NXP Semiconductors
User's Guide

Document Number: MCUXSDKIMX8MQRN
Rev. 0, 06/2019

MCUXpresso SDK Release Notes for i.MX8M Quad

Contents

Chapter 1
Overview

The MCUXpresso Software Development Kit (SDK) is a collection of software enablement for Microcontrollers that includes peripheral drivers, high-level stacks including other middleware packages, multicore support and integrated RTOS support for FreeRTOS™ OS. In addition to the base enablement, the MCUXpresso SDK is augmented with demo applications and driver example projects, and API documentation to help the customers quickly leverage the support of the MCUXpresso SDK.

For more details about MCUXpresso SDK, see the MCUXpresso SDK homepage MCUXpresso-SDK: Software Development Kit.

NOTE

See the attached Change Logs section at the end of this document to reference the device-specific driver logs, middleware logs, and RTOS log.

Chapter 2
MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpresso SDK is the evolution of Kinetis SDK v2.x.x, includes support for both LPC and i.MX System-on-Chips (SoC). The same drivers, APIs, and middleware are still available with support for Kinetis, LPC, and i.MX silicon. The MCUXpresso SDK adds support for the MCUXpresso IDE, an Eclipse-based toolchain that works with all MCUXpresso SDKs. Easily import your SDK into the new toolchain to have access to all of the available components, examples, and demos for your target silicon. In addition to the MCUXpresso IDE, support for the MCUXpresso Config Tools allows for easy cloning of existing SDK examples and demos, allowing users to easily leverage the existing software examples provided by the SDK for their own projects.

NOTE

In order to maintain compatibility with legacy Freescale code, the filenames and source code in MCUXpresso SDK containing the legacy Freescale prefix 'FSL' has been left as is. The 'FSL' prefix has been redefined as the NXP Foundation Software Library.

Chapter 3
Development tools

The MCUXpresso SDK was compiled and tested with these development tools:

- IAR Embedded Workbench for Arm version 8.32.2
- Makefiles support with GCC revision 7-2018-q2-update from Arm Embedded

Chapter 4
Supported development systems

This release supports boards and devices listed in this table. Boards and devices in boldface were tested in this release:

Table 1. Supported MCU devices and development boards

Development boards	MCU devices
MIMX8MQ-EVK	MIMX8MD6CVAHZ, MIMX8MD6DVAJZ, MIMX8MD7CVAHZ, MIMX8MD7DVAJZ, MIMX8MQ5CVAHZ, MIMX8MQ5DVAJZ, MIMX8MQ6CVAHZ, **MIMX8MQ6DVAJZ**, MIMX8MQ7CVAHZ, MIMX8MQ7DVAJZ

Chapter 5
Release contents

This table provides an overview of the MCUXpresso SDK release package contents and locations.

Table 2. Release contents

Deliverable	Location
Boards	<install_dir>/boards
Demo applications	<install_dir>/boards/<board_name>/demo_apps
Driver examples	<install_dir>/boards/<board_name>/driver_examples
Cortex Microcontroller Software Interface Standard (CMSIS) driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Multicore examples	<install_dir>/boards/<board_name>/multicore_examples
Documentation	<install_dir>/docs
Middleware	<install_dir>/middleware
Multicore stack	<install_dir>/middleware/multicore
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
CMSIS Arm Cortex®-M header files, DSP library source	<install_dir>/CMSIS
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
CMSIS drivers	<install_dir>/devices/<device_name>/cmsis_drivers
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools

Chapter 6
MCUXpresso SDK release package

The MCUXpresso SDK release package contents are aligned with the silicon subfamily it supports. This includes the boards, CMSIS, devices, documentation, middleware, and RTOS support.

6.1 Device support

The device folder contains all available software enablement for the specific System-on-Chip (SoC) subfamily. This folder includes clock-specific implementation, device register header file, device register feature header file, CMSIS derived device SVD, and the system configuration source files. Included with the standard SoC support are folders containing peripheral drivers, toolchain support, and a simple debug console.

The device-specific header files provide a direct access to the MCU peripheral registers. The device header file provides an overall SoC memory mapped register definition. In addition to the overall device memory mapped header file, the MCUXpresso SDK also includes the feature header file for each peripheral instantiated on the SoC.

The toolchain folder contains the startup code and linker files for each supported toolchain. The startup code is a CMSIScompliant startup that efficiently transfers the code execution to the main() function.

6.1.1 Board support

The boards folder provides the board-specific demo applications, driver examples, RTOS, and middleware examples.

6.1.2 Demo applications and other examples

The demo applications demonstrate the usage of the peripheral drivers to achieve a system level solution. Each demo application contains a readme file that describes the operation of the demo and required setup steps.

The driver examples demonstrate the capabilities of the peripheral drivers. Each example implements a common use case to help demonstrate the driver functionality.

6.2 Middleware

6.2.1 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.2 CMSIS

The MCUXpresso SDK is shipped with the standard CMSIS development pack, including the prebuilt libraries.

Chapter 7
MISRA compliance

All MCUXpresso SDK drivers and USB stack comply to MISRA 2012 rules with the following exceptions.

Table 3. MISRA exceptions

Exception Rules	Description
Directive 4.4	Sections of code should not be commented out.
Directive 4.5	Identifiers in the same name space with overlapping visibility should be typographically unambiguous.
Directive 4.6	Typedef that indicate size and signedness should be used in place of the basic numerical type.
Directive 4.8	If a pointer to a structure or union is never dereferenced within a transaction unit then the implementation of the object should hidden.
Directive 4.9	A function should be used in preference to a function like macro where they are interchangeable.
Directive 4.10	Precautions shall be taken in order to prevent the contents of a header file being included more than once.
Directive 4.11	The validity of values passed to library functions shall be checked.
Rule 2.3	A project should not contain unused type declarations.
Rule 2.4	A project should not contain unused tag declarations.
Rule 2.5	A project should not contain unused macro declarations.
Rule 2.7	There should be no unused parameters in functions.
Rule 3.1	The character sequences /* and // shall not be used within a comment.
Rule 5.1	External identifiers shall distinct.
Rule 5.3	A identifier declared in an inner scope shall not hide an identifier declared in an outer scope.
Rule 5.7	A tag name shall be a unique identifier.
Rule 5.9	Identifiers that define objects or functions with external linkage shall be unique.
Rule 8.13	A pointer should point to a const-qualified type whenever possible.
Rule 8.3	All declarations of an object or function shall use the same names and type qualifiers.
Rule 8.6	An identifier with external linage shall have exactly one external definition.
Rule 8.7	Octal constants shall not be used.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

Rule 8.9	A object should be defined at block scope if its identified only appears in a single function.
Rule 10.1	Operands shall not be of an inappropriate essential type.
Rule 10.3	The value of an expression shall not be assigned to an object with a narrower essential type of a different essential type category.
Rule 10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.
Rule 10.5	The value of an expression should not be cast to an inappropriate essential type.
Rule 10.6	The value of a composite expression shall not be assigned to an object with wider essential type.
Rule 10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type.
Rule 10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.
Rule 11.1	Conversions shall not be performed between a pointer to a function and any other type.
Rule 11.3	A case shall not be performed between a pointer to object type and a pointer to a different object type.
Rule 11.4	A conversion should not be performed between a pointer to object and an integer type.
Rule 11.5	A conversion should not be performed from pointer to void into pointer to object.
Rule 11.6	A cast shall not be performed between pointer to void and an arithmetic type.
Rule 12.1	The precedence of operators within expressions should be made explicit.
Rule 12.2	The right hand operator of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand.
Rule 13.3	A full expression containing an increment(++) or decrement(--) operator should have no other potential side effects other than that caused by the increment or decrement operator.
Rule 13.5	The right hand operand of a logical && or || operator shall not contain persistent side effects.
Rule 14.2	A for loop shall be well formed.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

Rule 14.4	The controlling expressions of an statement and the controlling expression of an iteration-statement shall have essentially Boolean type.
Rule 15.5	A function should have a single point of exit at the end.
Rule 16.1	All switch statements shall be well-formed.
Rule 17.1	The feature of <stdarg.h> shall not be used.
Rule 18.4	The +,-,+=and -=operators should not be applied to an expression of pointer type.
Rule 19.2	The union keyword should not be used.
Rule 20.1	#include directives should only be preceded by preprocessor directives or comments.
Rule 20.10	The #and ## preprocessor operators should not be used.
Rule 21.1	#define and #undef shall not be used on a reserved identifier or reserved macro name.

Chapter 8
Known issues

8.1 Maximum file path length in Windows 7® operating system

Windows 7 operating system imposes a 260 character maximum length for file paths. When installing the MCUXpresso SDK, place it in a directory close to the root to prevent file paths from exceeding the maximum character length specified by the Windows operating system. The recommended location is the `C:\nxp` folder.

8.2 Create new project without board template

The following components should be selected at the same time when creating a new project without using a board template, including serial_manager, serial_manager_uart, debug_console, and one UART adapter (lpuart_adapter for LPUART IP, uart_adapter for UART IP, lpsci_adapter for LPSCI IP, etc).

MCUXpresso SDK Release Notes for i.MX 8M Quad

Change Logs

Contents

1 Driver Change Log

ECSPI

The current eCSPI driver version is 2.0.1.

- 2.0.1
  - Memset local variable SDMA transfer configuration structure to make sure unused members in structure are cleared.
- 2.0.0
  - Initial version.

GPT

The current GPT driver version is 2.0.0.

- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.0.2.

- 2.0.2
  - Fixed bug enabling wrong GPIO clock gate in initial API. Since some GPIO instances may not have a clock gate enabled, it checks the clock gate number and makes sure the clock gate is valid.
- 2.0.1:
  - API interface changes:
    - Refined naming of API while keeping all original APIs, marking them as deprecated. Original API will be removed in next release. The main change is to update the API with prefix of _PinXXX() and _PortXXX().
- 2.0.0
  - Initial version.

I2C

The current I2C driver version is 2.0.5.

- 2.0.5
  - Fixed coverity issue of unchecked return value in I2C_RTOS_Transfer.
- 2.0.4
  - Bug fix:

        * Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) does not support the situation that master transfer with subaddress and transfer data size zero, which means no data follows by the subaddress.
- 2.0.3
  - Improvement:
    * Improved code readability, add new static API I2C_WaitForStatusReady for the status flag wait, and change to call I2C_WaitForStatusReady instead of polling flags with reading register.
- 2.0.2
  - Improvement:
    * Added I2C_WATI_TIMEOUT macro to allow user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.1
  - Bug fix:
    * Added proper handle for transfer config flag kI2C_TransferNoStartFlag to support transmit with kI2C_TransferNoStartFlag flag. Only supports write only or write+read with no start flag, does not support read only with no start flag.
- 2.0.0
  - Initial version.

PWM

The current PWM driver version is 2.0.0.

- 2.0.0
  - Initial version.

UART

The current UART driver version is 2.0.1.

- 2.0.1
  - Memset local variable SDMA transfer configuration structure to make sure unused members in structure are cleared.
- 2.0.0
  - Initial version.

MU

The current MU driver version is 2.0.2.

- 2.0.2
  - Added support for MIMX8MQx.
- 2.0.1

  - Added support for MCIMX7Ux_M4.
- 2.0.0
  - Initial version.

QSPI

The current QSPI driver version is 2.2.0.

- 2.2.0
  - New feature:
    * Added new API QSPI_ClearCache to clear cache for new IP feature FSL_FEATURE_QSPI_SOCCR_HAS_CLR_LPCAC.
  - Bug fix:
    * Fixed the QSPI_WriteBlocking API programming issue for low watermark, caused by previous improvement changing to use TX watermark signal to fill the TX FIFO. Revert change to previous implementation to use TX FIFO full flag for filling the FIFO, improve previous API by accessing TX data register directly.
    * Fixed the issue that QSPI_SetIPCommandSize incorrectly triggers a transaction.
    * Fixed clock divider accurate issue when use internal QSPI internal divider.
    * fixed build fail issue for some devices not support API QSPI_SetDqsConfig for DQS configuration.
- 2.1.0
  - New feature:
    * Added new API QSPI_SetDqsConfig for DQS configuration.
  - Improvements:
    * Updated the QSPI_WriteBlocking API to fill the TX FIFO once there are bytes of TX watermark room in the FIFO. This improves the performance of filling TX FIFO when watermark is high.
- 2.0.2
  - New Macro function:
    * Added QSPI_LUT_SEQ() function for users to set LUT table easily.
    * Added LUT command macros for users to easy use.
  - Comment update:
    * Added the comments for the limitation of QSPI_ReadBlocking and QSPI_TransferReceiveBlocking.
- 2.0.1
  - New API:
    * QSPI_SetReadArea to set the read area.
  - Bug fix:
    * Fixed QSPI_UpdateLUT function only update first LUT issue.
    * Fixed issue that some function that hardcode QSPI0 as base.
- 2.0.0
  - Initial version.

RDC

The current RDC driver version is 2.0.0.

- 2.0.0
  - Initial version.

RDC_SEMA42

The current RDC_SEMA42 driver version is 2.0.0.

- 2.0.0
  - Initial version.

SAI

The current SAI driver version is 2.2.1.

- 2.2.1
  - Improvements:
    - Added mclk post divider support in function SAI_SetMasterClockDivider.
    - Removed useless configuration code in SAI_RxSetSerialDataConfig.
  - Bug fix:
    - Fixed the SAI SDMA driver build issue caused by the wrong structure member name used in the function SAI_TransferRxSetConfigSDMA/SAI_TransferTxSetConfigSDMA.
    - Fixed BAD BIT SHIFT OPERATION issue caused by the FSL_FEATURE_SAI_CHANNEL_COUNTn.
    - Apply ERR05144: not set FCONT = 1 when TMR $>$ 0, otherwise the TX may not work.
- 2.2.0
  - Improvements:
    - Add new APIs for parameters collection and user interfaces simplify: SAI_Init SAI_SetMasterClockConfig SAI_TxSetBitClockRate SAI_TxSetSerialDataConfig SAI_TxSetFrameSyncConfig SAI_TxSetFifoConfig SAI_TxSetBitclockConfig SAI_TxSetConfig SAI_TxSetTransferConfig SAI_RxSetBitClockRate SAI_RxSetSerialDataConfig SAI_RxSetFrameSyncConfig SAI_RxSetFifoConfig SAI_RxSetBitclockConfig SAI_RXSetConfig SAI_RxSetTransferConfig SAI_GetClassicI2SConfig SAI_GetLeftJustifiedConfig SAI_GetRightJustifiedConfig SAI_GetTDMConfig
- 2.1.9
  - Improvements:
    - Improved SAI driver comment for clock polarity.
    - Added enumeration for SAI for sample inputs on different edge.
    - Changed FSL_FEATURE_SAI_CHANNEL_COUNT to FSL_FEATURE_SAI_CHAN-

NEL_COUNTn(base) for the difference between the different SAI instance.
        * Added new API: SAI_TxSetBitClockDirection SAI_RxSetBitClockDirection SAI_RxSetFrameSyncDirection SAI_TxSetFrameSyncDirection
- 2.1.8
  - Improvements:
    * Added feature macro test for the sync mode2 and mode 3.
    * Added feature macro test for masterClockHz in sai_transfer_format_t.
- 2.1.7
  - Improvements:
    * Added feature macro test for the mclkSource member in sai_config_t.
    * Changed "FSL_FEATURE_SAI5_SAI6_SHARE_IRQ" to "FSL_FEATURE_SAI_SAI5_SAI6_SHARE_IRQ".
    * Add #ifndef #endif check for SAI_XFER_QUEUE_SIZE to allow redefinition.
  - Bug fix:
    * Fixed build error caused by feature macro test for mclkSource.

-2.1.6

- Improvement:
  - Added feature macro test for mclkSourceClockHz check.
  - Added bit clock source name for general devices.
- Bug fix:
  - Fixed incorrect channel numbers setting while call RX/TX set format together.

-2.1.5

- Bug fix:
  - Corrected SAI3 driver IRQ handler name.
  - Added I2S4/5/6 IRQ handler.
  - Added base in handler structure to support different instances share one IRQ number.
- New features:
  - Updated SAI driver for MCR bit MICS.
  - Added 192 KHZ/384 KHZ in the sample rate enumeration.
  - Added multi FIFO interrupt/SDMA transfer support for TX/RX.
  - Added API to read/write multi FIFO data in a blocking method.
  - Added bclk bypass support when bclk is same with mclk.

2.1.4

- New feature:
  - Added API to enable/disable auto FIFO error recovery in platforms that support this feature.
  - Added API to set data packing feature in platform which support this feature.

2.1.3

- New feature:
  - Added feature to make I2S frame sync length configurable according to bitWidth.

2.1.2

- Bug fix:

  - Added 24-bit support for SAI eDMA transfer. All data shall be 32 bits for send/receive, as eDMA cannot directly handle 3 Byte transfer.

2.1.1

- Optimization:
  - Reduced code size while not using transactional API.

2.1.0

- API name change:
  - SAI_GetSendRemainingBytes -> SAI_GetSentCount.
  - SAI_GetReceiveRemainingBytes -> SAI_GetReceivedCount.
  - All transactional API name add "Transfer" prefix.
  - All transactional API use base and handle as input parameter.
  - Unify the parameter names.
- Bug fix:
  - Fixed WLC bug while reading TCSR/RCSR registers.
  - Fixed MOE enable flow issue, move MOE enable after MICS settings in SAI_TxInit/SAI_RxInit.

2.0.0

- Initial version.

SEMA4

The current SEMA4 driver version is 2.0.0.

- 2.0.0
  - Initial version.

TMU

The current TMU driver version is 2.0.2.

- 2.0.2
  - Fixed missing right pair definition for extern C.
- 2.0.1
  - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.
  - This module was first developed on i.MX 8MQuad.

WDOG

The current WDOG driver version is 2.1.0.

- 2.1.0
  - New Feature:
    - Added new API "WDOG_TriggerSystemSoftwareReset()" to allow user reset the system by software.
    - Added new API "WDOG_TriggerSoftwareSignal()" to allow user trigger a WDOG_B signal by software.
    - Removed the parameter "softwareAssertion" and "softwareResetSignal" out of the wdog_config_t structure.
    - Added new parameter "enableTimeOutAssert" to the wdog_config_t structure, with this parameter enabled, while the WDOG timeout occurred, a WDOG_B signal will be asserted, this signal can be routed to external pin of the chip. Note that WDOG_B signal remains asserted until a power-on reset (POR) occurs.
- 2.0.1
  - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

2 RTOS Change Log

FreeRTOS for MCUXpresso SDK.

The current version is Amazon-FreeRTOS 1.4.0 Original package is available at github.-com/aws/amazon-freertos.

- 1.4.7_rev0
  - New features:
    - Add optional allocation scheme heap_useNewlib.c by D. Nadler.
    - Enable task aware debugging for cm33 platforms
    - Move tickless implementation to application layer
  - Other changes:
    - Fix other build warnings, errors
- 1.4.6_rev0
  - New features:
    - Update support of CM33 port with Trustzone, MPU, FPU support
    - Add support for AWS test for Cypress WiFi
    - Use lwip netif api to avoid lwIP raw API calls outside of tcpip thread in aws_wifi.c
  - Other changes:
    - Fix issues with mflash driver
    - Fix other build warnings, errors
- 1.4.0_rev1
  - New features:
    - Add implementation of vTaskEndScheduler for CM0 GCC port.
    - Support for CM33, CM33F architectures based on CM3, CM4F ports
- 1.4.0_rev0
  - New features:
    - Support for pkcs11 for several platforms, secure element host library under pkcs11/portable/nxp folder
    - Lwip, wifi_qca support for secure_sockets in secure_sockets/portable/nxp folder
    - Flash driver support for several platforms in third_party/mcu_vendor/nxp folder
    - Generic support for aws_wifi under wifi/portable/nxp/common folder
  - Other changes:
    - Fix several build warnings, errors

Updates applied to FreeRTOS kernel up to version 10.0.0 (up to Amazon - FreeRTOS merge). New kernel related changes will be described in section above as part of AWS package.

- 9.0.0_rev3
  - New features:
    - Tickless idle mode support for Cortex-A7. Add fsl_tickless_epit.c and fsl_tickless_-generic.h in portable/IAR/ARM_CA9 folder.
    - Enabled float context saving in IAR for Cortex-A7. Added configUSE_TASK_FPU_SU-PPORT macros. Modified port.c and portmacro.h in portable/IAR/ARM_CA9 folder.

  - Other changes:
    * Transformed ARM_CM core specific tickless low power support into generic form under freertos/Source/portable/low_power_tickless/.
- 9.0.0_rev2
  - New features:
    * Enabled MCUXpresso thread aware debugging. Add freertos_tasks_c_additions.h and configINCLUDE_FREERTOS_TASK_C_ADDITIONS_H and configFRTOS_MEMORY_SCHEME macros.
- 9.0.0_rev1
  - New features:
    * Enabled -flto optimization in GCC by adding **attribute**((used)) for vTaskSwitchContext.
    * Enabled KDS Task Aware Debugger. Apply FreeRTOS patch to enable configRECORD_STACK_HIGH_ADDRESS macro. Modified files are task.c and FreeRTOS.h.
- 9.0.0_rev0
  - New features:
    * Example freertos_sem_static.
    * Static allocation support RTOS driver wrappers.
  - Other changes:
    * Tickless idle rework. Support for different timers is in separated files (fsl_tickless_systick.c, fsl_tickless_lptmr.c).
    * Removed configuration option configSYSTICK_USE_LOW_POWER_TIMER. Low power timer is now selected by linking of apropriate file fsl_tickless_lptmr.c.
    * Removed configOVERRIDE_DEFAULT_TICK_CONFIGURATION in RVDS port. Use of **attribute**((weak)) is the preferred solution. Not same as _weak!
- 8.2.3
  - New features:
    * Tickless idle mode support.
    * Added template application for Kinetis Expert (KEx) tool (template_application).
  - Other changes:
    * Folder structure reduction. Keep only Kinetis related parts.

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, 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. 2019. **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: 06/2019
Document identifier: MCUXSDKIMX8MQRN

arm