NXP Semiconductors
User's Guide

Document Number: MCUXSDKLPC55XXRN
Rev. 0, 06/2019

MCUXpresso SDK Release Notes Supporting LPC55XX

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 FatFs, USB, mbed TLS cryptography libraries, other middleware packages, multicore support and integrated RTOS support for FreeRTOS^{TM} 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 8.32.3
- 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
LPCXpresso55S69	**LPC55S69JBD100,** LPC55S69JET98, LPC55S66JBD100, LPC55S66JET98

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
Qualcomm WiFi	<install_dir>/middleware/wifi_qca
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
Cortex Microcontroller Software Interface Standard (CMSIS) driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
FatFS examples	<install_dir>/boards/<board_name>/fatfs_examples
emWin examples	<install_dir>/boards/<board_name>/emwin_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Multicore examples	<install_dir>/boards/<board_name>/multicore_examples
Qualcomm WiFi stack examples	<install_dir>/boards/<board_name>/wifi_qca_examples
mbed TLS examples	<install_dir>/boards/<board_name>/mbedtls_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
Middleware	<install_dir>/middleware
mbed TLS	<install_dir>/middleware/mbedtls
FatFS stack	<install_dir>/middleware/fatfs
Multicore stack	<install_dir>/middleware/multicore
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>
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 USB stack

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

6.2.1.1 Peripheral devices tested with USB Host stack

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

Table 3. Peripheral devices

Device type	Device

Table continues on the next page...

Table 3. Peripheral devices (continued)

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

Table continues on the next page...

Table 3. Peripheral devices (continued)

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 File system

The FatFs file system is integrated with MCUXpresso SDK and can be used to access either the SD card or the USB memory stick when the SD card driver or the USB Mass Storage Device class implementation is used.

6.2.3 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.4 CMSIS

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

6.2.5 emWin

The MCUXpresso SDK is pre-integrated with the SEGGER emWin GUIBuilder.

6.2.6 Other middleware

Optional middleware packages can be included in the release based on the user selection. See *<install_dir>/SW-Content-Register.txt* for a list of components and associated licenses.

Chapter 7
MISRA compliance

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

Table 4. 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 identifieror 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 4. 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 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.

8.3 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 LPC55SXX

Change Logs

Contents

Contents

Title Page Number

Contents

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.1.0.

- 2.1.0
  - New feature
    - Adding new API CLOCK_DelayAtLeastUs() implemented by DWT to allow users set delay in unit of microsecond.
- 2.0.4
  - Bug Fix:
    - Fix C++ build errors in CLOCK_GetClockAttachId() and CLOCK_AttachClk().
- 2.0.3
  - Bug Fix:
    - Fix attach incorrect attach_id.
- 2.0.2
  - New Feature:
    - add get actual clock attach id api to allow users to obtain the actual clock source in target register.
  - Bug Fix:
    - The attach clock and get actual clock attach id apis should check combination of two clock source.
  - Optimization:
    - Make the judgement statments more clear.
    - Strengthen the compatibility of clock attatch id.
    - Remove some unmeaningful definitions and add some useful ones to enhance readability.
- 2.0.1
  - some minor fixes.
- 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.

COMMON

The current COMMON driver version is 2.1.0.

- 2.1.0
  - New features:
    - Added IRQ operation for XCC toolchain.
    - Added group IDs for newly supported drivers.
- 2.0.2
  - MISRA C-2012 issue fixed.
    - Fixed rule contain: rule-10.4.
- 2.0.1
  - Removed the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
  - Added new feature macro switch "FSL_FEATURE_HAS_NO_NONCACHEABLE_SECTI-ON" for specific SoCs which have no noncacheable sections, that helps avoid an unnecessary complex in link file and the startup file.
  - Updated the align(x) to **attribute**(aligned(x)) to support MDK v6 armclang compiler.
- 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_UpdatePwmPulsePeriod API. These two APIs can set up the right PWM with high resolution.
- 2.0.0
  - Initial version.

CMP

The current CMP driver version is 2.0.0.

- 2.0.0
  - Initial version.

FLEXCOMM

The current FLEXCOMM driver version is 2.0.1.

- 2.0.1
  - Added more IRQHandler code in drivers to adapt new devices.
- 2.0.0
  - Initial version.

I2C

The current I2C driver version is 2.0.6.

- 2.0.6
  - Fixed coverity issue of unchecked return value in I2C_RTOS_Transfer.
- 2.0.5
  - Bug fix:
    - Fixed build warning caused by uninitialized variable.
- 2.0.4
  - Updated the "I2C_MasterSetBaudRate" API to support baudrate configuration for feature Q-N9090.
- 2.0.4
  - Updated the I2C_WATI_TIMEOUT macro to unified name I2C_RETRY_TIMES.
- 2.0.3
  - Unified component full name to FLEXCOMM I2C(DMA/FREERTOS) driver.
- 2.0.2
  - Improvements: In slave IRQ:
    1. Changed slave receive process to first set the I2C_SLVCTL_SLVCONTINUE_MASK to acknowledge the received data, then do data receive.
    2. Improved slave transmit process to set the I2C_SLVCTL_SLVCONTINUE_MASK immediately after write the data.
- 2.0.1
  - Improvements:
    - Added I2C_WATI_TIMEOUT macro to allow user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.0
  - Initial version.

I2S

The current I2S driver version is 2.1.0.

- 2.1.0
  - Improvements:
    - Added feature for the FLEXCOMM which supports I2S and has interconnection with DMIC.
    - Used feature to control PDMDATA instead of I2S_CFG1_PDMDATA.
    - Added member bytesPerFrame in i2s_dma_handle_t, used for DMA transfer width con-

figure instead of use sizeof(uint32_t) hardcode.
    - Used the macro provide by DMA driver to define the I2S DMA descriptor.
  - Bug fix:
    - Fixed I2S DMA driver to always generate duplicate callback.
- 2.0.3
  - Added feature to remove configuration for second channel on LPC51U68.
- 2.0.2
  - Added ENABLE_IRQ handle after register I2S interrupt handle.
- 2.0.1
  - Unified component full name to FLEXCOMM I2S (DMA) driver.
- 2.0.0
  - Initial version.

SPI

The current SPI driver version is 2.0.4.

- 2.0.4
  - Bug fix:
    - Fixed the bug of using read only mode in DMA transfer. In DMA transfer mode, if transfer->txData is NULL, code attempts to read data from the address of 0x0 for configuring the last frame.
- 2.0.3
  - Added "SPI_FIFO_DEPTH(base)" more definition.
- 2.0.2
  - Unified component full name to FLEXCOMM SPI(DMA/FREERTOS) driver.
- 2.0.1
  - Changed the data buffer from uint32_t to uint8_t which matches the real applications for 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 either transmit first or receive first. Also, the PCS pin can be configured as assert status in transmission (between transmit and receive) by setting the isPcsAssertInTransfer to true.
- 2.0.0
  - Initial version.

USART

- The current USART driver version is 2.1.0.
- 2.1.0
  - New feature:
    - Added features to allow users to configure the USART to synchronous transfer (master

and slave) mode.

- 2.0.3
  - New feature:
    - Added new APIs to allow users to enable the CTS which determines whether CTS is used for flow control.
- 2.0.2
  - Bug fix:
    - Fixed the bug where transfer abort APIs cannot disable the interrupts. The FIFOINTENSET register should not be used to disable the interrupts, instead using the FIFOINTENCLR register.
- 2.0.1
  - Unified component full name to FLEXCOMM USART (DMA/FREERTOS) driver.
- 2.0.0
  - Initial version.

GINT

The current GINT 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.

IAP

The current IAP driver version is 2.0.1.

- 2.0.1
  - Optimization:
    - Removed the enumeration item kSysToFlashFreq_100MHz which cannot be supported.
    - Removed the invalid FFR commands.
    - Improved the format of IAP driver version, used S_VersionMajor to obtain the major version of bootloader.
- 2.0.0
  - Initial version.

INPUTMUX

The current INPUTMUX driver version is 2.0.0.

- 2.0.0
  - Initial version.

LPADC

The current LPADC driver version is 2.1.1.

- 2.1.1
  - Updated the gain calibration formula.
  - Used feature to segregate the new item kLPADC_TriggerPriorityPreemptSubsequently.
- 2.1.0
  - New features:
    - Added the API LPADC_SetOffsetValue() to support configure offset trim value manually.
    - Added the API LPADC_DoOffsetCalibration() to do offset calibration independently.
  - Improvements:
    - Improved the usage of macros and remove invalid macros.
- 2.0.2
  - Added support for platforms with 2 FIFOs and different calibration measures.
- 2.0.1
  - Ensured the API LPADC_SetConvCommandConfig configure related registers correctly.
- 2.0.0
  - Initial version.

CRC

The current CRC driver version is 2.0.1.

- 2.0.1
  - Fixed KPSDK-13362. MDK compiler issue when writing to WR_DATA with -O3 optimize for time.
- 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. DM-A_InstallDescriptorMemory. DMA_SubmitChannelTransfer. DMA_SetChannelConfig-Valid. 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.

IOCON

The current IOCON driver version is 2.1.1.

- 2.1.1
  - Updated left shift format with mask value instead of a constant value to automatically adapt to all platforms.
- 2.1.0
  - Added a new IOCON_PinMuxSet() function with a feature IOCON_ONE_DIMENSION for LPC845MAX board.
- 2.0.0
  - Initial version.

RTC

The current RTC driver version is 2.0.0.

- 2.0.0
  - Initial version.

MAILBOX

The current MAILBOX driver version is 2.1.0.

- 2.1.0
  - Added support for the LPC55S69 series, cpu_id parameter can be newly assigned to kMAILBOX_CM33_Core0 or kMAILBOX_CM33_Core1.
- 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.

PDB

The current PDB driver version is 2.0.1.

- 2.0.1
  - Changed PDB register base array to const.
- 2.0.0
  - Initial version.

OSTIMER

The current OSTIMER driver version is 2.0.1.

- 2.0.1
  - Removed the software reset function out of the initialization API.
  - Enabled interrupt directly instead of enabling deep sleep interrupt. Users need to enable the deep sleep interrupt in application code if needed.
- 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.

PLU

The current PLU 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.

POWERQUAD

- 2.0.1
  - Fixed build error in C++ mode.
- 2.0.0
  - Initial version.

PRINCE

The current PRINCE 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.

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

SDIF

The current SDIF driver version is 2.0.11.

- 2.0.11
  - Improvements:
    - Added API SDIF_GetEnabledInterruptStatus/SDIF_GetEnabledDMAInterruptStatus and used in SDIF_TransferHandleIRQ.
    - Removed useless members interruptFlags/dmaInterruptFlags in the sdif_handle_t.
    - Improved SDIF_SendCommand with return success directly when timeout is 0.
    - Added timeout error check when sending update clock command in SDIF_SetCardClock.
    - Removed START_CMD status polling for normal command sending in SDIF_TransferBloking/SDIF_TransferNonBlocking.
    - Disabled timeout parameter in function SDIF_SendCommand.
  - Bug fix:
    - Added delay cycle for the default speed mode(400 K and 25 M) to fix the timing issue when different AHB clock is configured.
- 2.0.10
  - Bug fix:
    - Fixed API SDIF_EnableCardClock that cannot clear the clock enable bit issue.
- 2.0.9
- Bug fix:
  - Fixed MDK 66-D warning.
- 2.0.8
- New features:
  - Added control macro to enable/disable the RESET and CLOCK code in current driver.
  - Disabled useless interrupt while DMA is used.
  - Updated SDIF driver for one instance support two card.
- 2.0.7

- Bug fix:
  - Enlarged the timeout value to avoid a command conflict issue.
- 2.0.6
- Bug fix:
  - Removed assert(srcClock_Hz <= FSL_FEATURE_SDIF_MAX_SOURCE_CLOCK).
  - Used hardware reset instead of software reset during initialize.
- 2.0.5
- New features:
  - Added non-word aligned data address and DMA descriptor address transfer support. Once one of the above addresses is not aligned, switch to host transfer mode.
- Bug fix:
  - Fixed the DMA suspend during initialization issue.
  - Removed useless memset function call.
- 2.0.4
  - Added cardInserted/cardRemoved callback function.
  - Added host base address/user data parameter for all call back functions.
- 2.0.3
  - Improved Clock Delay macro to allow the user to redefine and remove useless delay for clock below 25 MHz.
- 2.0.2
  - Bug fix:
    - Fixed the issue where the status flag cannot be cleared entirely after transfer complete.
- 2.0.1
  - New features:
    - Improve interrupt transfer callback.
  - Bug fix:
    - Added assert to limit the SDIF source clock below 52 MHz.
- 2.0.0
  - Initial version.

SYSCTL

The current SYSCTL driver version is 2.0.2.

- 2.0.2
  - Removed kSYSCTL_Flexcomm3DataOut enumeration definition.
- 2.0.1
  - Fixed some typo error comments and improved driver integral ability.
- 2.0.0
  - Initial version.

UTICK

The current UTICK driver version is 2.0.2.

- 2.0.2
  - Added new feature definition macro to enable/disable power control in drivers for some devices have no power control function.
- 2.0.1
  - Added control macro to enable/disable the 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.

FLASHIAP

Current FLASHIAP driver version is 2.0.0

- 2.0.0
  - Initial version.

FLEXCOMM

Current FLEXCOMM driver version is 2.0.0

- 2.0.0
  - Initial version.

I2C

Current I2C driver version is 2.0.0

- 2.0.0
  - Initial version.

I2S

Current I2S driver version is 2.0.0

- 2.0.0
  - Initial version.

SPI

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 will match the real applications for SPI DMA driver.
  - Add dummy data setup API to allow users to configure the dummy data to be transferred.

USART

Current USART driver version is 2.0.0

- 2.0.0
  - Initial version.

GINT

Current GINT driver version is 2.0.0

- 2.0.0
  - Initial version.

INPUTMUX

Current INPUTMUX driver version is 2.0.0

- 2.0.0

  - Initial version.

INTMUX

Current INTMUX driver version is 2.0.1

- 2.0.0
  - Initial version.
- 2.0.1
  - Improvement:
    - Add pubweak DriverIRQHandler for K32W042S1M2 which only has INTMUX1 with 8 channels.

CRC

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.

GPIO

Current GPIO driver version is 2.0.0

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

IOCON

Current IOCON driver version is 2.0.0

MRT

Current PIT driver version is 2.0.0

- 2.0.0
  - Initial version.

PINT

Current PINT driver version is 2.0.0

- 2.0.0
  - Initial version.

SPI Flash Interface

Current spifi driver version is 2.0.0

- 2.0.0
  - Initial version.

USDHC

Current USDHC driver version is 2.1.1

- 2.0.0
  - Initial version.
- 2.1.0
  - Intergrate tuning into transfer function
  - Add strobe DLL feature
  - Add enableAutoCommand23 in data structure
  - Remove enable card clock function due to controller will handle the clock on/off
- 2.1.1
  - Add cache maintain operation
  - Add timeout status check for the DATA transfer which ignore error.
  - Add feature macro for SDR50/SDR104 mode
  - Remove useless IRQ handler for different platform

2 Middleware Change Log

emWin library

The current version of emWin is 5.48r.

FatFs for MCUXpresso SDK

The current version is FatFs R0.13b_rev0.

- R0.13b_rev0
  - Upgraded to version 0.13b
- R0.13a_rev0
  - Upgraded to version 0.13a. Added patch ff_13a_p1.diff.
- R0.12c_rev1
  - Added NAND disk support.
- R0.12c_rev0
  - Upgraded to version 0.12c and applied patches ff_12c_p1.diff and ff_12c_p2.diff.
- R0.12b_rev0
  - Upgraded to version 0.12b.
- R0.11a
  - Added glue functions for low-level drivers (SDHC, SDSPI, RAM, MMC). Modified diskio.c.
  - Added RTOS wrappers to make FatFs thread safe. Modified syscall.c.
  - Renamed ffconf.h to ffconf_template.h. Each application should contain its own ffconf.h.
  - Included ffconf.h into diskio.c to enable the selection of physical disk from ffconf.h by macro definition.
  - Conditional compilation of physical disk interfaces in diskio.c.

mbedTLS for MCUXpresso SDK

The current version of mbedTLS is based on mbedTLS 2.13.1 released 2018-09-06.

- 2.13.1_rev3
  - Bug fixes:
    - Forced to align AES_CCM and AES_GCM self-test keys to fix unaligned key issue when using hardware acceleration.
- 2.13.1_rev2
  - Bug fixes:
    - Disabled default hardware acceleration of SHA in parallel with AES.
- 2.13.1_rev1
  - Bug fixes:
    - Fixed incorrect macro check when skipping AES-192 or AES-256.
- 2.13.1

  - New features:
    - Ported mbedTLS 2.13.1 to SDK.
- 2.12.0_rev1
  - New features:
    - Added support for NIST P-256 elliptic curve with CASPER driver.
- 2.12.0
  - New features:
    - Ported mbedTLS 2.12.0 to SDK.
- 2.9.0_rev2
  - New features:
    - Added support for Hashcrypt driver.
- 2.9.0_rev1
  - New features:
    - Added support for CASPER driver.
- 2.9.0
  - New features:
    - Ported mbedTLS 2.9.0 to SDK.
- 2.6.0_rev2
  - Bug fixes:
    - ssl_cookie.c now uses SHA256 for COOKIE_MD (instead of original SHA224). Some hardware crypto acceleration (such as CAU3) does not support SHA224 but all support SHA256.
- 2.6.0_rev1
  - Bug fixes:
    - ksdk_mbedtls.c bignum functions now read sign of input mbedtls_mpi at beginning of functions to properly support in place computations (when output bignum is the same as one of input bignums). Affected functions: mbedtls_mpi_mul_mpi(), mbedtls_mpi_mod_mpi(), ecp_mul_comb().
- 2.6.0
  - New features:
    - Ported mbedTLS 2.6.0 to KSDK.
    - Added MBEDTLS_FREESCALE_FREERTOS_CALLOC_ALT to allow alternate implementation of pvPortCalloc() when using .c.
- 2.5.1_rev1
  - New features:
    - Added support for DCP driver.
- 2.5.1
  - New features:
    - Ported mbedTLS 2.5.1 to SDK.
- 2.4.2_rev2
  - New features:
    - Added Curve25519 support for CAU3.
    - Added MBEDTLS_ECP_MUL_MXZ_ALT configuration parameter enabling overloading of ecp_mul_mxz().
- 2.4.2_rev1

  - New features:
    - Added support for CAU3 driver.
    - Added new files:
    - .c - contains regular software implementation of DES algorithm with added MBEDTLS_DES3_SETKEY_DEC_ALT and MBEDTLS_DES3_SETKEY_ENC_ALT config parameters.
    - .h - contains modified mbedtls_des_context and mbedtls_des3_context structures.
    - Added MBEDTLS_DES3_SETKEY_DEC_ALT configuration parameter enabling reloading of mbedtls_des3_set2key_dec() and mbedtls_des3_set3key_dec().
    - Added MBEDTLS_DES3_SETKEY_ENC_ALT configuration parameter enabling reloading of mbedtls_des3_set2key_enc() and mbedtls_des3_set3key_enc().
- 2.4.2
  - New features:
    - Ported mbedTLS 2.4.2 to KSDK 2.0.0.
    - Added CRYPTO_InitHardware() function.
    - Added new file:
      - .h - contains declaration of CRYPTO_InitHardware() function and should be included in applications.
- 2.3.0_rev1
  - New features:
    - Added support for CAAM driver.
    - In LTC-specific wrapper, allocate temporary integers from heap in one large block.
- 2.3.0
  - New features:
    - Ported mbedTLS 2.3.0 to KSDK 2.0.0.

2.2.1

- New features:
  - Ported mbedTLS 2.2.1 to KSDK 2.0.0.
  - Added support of MMCAU cryptographic acceleration module. Accelerated MD5, SHA, AES, and DES.
  - Added support of LTC cryptographic acceleration module. Accelerated AES, DES, and PKHA.
  - Added new files:
  - .c - alternative implementation of cryptographic algorithm functions using LTC and MMCAU module drivers.
  - .h - configuration settings used by mbedTLS KSDK bare metal examples.
  - Added mbedTLS KSDK bare-metal examples:
    - <board name> - KSDK mbedTLS benchmark application.
    - <board name> - KSDK mbedTLS self-test application.
  - Added MBEDTLS_GCM_CRYPT_ALT configuration parameter enabling reloading of mbedtls_gcm_crypt_and_tag().
  - Added MBEDTLS_ECP_MUL_COMB_ALT to enable alternate implementation of ecp_mul_comb().
  - Added MBEDTLS_ECP_ADD_ALT configuration parameter enabling reloading of ecp_-

add().

- Added MBEDTLS_DES_SETKEY_DEC_ALT configuration parameter enabling reloading of mbedtls_des_setkey_dec(), mbedtls_des3_set2key_dec() and mbedtls_des3_set3key_dec().
- Added MBEDTLS_DES_SETKEY_ENC_ALT configuration parameter enabling reloading of mbedtls_des_setkey_enc(), mbedtls_des3_set2key_enc() and mbedtls_des3_set3key_enc().
- Added MBEDTLS_DES_CRYPT_CBC_ALT configuration parameter enabling reloading of mbedtls_des_crypt_cbc().
- Added MBEDTLS_DES3_CRYPT_CBC_ALT configuration parameter enabling reloading of mbedtls_des3_crypt_cbc().
- Added MBEDTLS_AES_CRYPT_CBC_ALT configuration parameter enabling reloading of mbedtls_aes_crypt_cbc().
- Added MBEDTLS_AES_CRYPT_CTR_ALT configuration parameter enabling reloading of mbedtls_aes_crypt_ctr().
- Added MBEDTLS_CCM_CRYPT_ALT configuration parameter enabling reloading of mbedtls_ccm_encrypt_and_tag() and mbedtls_ccm_auth_decrypt().
- Added MBEDTLS_MPI_ADD_ABS_ALT configuration parameter enabling reloading of mbedtls_mpi_add_abs().
- Added MBEDTLS_MPI_SUB_ABS_ALT configuration parameter enabling reloading of mbedtls_mpi_sub_abs().
- Added MBEDTLS_MPI_EXP_MOD_ALT configuration parameter enabling reloading of mbedtls_mpi_exp_mod().
- Added MBEDTLS_MPI_MUL_MPI_ALT configuration parameter enabling reloading of mbedtls_mpi_mul_mpi().
- Added MBEDTLS_MPI_MOD_MPI_ALT configuration parameter enabling reloading of mbedtls_mpi_mod_mpi().
- Added MBEDTLS_MPI_GCD_ALT configuration parameter enabling reloading of mbedtls_mpi_gcd().
- Added MBEDTLS_MPI_INV_MOD_ALT configuration parameter enabling reloading of mbedtls_mpi_inv_mod().
- Added MBEDTLS_MPI_IS_PRIME_ALT configuration parameter enabling reloading of mbedtls_mpi_is_prime().
- Added encrypt/decrypt mode to mbedtls_des_context and mbedtls_des3_context structure.
- Added carriage return ” for mbedtls_printf() in self test functions.

SDMMC

Current driver version is 2.1.4

- 2.1.0
  - Bug Fix:
    - Change the callback mechanism when sending a command;
    - Fix the performance low issue when transferring data;
  - Other changes:
    - Change the name of some error codes returned by internal function;

        - Merge all host related attributes to one structure;
        - Optimize the function of setting maxmium data bus width for MMC card;
- 2.1.1
    - Bug Fix:
        - Fix the block range boundry error when transferring data to MMC card;
        - Fix the bit mask error in the SD card switch to high speed function;
    - Other changes:
        - Add error code to indicate that SDHC ADMA1 transfer type is not supported yet;
        - Optimize 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_Write-Blocks
- 2.1.3
    - Bug fix:
        - Non high speed sdcard init fail at switch to high speed.
    - Misc:
        - Optimize tuning/mmc switch voltage/mmc select power class/mmc select timing function
        - Add strobe dll for mmc HS400 mode
        - Add Delay for sdcard power up
- 2.1.4
    - Misc:
        - Add Host reset function for card re-initialization
        - Add Go_Idle function for SDIO card
        - Add Host_ErrorRecovery function for host error recovery procedure.
        - Add cache maintain operation
        - Add HOST_CARD_INSERT_CD_LEVEL to imporve compatibility.
    - Bug fix:
        - Fix card cannot detect dynamically.

Multicore SDK

The current version of Multicore SDK is 2.6.0.

- 2.6.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.2
    - eRPC generator (erpcgen) v.1.7.2
    - Multicore Manager (MCMgr) v4.0.3
    - RPMsg-Lite v2.2.0
  - New features:
    - eRPC: Improved support of const types.
    - eRPC: Fixed Mac build.
    - eRPC: Fixed serializing python list.
    - eRPC: Documentation update.
    - eRPC: Add missing doxygen comments for transports.
    - RPMsg-Lite: Added configuration macro RL_DEBUG_CHECK_BUFFERS.
    - RPMsg-Lite: Several MISRA violations fixed.
    - RPMsg-Lite: Added environment layers for QNX and Zephyr.
    - RPMsg-Lite: Allow environment context required for some environments (controlled by the RL_USE_ENVIRONMENT_CONTEXT configuration macro).
    - RPMsg-Lite: Data types consolidation.
    - MCMgr: Documentation updated to describe handshaking in a graphic form.
    - MCMgr: Minor code adjustments based on static analysis tool findings.
- 2.5.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.1
    - eRPC generator (erpcgen) v.1.7.1
    - Multicore Manager (MCMgr) v4.0.2
    - RPMsg-Lite v2.0.2
  - New features:
    - RPMsg-Lite, MCMgr: Align porting layers to the updated MCUXpressoSDK feature files.
    - eRPC: Fixed semaphore in static message buffer factory.
    - erpcgen: Fixed MU received error flag.
    - erpcgen: Fixed TCP transport.
- 2.4.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.0
    - eRPC generator (erpcgen) v.1.7.0
    - Multicore Manager (MCMgr) v4.0.1
    - RPMsg-Lite v2.0.1
  - New features:
    - eRPC: Improved code size of generated code.
    - eRPC: Generating CRC value is optional.
    - eRPC: Fixed CMSIS UART driver. Removed dependency on SDK.
    - eRPC: List names are based on their types. Names are more deterministic.

        - eRPC: Service objects are as a default created as global static objects.
        - eRPC: Added missing doxygen comments.
        - eRPC: Forbid users use reserved words.
        - eRPC: Removed outByref for function parameters.
        - eRPC: Added support for 64-bit numbers.
        - eRPC: Added support of program language specific annotations.
        - eRPC: Optimized code style of callback functions.
        - RPMsg-Lite: New API rpmsg_queue_get_current_size().
        - RPMsg-Lite: Fixed bug in interrupt handling for lpc5411x, lpc5410x.
        - RPMsg-Lite: Code adjustments based on static analysis tool findings.
- 2.3.1
    - Multicore SDK component versions:
        - embedded Remote Procedure Call (eRPC) v1.6.0
        - eRPC generator (erpcgen) v.1.6.0
        - Multicore Manager (MCMgr) v4.0.0
        - RPMsg-Lite v1.2.0
    - New features:
        - eRPC: Improved code size of generated code.
        - eRPC: Improved eRPC nested calls.
        - eRPC: Improved eRPC list length variable serialization.
        - eRPC: Added support for scalar types.
        - MCMgr: Added new MCMGR_TriggerEventForce() API.
- 2.3.0
    - Multicore SDK component versions:
        - embedded Remote Procedure Call (eRPC) v1.5.0
        - eRPC generator (erpcgen) v.1.5.0
        - Multicore Manager (MCMgr) v3.0.0
        - RPMsg-Lite v1.2.0
    - New features:
        - eRPC: Added support for unions type non-wrapped by structure.
        - eRPC: Added callbacks support.
        - eRPC: Added support annotation for functions.
        - eRPC: Added support

NTAG I2C plus library

The current version is 1.0.0.

- 1.0.0
    - initial release.

SDMMC

The current driver version is 2.2.11.

- 2.2.11
  - Bug fixes:
    - Fixed NULL pointer dereference issue when calling function SDMMCHOST_CardDetectInit in host adaptor layer.
    - Fixed logical dead code issue in SDMMC_SwitchToVoltage function.
- 2.2.10
  - Bug fixes:
    - Added NUll pointer check for USDHC FreeRTOS adaptor transfer complete. callback.
    - Added event value check for all the FreeRTOS event to fix program hang when a card event occurs before created.
- 2.2.9
  - Improvements:
    - Added NULL pointer check for sdmmchostcard_usr_param_t member CD in card detect callback to avoid memory corruption.
    - Added card voltage switch function in sdmmhostcard_usr_param_t to allow application register card signal line voltage switch function.
  - Bug fixes:
    - Fixed host FreeRTOS adaptor and polling adaptor, where they cannot detect card insert bug for USDHC.
    - Fixed SDHC host layer build issue and typo issue.
- 2.2.8
  - Improvement:
    - Updated SDMMC to support SDIO interrupt.
- 2.2.7
  - Bug fixes:
    - Fixed MDK 66-D warning.
- 2.2.6
  - Improvements:
    - Removed some SoC-specific header files from porting layer.
    - Saved MMC OCR registers while sending CMD1 with argument 0.
  - Bug fixes:
    - Added MMC_PowerOn function in which there is delay function after powerup SD Card. Otherwise, the card may init fail.
- 2.2.5
  - New features:
    - Added SD_ReadStatus API to get 512-bit SD status.
    - Added error log support in SD Card functions.
    - Added SDMMC_ENABLE_SOFTWARE_TUNING to enable/disable software tuning. It is disabled by default.
    - Added error procedure in the transfer function to improve stability.
    - Removed deprecated GPIO API in host layer.
- 2.2.4

- Bug fix:
    - Fixed DDR mode data sequence miss issue, which is caused by NIBBLE_POS.
- New features:
    - Increased g_sdmmc 512 Bytes to improve performance when application uses a non-word align data buffer address.
    - Used OCR access mode bits to determine the MMC Card high capacity flag.
    - Enabled auto CMD12 for SD read/write.
    - Disabled DDR mode frequency multiply by 2.

- 2.2.3
    - Bug fix:
        - Added response check for send operation condition command. If not checked, the card may occasionally init fail.
- 2.2.2
    - Moved set card detect priority operation before enable IRQ.
- 2.2.1
    - New features:
        - Improved MMC Boot feature.
        - Kept SD_Init/SDIO_Init function for forward compatibility.
- 2.2.0
    - New features:
        - Separated the SD/MMC/SDIO init API to xxx_CardInit/xxx_HostInit.
        - Allowed user register card detect callback, select card detect type, and determine the card detect timeout value.
        - Allowed user register the power on/off function, and determine the power on/off delay time.
        - SD_Init/SDIO_Init is deprecated in the next version.
        - Added write complete wait operation for MMC_Write to fix command timeout issue.
- 2.1.6
    - Enhanced SD IO default driver strength.
- 2.1.5
    - Fixed coverity issue.
    - Fixed SD v1.x card write fail issue. It was caused by the block length set error.
- 2.1.4
    - Miscellaneous:
        - Added Host reset function for card re-initialization.
        - 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 card cannot detect dynamically.
- 2.1.3
    - Bug fix:
        - Non high-speed sdcard init fail at switch to high speed.
    - Miscellaneous:
        - Optimized tuning/mmc switch voltage/mmc select power class/mmc select timing func-

        tion.
        * Added strobe dll for mmc HS400 mode.
        * Added Delay for SDCard power up.
- 2.1.2
    - New features:
        * Added fsl_host.h to provide prototype to adapt different controller IPs(SDHC/SDIF).
        * Added adaptor code in SDMMC/Port folder to adapt different host controller IPs with different. transfer modes(interrupt/polling/freertos). Application includes a different adaptor code to make application more simple.
        * 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).
        * This change requires the user to include host adaptor code into the application. If not changed, link errors saying it cannot find the definition of HOST_Init/HOST_Deinit/-CardInsertDetect appear.
    - New features: Improved SDMMC to support SD v3.0 and eMMC v5.0.
    - Bug fix:
        * Fixed incorrect comparison between count and length in MMC_ReadBlocks/MMC_-WriteBlocks.
- 2.1.1
    - Bug fix:
        * Fixed the block range boundary error when transferring data to MMC card.
        * Fixed the bit mask error in the SD card switch to high speed function.
    - Other changes:
        * Added error code to indicate that SDHC ADMA1 transfer type is not supported yet.
        * Optimized the SD card initialization function.
- 2.1.0
    - Bug fix:
        * Changed callback mechanism when sending a command.
        * Fixed low performance issue when transferring data.
    - Other changes:
        * Changed the name of some error codes returned by internal function.
        * Merged all host related attributes to one structure.
        * Optimize the function of setting maximum data bus width for MMC card.

SDIO

The current driver version is 2.2.11.

- 2.2.11
    - Bug fix:
        * Added check card async interrupt capability in function SDIO_GetCardCapability.
        * Fixed OUT OF BOUNDS access in function SDIO_IO_Transfer.

- 2.2.10
  - Bug fix:
    - Fixed SDIO card driver get wrong IO number when the card IO number is bigger than 2.
  - New feature:
    - Added SDIO 3.0 support.
    - Added API SDIO_IO_RW_Direct for direct read/write card register access.
- 2.2.9
  - Improvement:
    - Added API SDIO_SetIOIRQHandler/SDIO_HandlePendingIOInterrupt to handle multi IO pending IRQ.
- 2.2.8
  - Improvement:
    - Updates SDMMC to support SDIO interrupt.
    - Added API SDIO_GetPendingInterrupt to get the pending IO interrupt.
- 2.2.7
  - Bug fix:
    - Fixed MDK 66-D warning.
- 2.2.6
  - New features:
    - Added a unified transfer interface for SDIO.
  - Bug fix:
    - Wrong pointer address used by SDMMCHOST_Init.
- 2.1.5
  - Bug fix:
    - Improved SDIO card init sequence and added retry option for SDIO_SwitchToHighSpeed function.
- 2.1.4
  - Miscellaneous:
    - Added Go_Idle function for SDIO card.
- 2.0.0
  - Initial version.

SDSPI

The current SDSPI driver version is 2.1.4.

- 2.1.4
  - Bug fix:
    - Fixed MDK 66-D warning.
- 2.1.3
  - Improved SDSPI code size and performance.
- 2.0.0
  - Initial version.

USB stack for MCUXpresso SDK

The current version of USB stack is 2.2.0.

- 2.2.0
  - New features:
    - Added device DFU support.
    - Supports OM13790DOCK on LPCXpresso54018.
    - Added multiple logical unit support in MSC class driver, updated usb_device_lba_information_struct_t to support this.
    - Supports multiple transfers for host ISO on IP3516HS.
  - Bug fixes:
    - Fixed device ip3511 prime data length than maxpacket size issue.
    - Initialized interval attribute in usb_device_endpoint_struct_t/usb_device_endpoint_init_struct_t.
    - Removed unnecessary header file in device CDC class driver, removed unnecessary usb_echo, and added DEBUG macro for necessary usb_echo in device CDC class driver.
    - Fixed device IP3511HS unfinished interrupt transfer missing issue.
- 2.1.0
  - New features:
    - Added host RNDIS support. example: lwip_dhcp_usb
    - Enabled USB 3.0 support on device stack.
    - Power Delivery feature: Added OM13790HOST support. Added auto policy feature. Printed e-marked cable information.
- 2.0.1
  - Bug fixes:
    - Fixed some USB issues: Fixed MSC CV test failed in MSC examples.
    - Changed audio codec interfaces.
- 2.0.0
  - New features:
    - PTN5110N support.
  - Bug fix:
    - Added some comments, fixed some minor USB issues.
- 1.9.0
  - New features:
    - Examples:
      - usb_pd_alt_mode_dp_host
- 1.8.2
  - Updated license.
- 1.8.1
  - Bug fix:
    - Verified some hardware issues, support aruba_flashless.
- 1.8.0
  - New features:
    - Examples:
      - usb_device_composite_cdc_vcom_cdc_vcom

- usb_device_composite_hid_audio_unified
- usb_pd_sink_battery
- Changed usb_pd_battery to usb_pd_charger_battery.

```
- Bug fix:
  - Code clean up, removed some irrelevant code.
```

- 1.7.0
  - New features:
    - USB PD stack support.
  - Examples:
    - usb_pd
    - usb_pd_battery
    - usb_pd_source_charger
- 1.6.3
  - Bug fix: -IP3511_HS driver control transfer sequence issue, enabled 3511 ip cv test.
- 1.6.2
  - New features:
    - Multi instance support.
- 1.6.1
  - New features:
  - Changed the struct variable address method for device_video_virtual_camera and host_phdc-_manager.
- 1.6.0
  - New features:
    - Supported Device Charger Detect feature on usb_device_hid_mouse.
- 1.5.0
  - New features:
    - Supported controllers
      - OHCI (Full Speed, Host mode)
      - IP3516 (High Speed, Host mode)
      - IP3511 (High Speed, Device mode)
    - Examples:
      - usb_lpm_device_hid_mouse
      - usb_lpm_device_hid_mouse_lite
      - usb_lpm_host_hid_mouse
- 1.4.0
  - New features:
    - Examples:
      - usb_device_hid_mouse/freertos_static
      - usb_suspend_resume_device_hid_mouse_lite
- 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
- 1.2.0
  - New features:
    - Supported controllers
      - LPC IP3511 (Full Speed, Device mode)
- 1.1.0
  - Bug fix:
    - Fixed some issues in USB certification.
    - Changed VID and Manufacturer string to NXP.
  - New features:
    - Supported classes
      - Pinter
    - Examples:
      - usb_device_composite_cdc_msc_sdcard
      - usb_device_printer_virtual_plain_text
      - usb_host_printer_plain_text
- 1.0.1
  - Bug fix:
    - Improved the efficiency of device audio speaker by changing the transfer mode from interrupt to DMA, thus providing the ability to eliminate the periodic noise.
- 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
· usb_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

QCA WiFi

The current version is 2.0.0.

- 2.0.0
  - Initial version.
    - Added QCA WiFi, ported from SDK 1.3, synchronized with latest MQX Qualcomm v3.-3.5.
  - Known issues:
    - Low power mode may not work, requires further investigation.
    - DHCP request requires a timeout to retrieve valid data.

3 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_SUPPORT 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.

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: MCUXSDKLPC55XXRN