

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.

1	Overview.....	1
2	Hardware Requirements.....	1
3	Working with the i.MX 8QuadXPlus/ 8QuadMax MEK Board.....	2
4	Revision History.....	8

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.

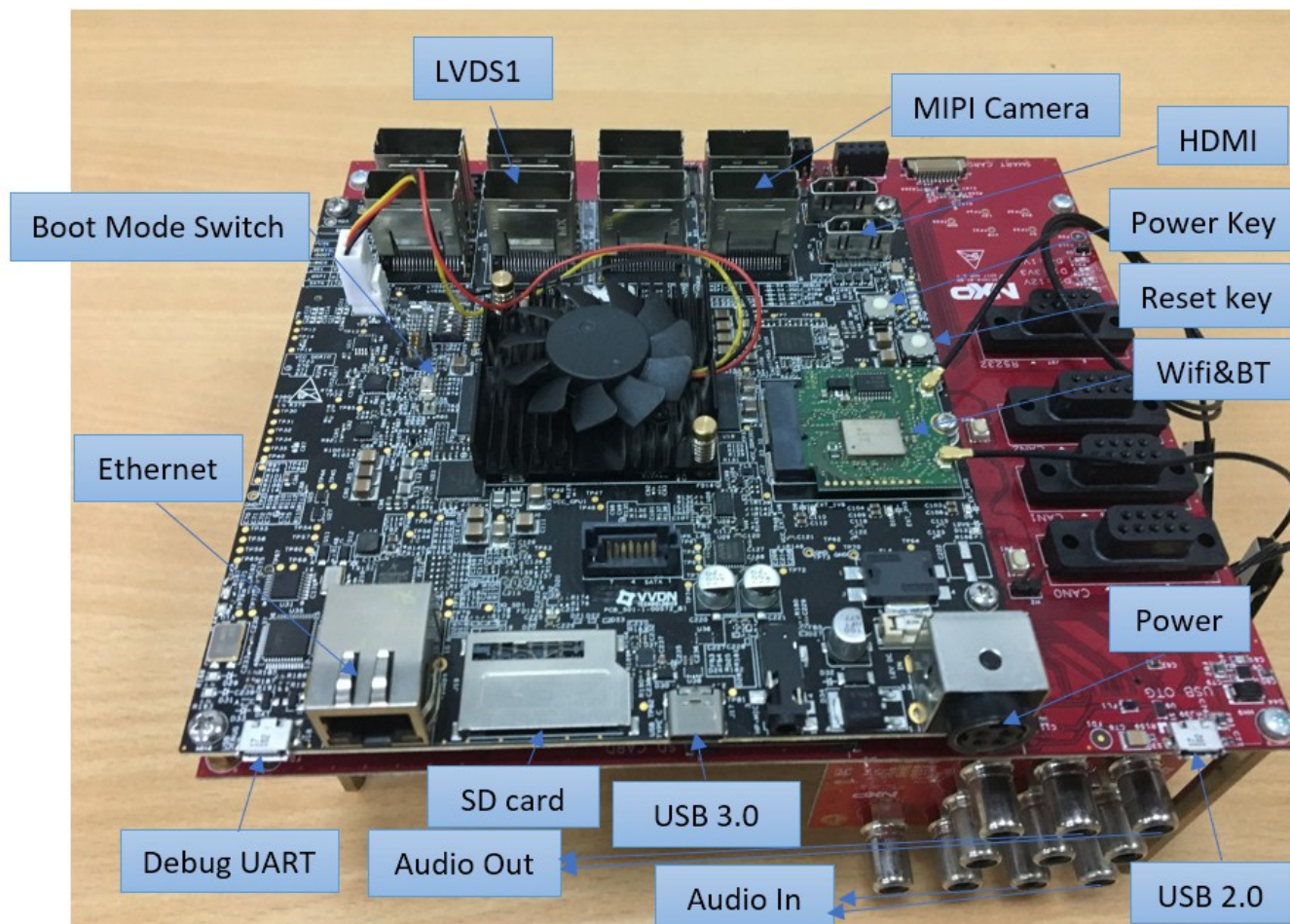


Figure 1. i.MX 8QuadMax MEK board

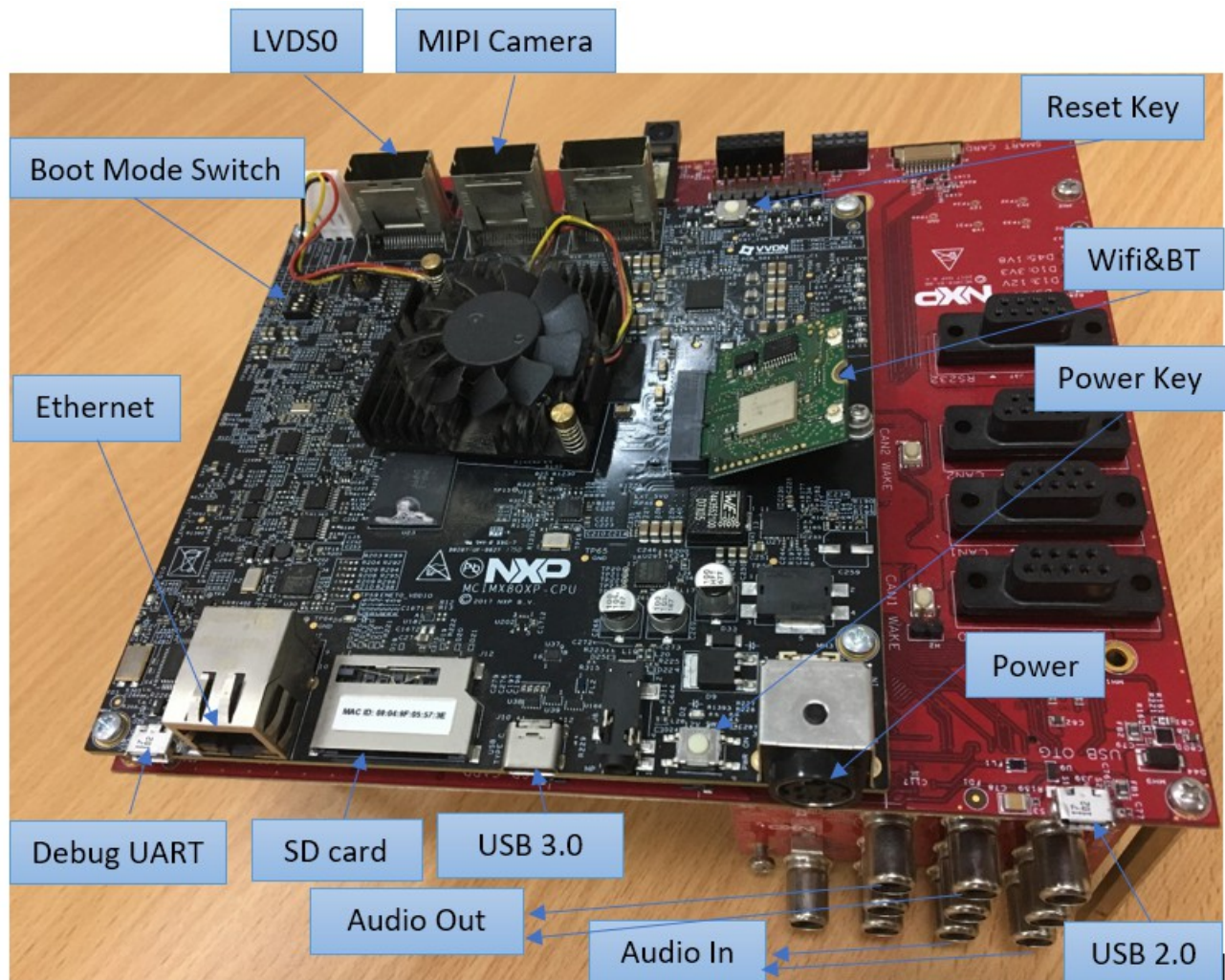


Figure 2. i.MX 8QuadXPlus/8QuadMax MEK board



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

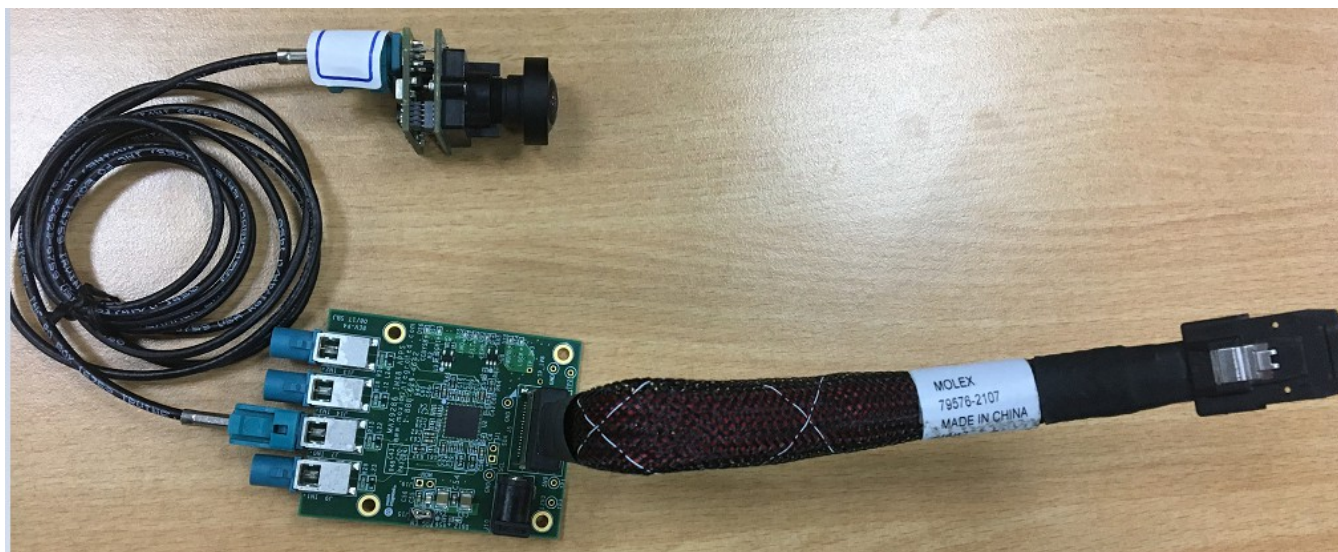


Figure 4. i.MX rearview camera (MAX9286)

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.

3.2 Board images

The table below describes the location in the board partitions of the software images in android_p9.0.0_1.0.2-auto-alpha_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	0K offset of MMC for i.MX 8QuadMax on Xen.
/u-boot-imx8qxp.imx	32K offset of MMC for i.MX 8QuadXPlus.
/uuu-u-boot-imx8qm.imx	Bootloader used by UUU for i.MX 8QuadMax MEK board. It is not flashed to MMC.
/uuu-u-boot-imx8qxp.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.

Table continues on the next page...

Table 1. Board images (continued)

/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.
/vbmata-imx8qm.img	vbmata_a and vbmata_b partitions for i.MX 8QuadMax to support LVDS-to-HDMI display.
/vbmata-imx8qm-xen.img	vbmata_a and vbmata_b partitions for i.MX 8QuadMax to support LVDS-to-HDMI display on Xen.
/vbmata-imx8qxp.img	vbmata_a and vbmata_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.

The table below describes UUU scripts in android_p9.0.0_1.0.2-auto-alpha_image_8qmek. 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_p9.0.0_1.0.2-auto-alpha_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 images can be flashed to the target board by using Universal Update Utility (UUU).

For UUU binary file, it can be downloaded from github: [uuu release page on github](#). You can download the latest version.

- For Linux OS users, download the file named "uuu".
- For Windows users, download the file named "uuu.exe" and "libusb-1.0.dll", which are in the same directory.

You can put these files in a path containing the system environment variable of "PATH". Then you can directly call uuu in cmd or shell terminal.

For details on UUU scripts, see Section 3.2 "Board Images".

NOTE

UUU uses the fastboot tool to flash images. Make sure that you have fastboot driver software installed on the computer that runs Windows OS.

Perform the following steps to download the board images:

1. Download the UUU binary file from github as described before.

- 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.
- 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.
- Decompress release_package/android_p9.0.0_1.0.2-auto-alpha_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

- 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}
```

- Wait for the script file execution to be completed. 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_p9.0.0_1.0.2-auto-alpha_image_8qmek\uuu-android-mx8qm-mek-emmc.lst
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.1.41-0-g668becf

Succuess 1      Failure 0

      1/ 0      [
2:2    21/21    [Done                      ] FB: done
```

As you can see, it is on the Windows system, and the absolute file path of the UUU script is used.

- Power down the board.
- Change the boot device as eMMC.
 - Change SW2 to switch the board back to 000100 (1-6 bit) to enter eMMC boot mode for i.MX 8QuadMax.
 - Change SW2 to switch the board back to 0100 (1-4 bit) to enter eMMC boot mode for i.MX 8QuadXPlus.

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 larger value. Currently, it is 100000, as shown below.

```

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

```

3.4 Booting with 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
androidboot.primary_display=imx-drm firmware_class.path=/vendor/firmware
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.

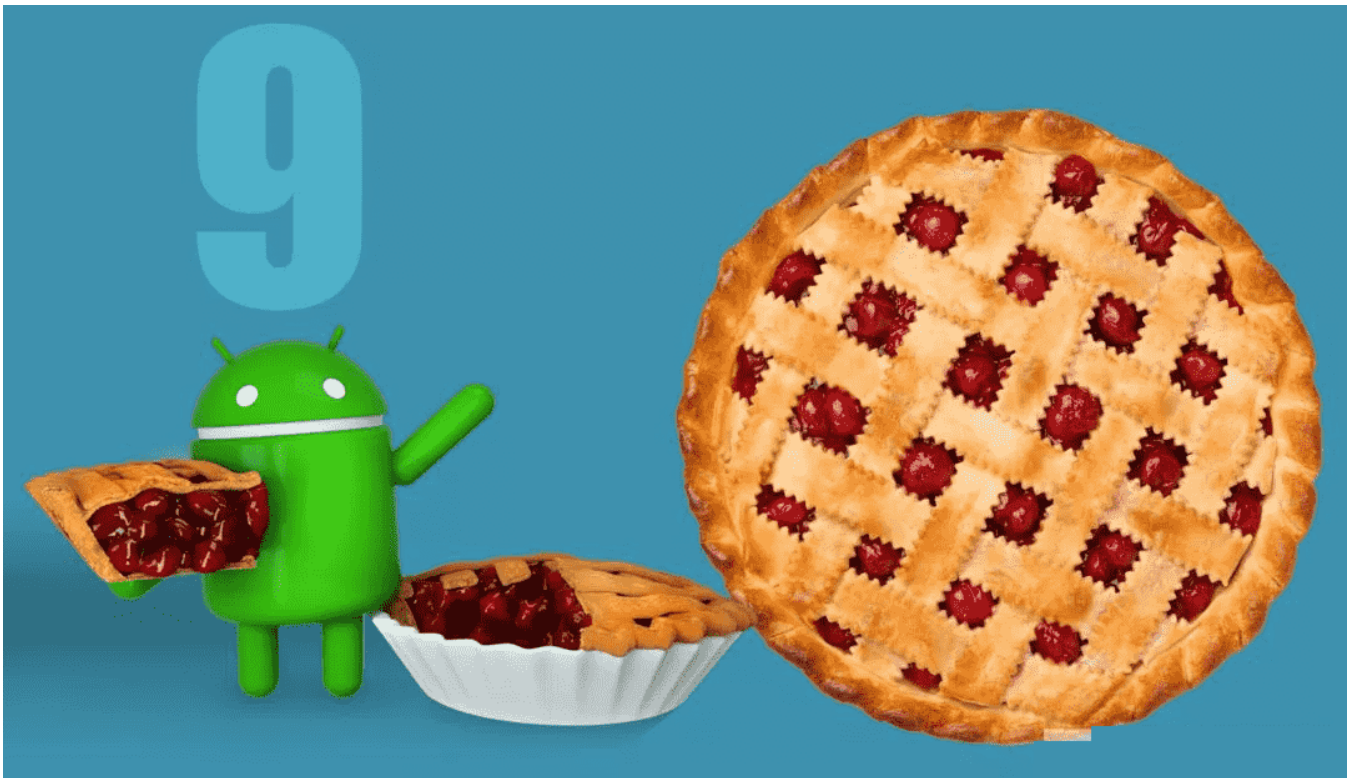


Figure 5. Android Pie image

4 Revision History

Table 4. 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
P9.0.0_1.0.2-AUTO-alpha	11/2018	i.MX 8QuadXPlus/8QuadMax Automotive 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.

© 2018 NXP B.V.

