

MCUXpresso SDK Release Notes

Supporting LPCXpresso8XX



Contents

Chapter 1 Overview.....	3
Chapter 2 MCUXpresso SDK.....	4
Chapter 3 Development tools.....	5
Chapter 4 Supported development systems.....	6
Chapter 5 Release contents.....	7
Chapter 6 MCUXpresso SDK release package.....	8
Chapter 7 MISRA compliance.....	9
Chapter 8 Known issues.....	11

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,. 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 8.32.3
- MDK-Arm Microcontroller Development Kit (Keil)[®] 5.27
- Makefiles support with GCC revision 8-2018-q4-major GCC8 from Arm Embedded
- MCUXpresso IDE v11.0.0

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
LPCXpresso802, LPCXpresso804, LPCXpresso812MAX, LPCXpresso824MAX, LPCXpresso845MAX, LPC845BREAKOUT	LPC802M011JDH20 , LPC802M001JDH16, LPC802M001JDH20, LPC802M001JHI33, LPC804M101JDH24 , LPC804M101JDH20, LPC804M111JDH24, LPC804M101JHI33, LPC812M101JDH20 , LPC810M021FN8, LPC811M001JDH16, LPC812M101JDH16, LPC812M101JD20, LPC812M101JTB16, LPC824M201JHI33 , LPC834M101FHI33, LPC832M101FDH20, LPC824M201JDH20, LPC845M301JBD64 , LPC845M301JBD48, LPC845M301JHI48, LPC845M301JHI33

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
Documentation	<install_dir>/docs
Middleware	<install_dir>/middleware
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
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 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
Rule 5.1	External identifiers shall be distinct.
Rule 5.4	Macro identifiers shall be distinct.
Rule 21.1	#define and #undef shall not be used on a reserved identifier or reserved macro name.
Rule 21.2	A reserved identifier or macro name shall not be declared.
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	Typedefs that indicate size and signedness should be used in place of the basic numerical types.
Directive 4.8	If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden.
Directive 4.9	A function should be used in preference to a function-like macro where they are interchangeable.
Directive 4.13	Functions which are designed to provide operations on a resource should be called in an appropriate sequence.
Rule 1.2	Language extensions should not be used.
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 2.6	A function should not contain unused label declarations.
Rule 2.7	There should be no unused parameters in functions.
Rule 4.2	Trigraphs should not be used.
Rule 5.9	Identifiers that define objects or functions with internal linkage should be unique.
Rule 8.7	Functions and objects should not be defined with external linkage if they are referenced in only one translation unit.
Rule 8.9	An object should be defined at block scope if its identifier only appears in a single function.
Rule 8.11	When an array with external linkage is declared, its size should be explicitly specified.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

Rule 8.13	A pointer should point to a const-qualified type whenever possible.
Rule 10.5	The value of an expression should not be cast to an inappropriate essential 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 12.1	The precedence of operators within expressions should be made explicit.
Rule 12.3	The comma operator should not be used.
Rule 12.4	Evaluation of constant expressions should not lead to unsigned integer wrap-around.
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 15.4	There should be no more than one break or go to statement used to terminate any iteration statement.
Rule 17.5	The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements.
Rule 17.8	A function parameter should not be modified.
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 21.12	The exception handling features of <fenv.h> should not be used. .

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).

8.3 New Project Wizard compile failure

The following components request the user to manually select other components that they depend on to pass the compile. These components depend on several components, and the New Project Wizard (NPW) is not able to decide which one is needed by the user.

NOTE

"xxx" means core variants like `cm0plus`, `cm33`, `cm4`, `cm33_nodsp`.

Components: `Assert`, `assert_cm0plus`, `assert_xxx`, `assert_lite`, `baremetal`, `button`, `codec_i2c`, `codec_i2c_xxx`, `debug_console`, `debug_console_xxx`, `debug_console_lite`, `dialog7212`, `led`, `misc_utilities`, `panic`, `serial_manager`, `serial_manager_xxx`, `serial_manager_swo`, `serial_manager_swo_xxx`, `serial_manager_uart`, `serial_manager_uart_xxx`, `serial_manager_usb_cdc`, `serial_manager_usb_cdc_xxx`, `sgtl_adapter`, `sgtl5000`, `shell`, `shell_xxx`, `timer_manager`, `wm8904`, `wm8904_xxx`, `wm8904_adapter`, `wm8904_adapter_xxx`, `wm8960`, `wm8960_adapter`, `xip_device`.

MCUXpresso SDK Release Notes Supporting LPCXpresso8XX

Change Logs

Contents

Driver Change Log	1
USART	1
IOCON	1
CTIMER	1
CAPT	2
CRC	2
ADC	2
DAC	3
LPC_ACOMP	3
DMA	3
GPIO	4
PINT	5
SYSCON	5
I2C	6
MRT	6
INPUTMUX	6
SWM	6
SCTIMER	7
WKT	7
WWDT	7
SPI	7

Contents

Title	Page Number
CLOCK	8
POWER	8
RESET	9

1 Driver Change Log

USART

The current USART driver version is 2.1.0.

- 2.1.0
 - New feature:
 - * Added new APIs to allow users to configure the USART continuous SCLK feature in synchronous mode transfer.
- 2.0.1
 - Bug fix:
 - * Fixed the repeated reading of the STAT register issue while dealing the IRQ routine.
 - New features:
 - * Added macro gate "FSL_SDK_ENABLE_USART_DRIVER_TRANSACTIONAL_APIS" to enable/disable the transactional APIs which helps reduce the code size when no nonblocking transfer is used. Default configuration is enabled.
 - * Added control macro to enable/disable the RESET and CLOCK code in current driver.
 - * Added macro switch gate "FSL_SDK_USART_DRIVER_ENABLE_BAUDRATE_AUTO_GENERATE" to enable/disable the baud rate to generate automatically, disable this feature will help reduce the code size to a certain degree. Default configuration is enable baud rate auto generate.
 - * Added baud rate checking while initializing the USART, the baud rate calculated is not precise and causes software assertion.
 - * Added a new API to allow users to enable the CTS, which determines whether CTS is used for flow control.
- 2.0.0
 - Initial version.

IOCON

The current IOCON driver version is 2.0.0.

- 2.0.0
 - Initial version.

CTIMER

The current CTimer driver version is 2.0.2.

- 2.0.2
 - Added new API "CTIMER_GetTimerCountValue" to get the current timer count value.
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.

- Added new feature macro to update the API of CTimer driver for lpc8n04.
- 2.0.1
 - API Interface Change Added CTIMER_SetupPwmPeriod and CTIMER_UpdatePwmPulse-Period API. These two APIs can set up the right PWM with high resolution.
- 2.0.0
 - Initial version.

CAPT

The current CAPT driver version is 2.0.1.

- 2.0.1
 - Fixed the coverity out-of-bounds error caused by missing an assert sentence to avoid the return value of CAPT_GetInstance() exceeds the array bounds.
- 2.0.0
 - Initial version.

CRC

The current CRC driver version is 2.0.1.

- 2.0.1
 - Bug fix:
 - * DATA and DATALL macro definition moved from header file to source file.
- 2.0.0
 - Initial version.

ADC

The current ADC driver version is 2.3.1.

- 2.3.1
 - Bug fix:
 - * Avoided write ADC STARTUP register in ADC_Init().
 - * Fixed coverity zero divider error in ADC_DoSelfCalibration().
- 2.3.0
 - Updated "ADC_Init()" "ADC_GetChannelConversionResult()" API and "adc_resolution_t" structure to match QN9090.
 - Added "ADC_EnableTemperatureSensor" API.
- 2.2.1
 - Improvement:
 - * Added a brief delay in uSec after ADC calibration start.
- 2.2.0
 - Updated "ADC_DoSelfCalibration" API and "adc_config_t" structure to match LPC845.

- 2.1.0
 - Renamed "ADC_EnableShresholdCompareInterrupt" to "ADC_EnableThresholdCompareInterrupt".
- 2.0.0
 - Initial version.

DAC

The current DAC driver version is 2.0.1.

- 2.0.1
 - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
 - Initial version.

LPC_ACOMP

The current LPC_ACOMP driver version is 2.0.2.

- 2.0.2
 - Fixed the coverity out-of-bounds error caused by missing an assert sentence to avoid the return value of ACOMP_GetInstance() exceeds the array bounds.
- 2.0.1
 - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
 - Initial version.

DMA

The current DMA driver version is 2.3.0.

- 2.3.0
 - Bug fix:
 - * Removed DMA_HandleIRQ prototype definition from header file.
 - * Added DMA_IRQHandle prototype definition in header file.
- 2.2.5
 - Improvements:
 - * Added new API DMA_SetupChannelDescriptor to support configure wrap descriptor.
 - * Added wrap support in function DMA_SubmitChannelTransfer.
- 2.2.4
 - Bug fix:
 - * Fixed the macro DMA_CHANNEL_CFER use wrong parameter to calculate DSTINC issue.
- 2.2.3

- Bug fix:
 - * Improved DMA driver Deinit function for correct logic order.
- Improvement:
 - * Added API DMA_SubmitChannelTransferParameter to support create head descriptor directly.
 - * Added API DMA_SubmitChannelDescriptor to support ping pong transfer.
 - * Added macro DMA_ALLOCATE_HEAD_DESCRIPTOR/DMA_ALLOCATE_LINK_DESCRIPTOR to simplify DMA descriptor allocation.
- 2.2.2
 - Bug fix:
 - * Do not use software trigger when hardware trigger is enabled.
- 2.2.1
 - Bug fix:
 - * Fixed coverity issue.
- 2.2.0
 - Improvements:
 - * Changed API DMA_SetupDMADescriptor to non-static.
 - * Marked below API as deprecated. DMA_PrepareTransfer. DMA_Submit transfer.
 - * Added below new API: DMA_SetChannelConfig. DMA_PrepareChannelTransfer. DMA_InstallDescriptorMemory. DMA_SubmitChannelTransfer. DMA_SetChannelConfigValid. DMA_DoChannelSoftwareTrigger. DMA_LoadChannelTransferConfig.
- 2.0.1
 - Improvement:
 - * Added volatile for DMA descriptor member xfercfg to avoid optimization.
- 2.0.0
 - Initial version.

GPIO

The current GPIO driver version is 2.1.3.

- 2.1.4
 - Added API GPIO_PortGetInterruptStatus to retrieve interrupt status for whole port.
 - Corrected typo in header file.
- 2.1.3
 - Updated "GPIO_PinInit" API. If it has DIRCLR and DIRSET registers, use them at set 1 or clean 0.
- 2.1.2:
 - Removed deprecated APIs.
- 2.1.1:
 - API interface changes:
 - * Refined naming of API while keep all original APIs, marking them as deprecated. Original API will be removed in next release. The mainin change is update API with prefix of _PinXXX() and _PorortXXX

- 2.1.0
 - Added GPIO initialize API.
- 2.0.0
 - Initial version.

PINT

The current PINT driver version is 2.1.3.

- 2.1.3
 - Bug fix:
 - * Updated PINT_PinInterruptClrStatus to clear PINT interrupt status when the bit is asserted and check whether was triggered by edge-sensitive mode.
 - * Write 1 to IST corresponding bit will clear interrupt status only in edge-sensitive mode and will switch the active level for this pin in level-sensitive mode.
 - * Fixed MISRA c-2012 rule 10.1, rule 10.6, rule 10.7.
 - * Added FSL_FEATURE_SECPINT_NUMBER_OF_CONNECTED_OUTPUTS to distinguish IRQ relevant array definitions for SECPINT/PINT on lpc55s69 board.
 - * Fixed PINT driver c++ build error and remove index offset operation.
- 2.1.2
 - Improvement:
 - * Improved way of initialization for SECPINT/PINT in PINT_Init API.
- 2.1.1
 - Improvement:
 - * Enabled secure pint interrupt and add secure interrupt handle.
- 2.1.0
 - Added PINT_EnableCallbackByIndex/PINT_DisableCallbackByIndex APIs to enable/disable callback by index.
- 2.0.2
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.1
 - Bug fix:
 - * Updated PINT driver to clear interrupt only in Edge sensitive.
- 2.0.0
 - Initial version.

SYSCON

The current SYSCON driver version is 2.0.0.

- 2.0.0
 - Initial version.

I2C

The current I2C driver version is 2.0.3.

- 2.0.3
 - Fixed coverity issue of unchecked return value in I2C_RTOS_Transfer.
- 2.0.2
 - New features:
 - * Added macro gate "FSL_SDK_ENABLE_I2C_DRIVER_TRANSACTIONAL_APIS" to enable/disable the transactional APIs which will help reduce the code size when no non-blocking transfer is used. Default configuration is enabled.
 - * Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.1
 - Improvements:
 - * Added I2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.0
 - Initial version.

MRT

The current MRT driver version is 2.0.1.

- 2.0.1
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
 - Initial version.

INPUTMUX

The current INPUTMUX driver version is 2.0.0.

- 2.0.0
 - Initial version.

SWM

The current SWM driver version is 2.0.0.

- 2.0.0
 - Initial version.
 - The API SWM_SetFixedMovablePinSelect() is targeted at the device that has PINASSIGNFIXED0 register, such as LPC804.

SCTIMER

The current SCTimer driver version is 2.1.0.

- 2.1.0
 - Bug fixes:
 - * Fixed issue where SCT application level Interrupt handler function is occupied by SCT driver.
 - * Fixed issue where wrong value for INSYNC field inside SCTIMER_Init function.
 - * Fixed issue to change Default value for INSYNC field inside SCTIMER_GetDefault-Config.
- 2.0.1
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
 - Initial version.

WKT

The current WKT driver version is 2.0.1.

- 2.0.1
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
 - Initial version.

WWDT

The current WWDT driver version is 2.1.2.

- 2.1.2
 - Updated "WWDT_ClearStatusFlags" and "WWDT_GetStatusFlags" API to match QN9090. WDTOF is not set in case of WD reset. Get info from PMC instead.
- 2.1.1
 - Added new feature definition macro for devices have no LCOK control bit in MOD register.
 - Implemented delay/retry in WWDT driver
- 2.1.0
 - Added new parameter in configuration when initializing WWDT module, this parameter allows the user to deliver the WWDT clock frequency, and this parameter must be set.
- 2.0.0
 - Initial version.

SPI

The current SPI driver version is 2.0.1.

- 2.0.1
 - Bug fix:
 - * Added wait mechanism in SPI_MasterTransferBlocking() API, which checks if master SPI becomes IDLE when the EOT bit is set before returning. This confirms all data has been sent out by SPI master.
 - * Fixed the bug that cannot set the EOT bit when only one frame is sent in polling mode and interrupt transfer mode.
 - New features:
 - * Added macro gate "FSL_SDK_ENABLE_SPI_DRIVER_TRANSACTIONAL_APIS" to enable/disable the transactional APIs which helps reduce the code size when no nonblocking transfer is used. Default configuration is enabled.
 - * Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
 - Initial version.

CLOCK

The current CLOCK driver version is 2.1.0.

- 2.1.0
 - New feature
 - * Adding new API CLOCK_DelayAtLeastUs() to implement a delay function which allows users to set delay in unit of microsecond.
- 2.0.3
 - add api to get uart clock frequency.
 - add api to set fractional multiplier value.
- 2.0.2
 - some minor fixes.
- 2.0.0
 - initial version.

POWER

The current POWER driver version is 2.0.2.

- 2.0.2
 - add the Enable/DisableDeepSleepIRQ() to enable/disable pin wake up.
- 2.0.1
 - Update power driver to support PMU.
- 2.0.0
 - initial version.

RESET

The current RESET driver version is 2.0.1.

- 2.0.1
 - Update component full_name to "Reset Driver".
- 2.0.0
 - initial version.

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.

© 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: MCUXSDKLPC8XXRN

