

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](#).

1	Overview.....	1
2	MCUXpresso SDK.....	1
3	Development Tools.....	2
4	Supported Development Systems.....	2
5	Release Contents.....	2
6	MCUXpresso SDK Release Package.....	3
7	MISRA Compliance.....	5
8	Known Issues.....	6
9	Change Log - Peripheral drivers.....	7
10	Change Log - Middleware.....	13
11	Change Log - RTOS.....	17
12	Revision History.....	18

2 MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpressoSDK is the evolution of Kinetis SDK v2.3.0, 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, a new Eclipse-based toolchain that works with all MCUXpresso



Development Tools

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 FSL 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.

3 Development Tools

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

- Kinetis Design Studio IDE v3.2
- IAR Embedded Workbench for Arm version 8.11.3
- MDK-Arm Microcontroller Development Kit (Keil)[®] 5.23
- Makefiles support with GCC revision v6-2017-q2 from Arm Embedded
- MCUXpresso IDE v10.1.0

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

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

Table continues on the next page...

Table 2. Release contents (continued)

USB demo applications	<install_dir>/boards/<board_name>/usb_examples
Driver examples	<install_dir>/boards/<board_name>/driver_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
Middleware	<install_dir>/middleware
DMA manager	<install_dir>/middleware/dma_manager
SDMMC card driver	<install_dir>/middleware/sdmmc
USB stack	<install_dir>/middleware/usb
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
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
Tools	<install_dir>/tools

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 CMSIS-compliant 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 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.

6.2 Middleware

6.2.1 USB stack

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

6.2.1.1 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

Table continues on the next page...

Table 3. Peripheral devices (continued)

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

6.2.2 CMSIS

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

6.2.3 emWin

The MCUXpresso SDK is pre-integrated with emWin.

7 MISRA Compliance

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

Known Issues

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.
	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.
10.1	
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

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 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.

8.3 USBFS controller issue

Because of the USBFS controller design issues, the USB host suspend/resume demos (usb_suspend_resume_host_hid_mouse) of the full speed controller do not support the low speed device directly.

8.4 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.

8.5 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.

8.6 USB High-Speed Interrupt Endpoint issue

If the user want to use a high-speed interrupt endpoint, the maximum packet size should be 512 bytes.

9 Change Log - Peripheral drivers

ADC

The current ADC driver version is 2.1.0

- 2.0.0
 - Initial version
- 2.1.0
 - Renames "ADC_EnableShresholdCompareInterrupt" to "ADC_EnableThresholdCompareInterrupt"

CRC

The current CRC driver version is 2.0.1

- 2.0.0
 - Initial version
- 2.0.1
 - Fix KPSDK-13362. MDK compiler issue when writing to `WR_DATA` with `-O3` optimize for time.

CTIMER

The current CTIMER driver version is 2.0.0

- 2.0.0
 - Initial version

DMA

The current DMA driver version is 2.0.0

- 2.0.0
 - Initial version

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 2.0.0

- 2.0.0
 - Initial version

FlashIAP

The current CTIMER driver version is 2.0.0

- 2.0.0
 - Initial version

FLEXCOMM

The current FLEXCOMM 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 posted for LPCOpen.

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

- 2.0.0
 - Initial version
- 2.1.0
 - Add GPIO initialize API.
- 2.1.1
 - API Interface Change
 - Refine naming of API while keeping all original APIs with marking them as deprecated. Original API will be removed in next release. The main change is updated API with prefix of `_PinXXX()` and `_PorortXXX`

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

Change Log - Peripheral drivers

- 2.0.0
 - Initial version

I2C

The current I2C driver version is 2.0.1

- 2.0.0
 - Initial version
- 2.0.1
 - Improvement
 - Add I2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.

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

MCAN

The current MCAN driver version is 2.0.1

- 2.0.0
 - Initial 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 2.0.4

- 2.0.0
 - Initial version
- 2.0.1
 - New features
 - Improve interrupt transfer callback
 - Bug fix
 - Add assert to limit the SDIF source clk below 52 MHZ
- 2.0.2

Change Log - Peripheral drivers

- Bug fix
 - Fix the status or the flag cannot be cleared entirely after transfer is complete.
- 2.0.3
 - Improve Clock Delay macro to allow users to redefine and remove useless delays for the clock below 25 MHZ.
- 2.0.4
 - Add cardInserted/cardRemoved callback function.
 - Add host base address/user data parameter for all callback functions.

SPI

The current SPI driver version is 2.0.1

- 2.0.0
 - Initial version
- 2.0.1
 - Change the data buffer from uint32_t to uint8_t which matches the real applications for the SPI DMA driver.
 - Added dummy data setup API to allow users to configure the dummy data to be transferred.
 - Added new APIs for half-duplex transfer function. Users can send and receive data by one API in polling/interrupt/DMA way, and users can choose to either transmit first or receive first. Additionally, the PCS pin can be configured as assert status in transmission (between transmit and receive) by setting the isPcsAssertInTransfer to true.

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

10 Change Log - Middleware

emWin library

Currently supported version is 5.38a

FatFs

The current FatFs driver version is R0.12c

- 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
 - Include ffconf.h into diskio.c to enable selection of physical disk from ffconf.h by macro definition
 - Conditional compilation of physical disk interfaces in diskio.c
- R0.12b_rev0
 - Upgrade to version 0.12b
- R0.12c_rev0
 - Upgrade to version 0.12c and apply patches ff_12c_p1.diff and ff_12c_p2.diff

lwIP

The current lwIP version is based on the 2.0.2 (2017-03-13, SHA-1: c0862d60746e2d1ceae69af4c6f24e469570ecef)

- 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
- 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
- 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_rev3
 - New Features:

Change Log - Middleware

- Ported lwIP 2.0.0 (2016-11-10, SHA-1: 216bf89491815029aa15463a18744afa04df58fe) to KSDK 2.0.0
- 2.0.2_rev1
 - New Features:
 - Ported lwIP 2.0.2 (2017-03-13, SHA-1: c0862d60746e2d1ceae69af4c6f24e469570ecef) to KSDK 2.0.0

SDMMC

The current SDMMC driver version is 2.1.4

- 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 include different adaptor code to make application simpler
 - 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
- 2.1.3
 - Bug fix:
 - Non high-speed SD Card init fail at switch to high speed
 - Miscellaneous changes:
 - Optimized tuning/mmc switch voltage/mmc select power class/mmc select timing function
 - Added strobe dll for mmc HS400 mode
 - Added Delay for SD Card power up
- 2.1.4
 - Miscellaneous changes:
 - Added Host reset function for card re-initialization
 - Added Go_Idle function for SDIO card
 - Added Host_ErrorRecovery function for host error recovery procedure
 - Added cache maintain operation
 - Added HOST_CARD_INSERT_CD_LEVEL to improve compatibility
 - Bug fix:
 - Fixed issue where card cannot detect dynamically

USB stack

The current USB stack version is 1.8.0

- 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, enable 3511 ip cv test
- 1.7.0
 - New features:
 - USB Type-C PD stack support.
 - Examples:
 - usb_pd
 - usb_pd_battery
 - usb_pd_source_charger
- 1.8.0
 - New features:
 - Examples:
 - usb_device_composite_cdc_vcom_cdc_vcom
 - usb_device_audio_speaker_lpc
 - usb_device_composite_hid_audio_unified
 - usb_device_composite_hid_audio_unified_lpc
 - usb_pd_sink_battery
 - Change usb_pd_battery to usb_pd_charger_battery
 - Bug fix:
 - Code clean up, removed some irrelevant code.

11 Change Log - RTOS

FreeRTOS OS

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

- 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
- 9.0.0_rev0
 - New features:

Revision History

- 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**
- 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_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_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.
 - Enable float context saving in IAR for Cortex-A7. Add configUSE_TASK_FPU_SUPPORT macros. Modify port.c and portmacro.h in portable/IAR/ARM_CA9 folder.
 - Other changes:
 - Transform ARM_CM core specific tickless low power support into generic form under FreeRTOS.

12 Revision History

This table summarizes revisions to this document.

Table 4. Revision history

Revision number	Date	Substantive changes
0	03/2017	MCUXpresso SDK initial release
1	05/2017	Adding LPCXpresso54618 board
2	11/2017	Updated for MCUXpresso SDK v2.3.0 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.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, Freescale, the Freescale logo, Kinetis, Tower are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, Cortex, Keil, and μ Vision are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2017 NXP B.V.

Document Number MCUXSDKLPC546XRN
Revision 2, 11/2017

