QuadMax Beta release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number AQSUG
Revision O8.1.0_2.0.0-AUTO-beta, 01/2019