NXP Semiconductors
Release Notes

Document Number: MCUXSDKLPC546XRN
Rev. 1, 05/2017

MCUXpresso SDK Release Notes Supporting LPCXpresso54608, LPCXpresso54618, and LPCXpresso54S618

Contents

1 Overview

The MCUXpresso Software Development Kit (SDK) is a collection of software enablement for Microcontrollers that includes peripheral drivers, high-level stacks including USB and lwIP, integration with WolfSSL and mbed TLS cryptography libraries, other middleware packages, such as multicore support and FatFs, 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 the latest version of this and other MCUXpresso SDK documents, see the MCUXpresso SDK homepage MCUXpresso-SDK: Software Development Kit.

2 Development Tools

The SDK 2.0.0 was compiled and tested with these development tools:

- IAR Embedded Workbench for ARM version 7.80.4
- MDK-ARM Microcontroller Development Kit (Keil)® 5.21a
- MCUXpresso IDE v10.0.0

3 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
LPCXpresso54608, LPCXpresso54S618, LPCXpresso54618	LPC54605J256ET180, LPC54605J512ET180, LPC54606J256ET180, LPC54606J512BD208, LPC54607J256ET180, LPC54607J512ET180, LPC54607J256BD208, **LPC54608J512ET180**, LPC54608J512BD208, **LPC54S618J512ET180**, LPC54S618J512BD208, LPC54616J256ET180, LPC54616J512BD208, **LPC54618J512ET180**, LPC54618J512BD208

4 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
USB demo applications	<install_dir>/boards/<board_name>/usb_examples
Driver examples	<install_dir>/boards/<board_name>/driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Documentation	<install_dir>/docs
CMSIS Driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
emWin examples	<install_dir>/boards/<board_name>/emwin_examples
USB Documentation	<install_dir>/docs/usb
lwIP Documentation	<install_dir>/docs/lwip
Middleware	<install_dir>/middleware
lwIP stack	<install_dir>/middleware/lwip_<version>
FatFs stack	<install_dir>/middleware/fatfs_<version>
SDMMC card driver	<install_dir>/middleware/sdmmc_<version>
USB stack	<install_dir>/middleware/usb_<version>
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source	<install_dir>/CMSIS
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers

Table continues on the next page...

Table 2. Release contents (continued)

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

5 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.

5.1 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 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 CMSIS-compliant startup that efficiently transfers the code execution to the main() function.

5.1.1 Board support

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

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

The middleware folders each contain examples demonstrating the use of the included source.

5.2 Middleware

5.2.1 USB stack

See the *MCUXpresso SDK USB Stack User's Guide* (document USBSUG) for more information.

5.2.1.1 Peripheral devices tested with the USB Host stack

This table provides a list of USB devices tested with the USB Host stack.

Table 3. Peripheral devices

Device type	Device
USB HUB	BELKIN F5U233
	BELKIN F5U304
	BELKIN F5U307
	BELKIN F4U040
	UNITEK Y-2151
	Z-TEK ZK032A
	HYUNDAI HY-HB608
USB flash drive	ADATA C008 32 GB
	ADATA S102 8 G
	ADATA S102 16 G
	Verbatim STORE N GO USB Device 8 G
	Kingston DataTraveler DT101 G2
	SanDisk Cruzer Blade 8 GB
	Unisplendour 1 G
	Imation 2 GB
	V-mux 2 GB
	Sanmina-SCI 128 M
	Corporate Express 1 G
	TOSHIBA THUHYBS-008G 8 G
	Transcend JF700 8 G
	Netac U903 16 G
	SSK SFD205 8 GB
	Rex 4 GB
	SAMSUNG USB3.0 16GB
USB card reader/adapter	SSK TF adapter
	Kawau Multi Card Reader
	Kawau TF adapter
	Kawau SDHC card
USB Mouse	DELL MS111-P

Table continues on the next page...

Table 3. Peripheral devices (continued)

	DELL M066U0A DELL MUAVDEL8 TARGUS AMU76AP DELL MD56U0 DELL MS111-T RAPOO M110
USB Keyboard	DELL SK8135 DELL SK8115

5.2.2 TCP/IP stack

The lwIP TCP/IP stack is pre-integrated with SDK and runs on top of the SDK Ethernet driver with Ethernet-capable devices/boards. For details, see the *lwIP TCPIP Stack and MCUXpresso SDK Integration User's Guide* (document MCUXSDKLWIPUG).

5.2.3 RTOS

The MCUXpresso SDK is pre-integrated with FreeRTOS OS.

5.2.4 CMSIS

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

5.2.5 emWin

The MCUXpresso SDK is pre-integrated with emWin.

6 MISRA Compliance

All MCUXpresso SDK drivers and USB stack comply to MISRA 2004 rules with the following exceptions:

Exception Rules	Description
1.1	All code shall conform to ISO 9899:1990 Programming languages - C, amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC
2.4	Sections of code should not be commented out.
5.1	Identifiers (internal and external) shall not rely on the significance of more than 31 characters.
6.3	typedefs that indicate size and signedness should be used in place of the basic types.
6.4	Bitfields shall only be defined to be of type unsigned int or signed int.
8.1	Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call.
8.5	There shall be no definitions of objects or functions in a header file.
8.1	All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.
8.12	When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialization.
10.1	The value of an expression of integer type shall not be implicitly converted to a different underlying type if: a. it is not a conversion to a wider integer type of the same signedness, or b. the expression is complex, or c. the expression is not constant and is a function argument, or d. the expression is not constant and is a return expression.
10.3	The value of a complex expression of integer type shall only be cast to a type that is not wider and of the same signedness as the underlying type of the expression.
11.3	A cast should not be performed between a pointer type and an integral type.
11.4	A cast should not be performed between a pointer to object type and a different pointer to object type.
11.5	A cast shall not be performed that removes any const or volatile qualification from the type addressed by a pointer.
12.2	The value of an expression shall be the same under any order of evaluation that the standard permits.
12.4	The right-hand operand of a logical && or | | operator shall not contain side effects.
12.6	The operands of logical operators (&&, | |, and !) should be effectively boolean. Expressions that are effectively boolean should not be used as operands to operators other than (&&, | |, !, =, ==, !=, and ?:).
12.13	The increment (++) and decrement (--) operators should not be mixed with other operators in an expression.
14.3	Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment, provided that the first character following the null statement is a whitespace character.
14.5	The continue statement shall not be used.
14.7	A function shall have a single point of exit at the end of the function.
16.1	Functions shall not be defined with a variable number of arguments.
17.4	Array indexing shall be the only allowed form of pointer arithmetic.
18.4	Unions shall not be used.
19.1	#include statements in a file should only be preceded by other preprocessor directives or comments.
19.1	In the definition of a function-like macro, each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##.
20.4	Dynamic heap memory allocation shall not be used.
20.9	The input/output library <stdio.h> shall not be used in production code.

Figure 1. MISRA exceptions

7 Known Issues

7.1 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.

7.2 USB HUB Power supply

The external power supply of the USB HUB must be provided before it can be used. The development board power is not enough to supply multi-level USB HUBs and connected devices. Therefore, the external USB HUB that is connected to the development board should have its own power supply.

7.3 SDIF issue

If the user uses this module to access the SD card/MMC card, sd_data[4-7] must be enable and configured as input, and pulled up.

7.4 SDIF issue

If the user uses this module to access the SD card/MMC card, sd_data[4-7] must be enabled and configured as input and pull up. It is better to use P4_17 to P5_10, otherwise 4-bit mode does not work.

8 Change Log - Peripheral drivers

ADC

The current ADC driver version is 2.0.0

- 2.0.0
  - Initial version

CRC

The current CRC driver version is 2.0.1

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

CTIMER

The current CTIMER driver version is 2.0.0

- 2.0.0
  - Initial version

DMA

The current DMA driver version is 2.0.1

- 2.0.0
  - Initial version
- 2.0.1
  - Bug fix:
    - Fixed the DMA driver build fail by adding parenthesis. The issue occurs because of the MISRA C 2004 rule 12.5

DMIC

The current DMIC driver version is 2.0.0

- 2.0.0
  - Initial version

EEPROM

The current EEPROM driver version is 2.0.0

- 2.0.0
  - Initial version

EMC

The current EMC driver version is 2.0.0

- 2.0.0
  - Initial version

ENET

The current ENET driver version is 2.1.1

- 2.0.0
  - Initial version
- 2.0.1
  - Bug fix:
    - Used the direct transmit busy check during data transmission
  - Changes:
    - Updated the IRQ handler workflow
    - Changed the TX/RX interrupt macro from kENET_RxByteInterrupt to kENET_RxBufferInterrupt and from kENET_TxByteInterrupt to kENET_TxBufferInterrupt
    - Deleted unnecessary parameters in the ENET handler
- 2.1.1
  - Add the extended MDIO IEEE802.3 Clause 45 MDIO format SMI command APIs
  - Add the extended interrupt coalescing feature
  - Combine all storage operations in the ENET_Init to ENET_SetHandler API

- 2.3.0
  - New features:
    - Add support for device with LP flash (K3S/G)
    - Add flash prefetch speculation APIs
  - Improvement
    - Refine flash_cache_clear function
    - Reorganize the member of flash_config_t struct
- 2.3.1
  - Bug fix
    - Unified Flash IFR design from K3
    - New encoding rule for K3 flash size

FlashIAP

The current CTIMER driver version is 2.0.0

- 2.0.0
  - Initial version

FMC

The current FMC driver version is 2.0.0

- 2.0.0
  - Initial version

FMEAS

The current FMEAS driver version is 2.0.0

- 2.0.0
  - Initial version

GINT

The current GINT driver version is 2.0.0

- 2.0.0
  - Initial version

GPIO

The current GPIO driver version is 2.1.1

- 2.1.0
  - API Interface Change:
    - Added "pins" or "pin" to some API names
    - Renamed the "GPIO_PinConfigure" to "GPIO_PinInit"
- 2.1.1
  - API Interface Change
    - Added API for the check attribute bytes

INPUTMUX

The current INPUTMUX driver version is 2.0.0

- 2.0.0
  - Initial version

IOCON

The current IOCON driver version is 2.0.0

- 2.0.0
  - Initial version

I2C

The current I2C driver version is 2.0.3

- 2.0.1
  - New features
    - Added a double buffer enable configuration for SoCs which have the DFEN bit in S2 register
    - Added the flexible transmit/receive buffer size support in I2C_SlaveHandleIRQ
    - Added the start flag clear address match and release bus operation in I2C_SlaveWrite/ReadBlocking API
  - Bug fix:
    - Updated the kI2C_SlaveRepeatedStartEvent to kI2C_SlaveStartEvent
- 2.0.2
  - Bug Fix:
    - Fixed the issue that occurs in master receive and slave transmit mode with no stop flag and master can't start a next transfer because it can't send out restart signal
    - Fixed a data transfer out of order issue which occurs because of a memory barrier
  - New Features:
    - Added an address nak event for the master.
    - Added a general call event for the slave.
- 2.0.3
  - Bug fix
    - Remove enableHighDrive member in the master/slave configuration structure because the operation to HDRS bit is useless. The user needs to use DSE bit in port register to configure the high drive capability.
    - Add reset registers operation in I2C_MasterInit and I2C_SlaveInit APIs, and fix the issue that I2C could not switch between master and slave mode.
    - Improve slave IRQ handler to handle the corner case that stop flag and address match flag come synchronously.

I2S

The current I2S driver version is 2.0.0

- 2.0.0
  - Initial version

LCDC

The current IRTC driver version is 2.0.0

- 2.0.0
  - Initial version

- 2.1.1
  - Bug fix:
    - Disable the auto stop feature in the EDMA driver. Previously the autostop feature is enabled at transfer when transfer with stop flag. If the previous transfer is without stop flag, then when starting a new transfer with stop flag, because the auto stop feature is enabled, the stop flag sends before starting the new transfer and the start flag can not successfully sent, so the transfer cannot start.
    - Change default slave configuration with address stall false

MCAN

The current MCAN driver version is 2.0.1

- 2.0.0
  - Intiial version
- 2.0.1
  - LPC54608 chip does not support the FD feature, so add the feature macro for it.

MRT

The current MRT driver version is 2.0.0

- 2.0.0
  - Initial version

OTP

The current OTP driver version is 2.0.0

- 2.0.0
  - Initial version

PINT

The current PINT driver version is 2.0.0

- 2.0.0
  - Initial version

RIT

The current RIT driver version is 2.0.0

- 2.0.0
  - Initial version

RNG

The current RNG driver version is 2.0.0

- 2.0.0

        - Initial version

RTC

The current RTC driver version is 2.0.0

- 2.0.0
    - Initial version

SCTIMER

The current SCTIMER driver version is 2.0.0

- 2.0.0
    - Initial version

SDIF

The current SDIF driver version is 2.0.0

- 2.0.0
    - Initial version

SPI

The current SPI driver version is 2.0.1

- 2.0.0
    - Initial version
- 2.0.1
    - Bug fix
        - Fixed the SPI_Enable function parameter error
        - Set the s_dummy variable as a static variable in fsl_spi_dma.c
    - Optimization
        - Optimizes the code size when not using the transactional API
        - Improved performance in a polling method

SPI Flash interface

The current SPI Flash interface driver version is 2.0.0

- 2.0.0
    - Initial version

USART

The current USART driver version is 2.0.0

- 2.0.0
    - Initial version

UTICK

The current UTICK driver version is 2.0.0

- 2.0.0
    - Initial version

WWDT

The current WWDT driver version is 2.0.0

- 2.0.0
  - Initial version

9 Change Log - Middleware

emWin library

Currently supported version is 5.38a

FatFs

The current FatFs driver version is R0.12b

- R0.12b_rev0
- R0.11a
  - Added glue functions for low level drivers (SDHC, SDSPI, RAM, and MMC) and modified the diskio.c file
  - Added RTOS wrappers to make FatFs thread-safe. Modified the syscall.c file
  - Renamed ffconf.h file to ffconf_template.h file. Each application should contain its own ffconf.h file
  - Conditional compilation of physical disk interfaces in diskio.c

lwIP

The current lwIP version is based on the lwIP 2.0.0 (2016-11-10, SHA-1: 216bf89491815029aa15463a18744afa04df58fe)

- 2.0.0_rev3
  - New Features:
    - Ported lwIP 2.0.0 (2016-11-10, SHA-1: 216bf89491815029aa15463a18744afa04df58fe) to KSDK 2.0.0.
- 2.0.0_rev2
  - New Features:
    - Ported lwIP 2.0.0 RC2 (2016-08-08, SHA-1: b1dfd00f9233d124514a36a8c8606990016f2ad4) to KSDK 2.0.0
- 2.0.0_rev1
  - New Features:
    - Ported lwIP 2.0.0 RC0 (2016-05-26) to KSDK 2.0.0
    - Changed lwIP bare metal examples to use a poll-driven approach instead of an interrupt-driven one
- 1.4.1_rev1
  - New features:
    - Ported the lwIP 1.4.1 to the KSDK 2.0.0
- 1.4.1_rev2
  - New features:
    - Enabled critical sections in lwIP
  - Bug fix
    - Fixed the default lwIP packet buffer size to accept a maximum size frame from the ENET driver
    - Fixed a possible drop of multiframe packets during transmission

SDMMC

The current SDMMC driver version is 2.1.2

- 2.1.0
  - Bug fix:

        - Changed the callback mechanism when sending a command
        - Fixed the performance low issue when transferring data
    - Changes:
        - Changed the name of error codes returned by an internal function
        - Merged all host-related attributes into one structure
        - Optimized the function to set a maximum data bus width for the MMC card
- 2.1.1
    - Bug fix:
        - Fixed the block range boundary error when transferring data to the MMC card
        - Fixed the bit mask error in the SD card when switching to a high-speed function
    - Changes:
        - Added an error code to indicate that SDHC ADMA1 transfer type is not supported
        - Optimized the SD card initialization function

- 2.1.2
    - New feature
        - Add fsl_host.h to provide prototype to adapt different controller IPs(SDHC/SDIF)
        - Add adaptor code in sdmmc/port folder to adapt different host controller IPs with different transfer modes(int/polling/freertos). Application inlude different adpator code to make application simplier
        - Adaptor code provides HOST_Init/HOST_Deinit/CardInsertDetect APIs to do host controller initialize and transfer function configuration. SDMMC card stack uses adaptor code inside stack to wait card insert and configure host when calling card init APIs (SD_Init/MMC_Init/SDIO_Init)
        - So this change requires user to include host adaptor code into application. If not, link errors for cannot find the definition of HOST_Init/HOST_Deinit/CardInsertDetect will appear
    - New feature
        - Improve SDMMC to support SD v3.0 and EMMC v5.0
    - Bug fix:
        - Fix wrong comparison between count and length in MMC_ReadBlocks/MMC_WriteBlocks

USB stack

The current USB stack version is 1.6.3

- 1.0.0
    - New features:
        - Supported roles
            - Device
            - Host
        - Supported controllers
            - KHCI (full-speed)
            - EHCI (high-speed)
        - Supported classes
            - AUDIO
            - CCID
            - CDC
            - HID
            - MSC
            - PHDC
            - VIDEO
        - Examples
            - usb_device_audio_generator
            - usb_device_audio_speaker
            - usb_device_ccid_smart_card
            - usb_device_cdc_vcom
            - usb_device_cdc_vnic
            - usb_device_composite_cdc_msc

        - usb_device_composite_hid_audio
        - usb_device_composite_hid_mouse_hid_keyboard
        - vusb_device_hid_generic
        - usb_device_hid_mouse
        - usb_device_msc_ramdisk
        - usb_device_msc_sdcard
        - usb_device_phdc_weighscale
        - usb_device_video_flexio_ov7670
        - usb_device_video_virtual_camera
        - usb_host_audio_speaker
        - usb_host_cdc
        - usb_host_hid_generic
        - usb_host_hid_mouse
        - usb_host_hid_mouse_keyboard
        - usb_host_msd_command
        - usb_host_msd_fatfs
        - usb_host_phdc_manager
        - usb_keyboard2mouse
        - usb_pin_detect_hid_mouse
- 1.0.1
    - Bug fix
        - Improved the device audio speaker efficiency by changing the transfer mode from interrupt to DMA to eliminate the periodic noise
- 1.1.0
    - Bug fix:
        - Fixed issues in the USB certification
        - Updated the VID and Manufacturer string to NXP Semiconductors
    - New features:
        - Supported classes
            - Printer
        - Examples
            - usb_device_composite_cdc_msc_sdcard
            - usb_device_printer_virtual_plain_text
            - usb_host_printer_plain_text
    - Changes:
        - Renamed example usb_device_composite_cdc_msc to usb_device_composite_cdc_msc_ramdisk
- 1.2.0
    - New features:
        - Supported controllers
            - LPC IP3511 (Full speed, device mode)
- 1.3.0
    - New features:
        - Supported roles
            - OTG
        - Supported classes
            - CDC RNDIS
        - Examples
            - usb_otg_hid_mouse
            - usb_device_cdc_vnic
            - usb_suspend_resume_device_hid_mouse
            - usb_suspend_resume_host_hid_mouse
- 1.4.0
    - New features:
        - Examples
            - usb_device_hid_mouse/freertos_static

        - usb_suspend_resume_device_hid_mouse_lite
- 1.5.0
    - New features:
        - Supported controllers
            - OHCI (Full Speed, Host mode)
            - IP3516 (High Speed, Host mode)
            - IP3511 (High Speed, Device mode)
- 1.6.0
    - New features:
        - Supported Device Charger Detect feature on usb_device_hid_mouse
- 1.6.1
    - New features:
        - Change the struct variable address method for device_video_virtual_camera and host_phdc_manager
- 1.6.2
    - New features:
        - Multi instance support
- 1.6.3
    - Bug fix:
        - -IP3511_HS driver control transfer sequence issue, enbale 3511 ip cv test

10 Change Log - RTOS

FreeRTOS OS

The current version is FreeRTOS OS 9.0.0. The original package is available at freertos.org.

- 9.0.0_rev2
    - New features:
        - Enabled MCUXpresso thread aware debugging. Added freertos_tasks_c_additions.h and configINCLUDE_FREERTOS_TASK_C_ADDITIONS_H and configFRTOS_MEMORY_SCHEME macros
- 9.0.0_rev1
    - New features:
        - Enable -flto optimization in GCC by adding **attribute**((used)) for vTaskSwitchContext
        - Enable 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)
        - Remove configuration option configSYSTICK_USE_LOW_POWER_TIMER. Low power timer is now selected by linking of appropriate file fsl_tickless_lptmr.c
        - Remove configOVERRIDE_DEFAULT_TICK_CONFIGURATION in RVDS port. Use of **attribute**((weak)) is preferred solution. Not same as _weak
- 8.2.3
    - New features:

- Added tickless idle mode support
- Added a template application for Kinetis Expert (KEx) tool (template_application)

- Changes:
  - Reduced the folder structure to keep only Kinetis-related information

11 Revision History

This table summarizes revisions to this document.

Table 4. Revision history

Revision number	Date	Substantive changes
0	03/2017	Initial release
1	05/2017	Initial release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number MCUXSDKLPC546XRN
Revision 1, 05/2017