NXP Semiconductors
Release Notes

Document Number: MCUXSDKLPC55XXRN
Rev. 0, 01/2019

MCUXpresso SDK Release Notes Supporting LPC55xx

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 mbed TLS cryptography libraries, other middleware packages, such as multicore support and FatFs, 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 the latest version of this and other MCUXpresso SDK documents, 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.

Contents

2 MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpressoSDK is the evolution of Kinetis SDK v2.5.0, 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 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.

3 Development tools

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

- IAR Embedded Workbench® for Arm® version 8.32.1 with ewarm8321-lpc55xx-2.1.zip
- MDK-Arm Microcontroller Development Kit (Keil)® 5.26 with NXP.LPC55S69_DFP.11.0.0.pack
- Makefiles support with GCC revision 7-2018-q2-update from Arm Embedded
- MCUXpresso IDE v10.3.0 build 2200

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

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
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

Table continues on the next page...

Table 2. Release contents (continued)

mbed TLS examples	<install_dir>/boards/<board_name>/mbedtls_examples
NTAG examples	<install_dir>/boards/<board_name>/ntag_i2c_plus_examples
Multicore examples	<install_dir>/boards/<board_name>/multicore_examples
CMSIS driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
emWin examples	<install_dir>/boards/<board_name>/emwin_examples
FatFs examples	<install_dir>/boards/<board_name>/fatfs_examples
Trustzone examples	<install_dir>/boards/<board_name>/trustzone_examples
Wifi_qca examples	<install_dir>/boards/<board_name>/wifi_qca_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
Middleware	<install_dir>/middleware
SDMMC card driver	<install_dir>/middleware/sdmmc
USB stack	<install_dir>/middleware/usb
Multicore stack	<install_dir>/middleware/multicore
FatFS stack	<install_dir>/middleware/fatfs
NTAG	<install_dir>/middleware/ntag_i2c_plus
Mcu-boot	<install_dir>/middleware/mcu-boot
mbed TLS	<install_dir>/middleware/mbedtls
wifi_qca	<install_dir>/middleware/wifi_qca
emWin	<install_dir>/middleware/emwin
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
RTOS Kernal Code	<install_dir>/rtos
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 demonstrate the capabilities of the peripheral drivers. Each example implements a common use case to help demonstrate the driver functionality.

The RTOS and 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 MCUXSDKUSBSUG) for more information.

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 NTAG_I2C_PLUS

The NTAG I2C plus communication library can be used for communication with NXP NTAG I2C plus (a family of connected NFC tags that combine a passive NFC interface with a contact I2C interface) over I2C.

6.2.6 SDMMC

The SDMMC software is integrated with MCUXpresso SDK to support SD/MMC/SDIO standard specification. This also includes a host adapter layer for bare-metal/RTOS applications.

6.2.7 emWin

The emWin graphics library is integrated with MCUXpresso SDK and is suitable for creation of a highly efficient and high quality GUI on any embedded system. It is independent from any target and display and can be used with either single-task or multitask environments with any operating system.

6.2.8 Multicore

This software package contains components for efficient work with multicore devices as well as for multiprocessor communication. See the release notes in the *docs/multicore* and the change log in the *middleware/multicore* for more information.

6.2.9 mcu-boot: MCU Bootloader (MCUBOOT)

The MCUXpresso SDK is integrated with host tools/source code for MCU Bootloader. See the release notes in the *middleware/mcu-boot/doc* for more information.

6.2.10 mbedtls: Cryptographic and SSL/TLS Library

The MCUXpresso SDK is integrated with the mbed TLS Cryptographic and SSL/TLS Library.

6.2.11 wifi_qca: Qualcomm Wi-Fi stack

The MCUXpresso SDK is integrated with the Qualcomm Wi-Fi stack.

7 MISRA compliance

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

Table 3. MISRA exceptions

Exception Rules	Description
Directive 4.4	Sections of code should not be commented out.
Directive 4.5	Identifiers in the same name space with overlapping visibility should be typographically unambiguous.
Directive 4.6	Typedef that indicate size and signedness should be used in place of the basic numerical type.
Directive 4.8	If a pointer to a structure or union is never dereferenced within a transaction unit then the implementation of the object should hidden.
Directive 4.9	A function should be used in preference to a function like macro where they are interchangeable.
Directive 4.10	Precautions shall be taken in order to prevent the contents of a header file being included more than once.
Directive 4.11	The validity of values passed to library functions shall be checked.
Rule 2.3	A project should not contain unused type declarations.
Rule 2.4	A project should not contain unused tag declarations.
Rule 2.5	A project should not contain unused macro declarations.
Rule 2.7	There should be no unused parameters in functions.
Rule 3.1	The character sequences /* and // shall not be used within a comment.
Rule 5.1	External identifiers shall distinct.
Rule 5.3	A identifier declared in an inner scope shall not hide an identifier declared in an outer scope.
Rule 5.7	A tag name shall be a unique identifier.
Rule 5.9	Identifiers that define objects or functions with external linkage shall be unique.
Rule 8.13	A pointer should point to a const-qualified type whenever possible.
Rule 8.3	All declarations of an object or function shall use the same names and type qualifiers.
Rule 8.6	An identifier with external linage shall have exactly one external definition.
Rule 8.7	Octal constants shall not be used.
Rule 8.9	A object should be defined at block scope if its identified only appears in a single function.
Rule 10.1	Operands shall not be of an inappropriate essential type.
Rule 10.3	The value of an expression shall not be assigned to an object with a narrower essential type of a different essential type category.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

Rule 10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.
Rule 10.5	The value of an expression should not be cast to an inappropriate essential type.
Rule 10.6	The value of a composite expression shall not be assigned to an object with wider essential type.
Rule 10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type.
Rule 10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.
Rule 11.1	Conversions shall not be performed between a pointer to a function and any other type.
Rule 11.3	A case shall not be performed between a pointer to object type and a pointer to a different object type.
Rule 11.4	A conversion should not be performed between a pointer to object and an integer type.
Rule 11.5	A conversion should not be performed from pointer to void into pointer to object.
Rule 11.6	A cast shall not be performed between pointer to void and an arithmetic type.
Rule 12.1	The precedence of operators within expressions should be made explicit.
Rule 12.2	The right hand operator of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand.
Rule 13.3	A full expression containing an increment(++) or decrement(--) operator should have no other potential side effects other than that caused by the increment or decrement operator.
Rule 13.5	The right hand operand of a logical && or || operator shall not contain persistent side effects.
Rule 14.2	A for loop shall be well formed.
Rule 14.4	The controlling expressions of an statement and the controlling expression of an iteration-statement shall have essentially Boolean type.
Rule 15.5	A function should have a single point of exit at the end.
Rule 16.1	All switch statements shall be well-formed.
Rule 17.1	The feature of <stdarg.h> shall not be used.
Rule 18.4	The +,-,+=and -=operators should not be applied to an expression of pointer type.
Rule 19.2	The union keyword should not be used.
Rule 20.1	#include directives should only be preceded by preprocessor directives or comments.
Rule 20.10	The #and ## preprocessor operators should not be used.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

Rule 21.1	#define and #undef shall not be used on a reserved identifier or reserved macro name.

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 USBHS controller issue

Because of the USBHS controller design issue, the host examples of high-speed controller do not support the low-speed device directly.

8.3 Create new project without board template

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

8.4 arm_math.h in CMSIS v5.3 does not support Cortex-m33

While building CMSIS-PACK PowerQuad examples in IAR and Keil, use version 5.3 CMSIS component and manually sync `arm_math.h` with `<sdk_root>\CMSIS\include\arm_math.h` For Keil, the `arm_math.h` is in the *ARM\PACK\ARM\CMSIS\5.3.0\CMSIS\Include* folder. For IAR, the `arm_math.h` is in the *<iar cmsis-pack directory>\ARM\CMSIS\5.3.0\CMSIS\Include* folder.

8.5 TrustZone example and PowerQuad example cannot build with Redlib in MCUXpresso IDE

In MCUXpresso IDE, while building the TrustZone or PowerQuad example, make sure the library is using newlibnano, not redlib.

8.6 Download the binary failed after running the case of Trustzone

After running the case of Trustzone, short the J10 on the board to download a new example (if the download still fails, use J-Link to erase the flash), then open the J10 and run new example.

MCUXpresso SDK Release Notes Supporting LPC55XX

Change Logs

Contents

Contents

Title **Page Number**

Contents

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.0.2.

- 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 statements 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 2.0.0.

- 2.0.0
  - initial version.

ANACTRL

The current ANACTRL driver version is 2.0.0.

- 2.0.0
  - Initial version.

COMMON

The current COMMON driver version is 2.0.1.

- 2.0.1
  - Removed the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
  - Added new feature macro switch "FSL_FEATURE_HAS_NO_NONCACHEABLE_SECTI-

    ON" for a specific SoC which has no non-cacheable sections, this helps avoid an unnecessary complex in link file and 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 a control macro to enable/disable the RESET and CLOCK code in current driver.
  - Added a 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 2.0.0.

- 2.0.0
  - Initial version.

I2C

The current I2C driver version is 2.0.3.

- 2.0.3
  - Unify 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

        ack 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 2.0.2.

- 2.0.2
  - Add ENABLE_IRQ handle after register i2s interrupt handle
- 2.0.1
  - Unify component full name to FLEXCOMM I2S(DMA) Driver
- 2.0.0
  - Initial version.

SPI

The current SPI driver version is 2.0.3.

- 2.0.3
  - Bug Fix:
    - Fix the bug of using read only mode in DMA transfer. In DMA transfer mode, if xfer->txData is NULL, code will attempt to read data from the address of 0x0 for configuring the last frame.
- 2.0.2
  - Unify 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 receivefirst. Besides, 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 2.0.3.
- 2.0.3
  - New feature:
    - Add new API to allow users enable the CTS which determines whether CTS is used for flow control.
- 2.0.2
  - Bug Fix:
    - Fix the bug of transfer abort APIs can not disable the interrupts, the FIFOINTENSET register should not be used to disable the interrupts, using FIFOINTENCLR register instead.
- 2.0.1
  - Unify component full name to FLEXCOMM USART(DMA/FREERTOS) Driver
- 2.0.0
  - Initial version.

GINT

The current GINT driver version is 2.0.01.

- 2.0.1
  - Add 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 2.0.0.

- 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 2.0.3.

- 2.0.3
  - Improvements:
    - Add the API LPADC_SetOffsetValue() to support configure offset trim value manually.
    - Add the API LPADC_DoOffsetCalibration() to do offset calibration independently.
    - Improve the usage of macros and remove invalid macros.
- 2.0.2
  - Add supports for platforms with 2 FIFOs and different calibration measures.
- 2.0.1
  - Ensure 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 2.2.1.

- 2.2.1
  - Fix coverity issue.
- 2.2.0
  - Change api DMA_SetupDMADescriptor to non-static.
  - Mark below api as deprecated. DMA_PrepareTransfer. DMA_Submit transfer.
  - Added below new api: DMA_SetChannelConfig. DMA_PrepareChannelTransfer. DMA_-InstallDescriptorMemory. DMA_SubmitChannelTransfer. DMA_SetChannelConfigValid. D-MA_DoChannelSoftwareTrigger. DMA_LoadChannelTransferConfig.
- 2.0.1
  - Add volatile for DMA descriptor member xfercfg to avoid optimization.
- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.1.3.

- 2.1.3:

  - Update "GPIO_PinInit" API. if it have DIRCLR and DIRSET registers, using them set 1 or clean 0.
- 2.1.2:
  - Remove deprecated APIs.
- 2.1.1:
  - API interface changes:
    - Refined naming of API while keeping all original APIs, marking them as deprecated. Original API will be removed in next release. The main change is update API with prefix of _-PinXXX() and _PortXXX
- 2.1.0
  - Added GPIO initialize API.
- 2.0.0
  - Initial version.

IOCON

The current IOCON 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.

MAILBOX

The current MAILBOX driver version is 2.1.0.

- 2.1.0
  - Adds support for the LPC55S69 series, cpu_id parameter can be newly assigned to kMAILBO-X_CM33_Core0 or kMAILBOX_CM33_Core1.
- 2.0.0
  - Initial version.

MRT

The current MRT driver version is 2.0.1.

- 2.0.1

  - Add 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 2.0.0.

- 2.0.0
  - Initial version.

PINT

The current PINT driver version is 2.0.4.

- 2.0.4
  - Inprovements: Enabled secure pint interrupt and add secure interrupt handle.
- 2.0.3 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
  - Add control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0

  - Initial version.

POWERQUAD

- 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 2.0.1.

- 2.0.1
  - Add 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 2.0.10.

- 2.0.10
- Bug fix:
  - Fixed API where SDIF_EnableCardClock cannot clear 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 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 2.0.1.

- 2.0.1
    - Fixed some comments typo error and improve driver integrality.
- 2.0.0
    - Initial version.

UTICK

The current UTICK driver version is 2.0.2.

- 2.0.2
  - Add new feature definition macro to enable/disable power control in drivers for some devices have no power control function.
- 2.0.1
  - Add control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

WWDT

The current WWDT driver version is 2.1.1.

- 2.1.1
  - Add new feature definition macro for devices have no LCOK control bit in MOD register.
- 2.1.0
  - Add new parameter in configuration when initializing WWDT module, this parameter will allow user to deliver the WWDT clock frequency, and this parameter must be set.
- 2.0.0
  - Initial version.

2 Middleware Change Log

emWin library

The currently supported version is 5.48.

FatFs for MCUXpresso SDK

Current version is FatFs R0.13a_rev0.

- R0.13a_rev0
  - Upgraded to version 0.13a. Added patch ff_13a_p1.diff.
- R0.12c_rev1
  - Add 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 2.12.0 released 2018-July-25.

- 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 KSDK.
- 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 KSDK.
- 2.6.0_rev2
  - Bug fixes:
    - ssl_cookie.c now uses SHA256 for COOKIE_MD (instead of original SHA224). Some hw crypto acceleration (such as CAU3) don't 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 KSDK.
- 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.

Multicore SDK

The current version of Multicore SDK is 2.4.0.

- 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 v1.2.0
  - New features:
    - eRPC: Improved code size of generated code.
    - eRPC: Generating crc value is optional.
    - eRPC: Fixed CMSIS UART driver. Removed dependency on MCUXpresso 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 64bit numbers.
    - eRPC: Added support of program language specific annotations.
    - eRPC: Optimized code style of callback functions.
- 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 2.2.7.

- 2.2.7
    - Bug fix:
        - 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 fix:
        - Added MMC_PowerOn function in which there is delay function after powerup sdcard. Otherwise, the card may init failed.
- 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 and 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 512byte to improve the performance when application use a non-word align data buffer address.
        - Used OCR access mode bits to determine the mmccard 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.
        - Keep 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 will be 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 SD card init fail at switch to high speed.
  - Miscellaneous:
    - Optimized tuning/mmc switch voltage/mmc select power class/mmc select timing function.
    - 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:
    - Change the callback mechanism when sending a command.
    - Fix the performance low 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 SDIO driver version is 2.2.7.

- 2.2.7
  - Bug fix:
    - Fixed MDK 66-D warning.
- 2.2.6
  - New features:
    - Added a unify transfer interface for SDIO.
  - Bug fix:
    - Wrong pointer address used by SDMMCHOST_Init.
- 2.1.5
  - Bug fix:
    - Improved SDIO card init sequence and add 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 driver version is 2.1.3.

- 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 2.1.0.

- 2.1.0
  - New features:
    - add host RNDIS support. example: lwip_dhcp_usb
    - enable USB 3.0 support on device stack.
    - pd feature Add OM13790HOST support; Add auto policy feature; Print e-marked cable information;
- 2.0.1
  - Bug fix:
    - fixed some USB issues: fix msc cv test failed in msc examples.
    - Change the 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, require further investigation.
    - DHCP request requires some timeout to retrieve valid data.

3 RTOS Change Log

FreeRTOS for MCUXpresso SDK

The current version is FreeRTOS 9.0.0. Original package is available at `freertos.org`.

- 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