NXP Semiconductors
Release Notes

Document Number: MCUXSDKMIMXRT105XRN
Rev. 0, 11/2017

MCUxpresso SDK Release Notes Supporting EVK-MIMXRT1050

Contents

1 Overview....1
2 Development Tools.... 1
3 Supported Development Systems....2
4 Release Contents.... 2
5 MCUXpresso SDK Release Package... 3
6 MISRA Compliance.... 5
7 Change Log - Peripheral drivers.... 6
8 Change Log - Middleware.... 17
9 Change Log - RTOS.... 23

1 Overview

The MCUXpresso Software Development Kit (SDK) is a collection of software enablement for Microcontrollers that includes peripheral drivers, high-level stacks including USB and lwIP, integration with WolfSSL and mbed TLS cryptography libraries, other middleware packages, such FatFs, and integrated RTOS support for FreeRTOS™ OS. In addition to the base enablement, the MCUXpresso SDK is augmented with demo applications and driver example projects, and API documentation to help the customers quickly leverage the support of the MCUXpresso SDK.

For the latest version of this and other MCUXpresso SDK documents, see the MCUXpresso SDK homepage MCUXpresso-SDK: Software Development Kit.

2 Development Tools

The MCUXpresso SDK for EVK-MIMXRT1050 was compiled and tested with these development tools:

- IAR Embedded Workbench for ARM version 8.11.3
- MDK-ARM Microcontroller Development Kit (Keil)® 5.23

- Makefiles support with GCC revision v6-2017-q2 from ARM Embedded
- MCUXpresso IDE v10.1.0

3 Supported Development Systems

This release supports boards and devices listed in this table.

Table 1. Supported MCU devices and development boards

Development boards	MCU devices
EVK-MIMXRT1050	MIMXRT1052DVL6A

4 Release Contents

This table provides an overview of the MCUXpresso SDK release package contents and locations.

Table 2. Release contents

Deliverable	**Location**
Boards	<install_dir>/boards
Demo applications	<install_dir>/boards/<board_name>/demo_apps
CMSIS driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
Driver examples	<install_dir>/boards/<board_name>/driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
EMWIN examples	<install_dir>/boards/<board_name>/emwin_examples
USB demo applications	<install_dir>/boards/<board_name>/usb_examples
Documentation	<install_dir>/docs
Middleware	<install_dir>/middleware
USB Documentation	<install_dir>/docs/usb
lwIP Documentation	<install_dir>/docs/lwip
SDMMC card driver	<install_dir>/middleware/sdmmc
lwIP stack	<install_dir>/middleware/lwip
USB stack	<install_dir>/middleware/usb
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source	<install_dir>/CMSIS
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools

5 MCUXpresso SDK Release Package

The MCUXpresso SDK release package contents are aligned with the silicon subfamily it supports. This includes the boards, CMSIS, devices, documentation, middleware, and RTOS support.

5.1 Device support

The device folder contains all available software enablement for the specific System-on-Chip (SoC) subfamily. This folder includes clock-specific implementation, device register header file, device register feature header file, CMSIS derived device SVD, and the system configuration source files. Included with the standard SoC support are folders containing peripheral drivers, toolchain support, and a simple debug console.

The device-specific header files provide a direct access to the MCU peripheral registers. The device header file provides an overall SoC memory mapped register definition. In addition to the overall device memory mapped header file, the 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.

5.1.1 Board support

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

5.1.2 Demo applications and other examples

The demo applications demonstrate the usage of the peripheral drivers to achieve a system level solution. Each demo application contains a readme file that describes the operation of the demo and required setup steps.

The driver examples 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.

5.2 Middleware

5.2.1 USB stack

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

5.2.1.1 Peripheral devices tested with the USB Host stack

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

Table 3. Peripheral devices

Device type	Device
USB HUB	BELKIN F5U233 BELKIN F5U304 BELKIN F5U307 BELKIN F4U040 UNITEK Y-2151 Z-TEK ZK032A HYUNDAI HY-HB608
USB flash drive	ADATA C008 32 GB ADATA S102 8 G ADATA S102 16 G Verbatim STORE N GO USB Device 8 G Kingston DataTraveler DT101 G2 SanDisk Cruzer Blade 8 GB Unisplendour 1 G Imation 2 GB V-mux 2 GB Sanmina-SCI 128 M Corporate Express 1 G TOSHIBA THUHYBS-008G 8 G Transcend JF700 8 G Netac U903 16 G SSK SFD205 8 GB Rex 4 GB SAMSUNG USB3.0 16GB
USB card reader/adapter	SSK TF adapter Kawau Multi Card Reader Kawau TF adapter Kawau SDHC card
USB Mouse	DELL MS111-P DELL M066U0A DELL MUAVDEL8 TARGUS AMU76AP DELL MD56U0 DELL MS111-T RAPOO M110
USB Keyboard	DELL SK8135 DELL SK8115

5.2.2 TCP/IP stack

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

5.2.3 Security libraries

The MCUXpresso SDK is integrated with mbedTLS and wolfSSL libraries. The integration demonstrates hardware acceleration of various cryptography algorithms and random number generation. The packages are available through separate add-on packages at mcuxpresso.nxp.com, Optional Middleware section.

5.2.4 RTOS

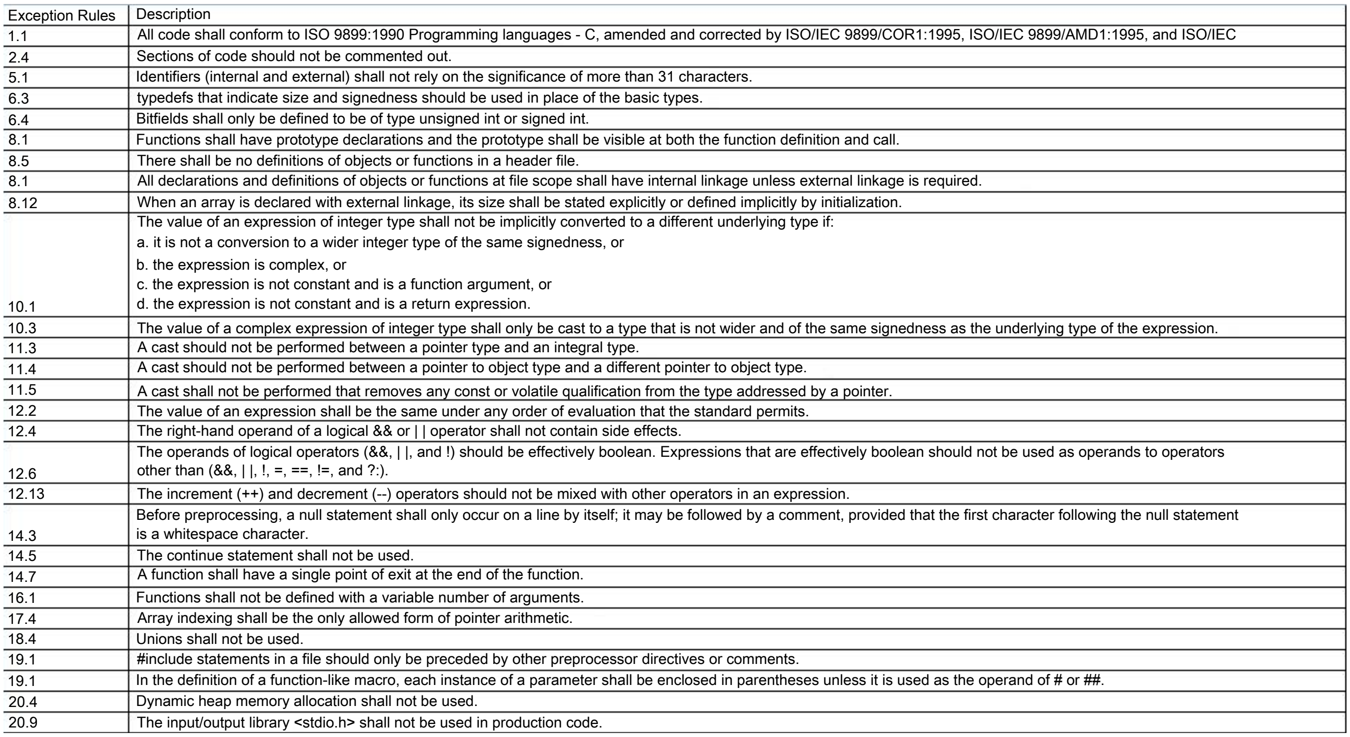
The MCUXpresso SDK is integrated with FreeRTOS OS.

5.2.5 CMSIS

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

6 MISRA Compliance

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



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

Figure 1. MISRA exceptions

7 Change Log - Peripheral drivers

ADC

Current ADC driver version is 2.0.0

- 2.0.0
  - Initial version.

ADC_ETC

Current ADC_ETC driver version is 2.0.0

- 2.0.0
  - Initial version.

AIPSTZ

Current AIPSTZ driver version is 2.0.0

- 2.0.0
  - Initial version.

AOI

Current AOI driver version is 2.0.0

- 2.0.0
  - Initial version.

BEE

Current BEE driver version is 2.0.0

- 2.0.0
    - Initial version.

CACHE

Current CACHE driver version is 2.0.1

- 2.0.0
    - Initial version.
- 2.0.1
    - Fix cache size issue in L2CACHE_GetDefaultConfig API.

CMP

Current CMP driver version is 2.0.0

- 2.0.0
    - Initial version.

CSI

Current PXP driver version is 2.0.0

- 2.0.0
    - Initial version.

DCDC

Current DCDC driver version is 2.0.0

- 2.0.0
    - Initial version.

DMAMUX

Current DMAMUX driver version is 2.0.2

- 2.0.0
    - Initial version.
- 2.0.1
    - Bug fix:
        - Fix build warning while setting DMA request source in DMAMUX_SetSourceChange issue, by changing the type of the parameter source from uint8_t to uint32_t.
- 2.0.2
    - New feature:
        - Add always on enable feature of a certain DMA channel for ULP1 DMAMUX support.

EDMA

Current EDMA driver version is 2.1.2

- 2.0.0
    - Initial version.
- 2.0.1
    - Bug fix:
        - Fix the eDMA callback does not check valid status issue in EDMA_HandleIRQ API.
- 2.0.2
    - Bug fix:
        - Fix incorrect minorLoopBytes type definition in _edma_transfer_config struct. Define minorLoopBytes as uint32_t instead of uint16_t.
- 2.0.3

  - Bug fix:
    - Fix the wrong pubweak IRQHandler name issue which will cause re-definition build errors when client sets his/her own IRQHandler, by changing the 32-channel IRQHandler name to DriverIRQHandler.
- 2.0.4
  - Improvement:
    - Add support for SoCs with multiple eDMA instances.
    - Add pubweak DriverIRQHandler for KL28T DMA1 and MCIMX7U5_M4.
- 2.0.5
  - Improvement:
    - Add pubweak DriverIRQHandler for K32H844P (16 channels shared).
- 2.1.0
  - Improvement:
    - Change the EDMA_GetRemainingBytes API into EDMA_GetRemainingMajorLoopCount, due to eDMA IP limitation (refer to the API comments/note for further details).
- 2.1.1
  - Improvement:
    - Add documentation of the eDMA data flow when scatter/gather is implemented for the EDMA_HandleIRQ API.
    - Update and correct some related comments in the EDMA_HandleIRQ API and edma_handle_t struct.
- 2.1.2
  - Improvement:
    - Add interface to get next TCD address.
    - Add interface to get the unused TCD number.

ELCDIF

Current ELCDIF driver version is 2.0.0

- 2.0.0
  - Initial version.

ENC

Current ENC driver version is 2.0.0

- 2.0.0
  - Initial version.

ENET

Current enet driver version is 2.2.1

- 2.0.0
  - Initial version.
- 2.0.1
  - Bug Fix:
    - Use direct transmit busy check when do data transmit.
  - Misc Changes:
    - Update IRQ handler work flow.
    - Change the tx/rx interrupt macro from kENET_RxByteInterrupt to kENET_RxBufferInterrupt, from kENET_TxByteInterrupt to kENET_TxBufferInterrupt.
    - Delete unnecessary parameter in ENET handler.
- 2.1.1
  - Add the extended MDIO IEEE802.3 Clause 45 MDIO format SMI command APIs
  - Add the extended interrupt coalescing feature.
  - combine all storage operations in the ENET_Init to ENET_SetHandler API
- 2.2.1
  - Change the input data pointer attribute to const in ENET_SendFrame()
- 2.2.2

    - Add the APIs for extended multi-ring support.
    - Add the avb configure API for extended avb feature support.

EWM

Current EWM driver version is 2.0.1

- 2.0.0
    - Initial version.
- 2.0.1
    - Fix EWM_Deinit hardfault issue.

FLEXCAN

Current FLEXCAN driver version is 2.2.0

- 2.0.0
    - Initial version.
- 2.1.0
    - Bug Fix:
        - Fix wrong function name spelling: FlEXCAN_XXX() -> FLEXCAN_XXX();
        - Move Freeze Enable/Disable setting from FLEXCAN_Enter/ExitFreezeMode() to FLEXCAN_Init();
        - Fix wrong helper macro values.
    - Other changes:
        - Hide FLEXCAN_Reset() to user.
        - Use NDEBUG macro to wrap FLEXCAN_IsMbOccupied() function instead of DEBUG macro
- 2.2.0
    - Improvement Add FSL_FEATURE_FLEXCAN_HAS_SUPPORT_ENGINE_CLK_SEL_REMOVE feature to support SoCs without CAN Engine Clock selection in FlexCAN module.
    - Add FlexCAN Serial Clock Operation to support i.MX SoCs.

FLEXRAM

Current FLEXRAM driver version is 2.0.1

- 2.0.0
    - Initial version.
- 2.0.1
    - Fix MISRA issue.

FLEXIO

Current FLEXIO driver version is 2.0.1

- 2.0.1
    - Bug fix:
        - Fix the Dozen mode configuration error in FLEXIO_Init API. For enableInDoze = true, the configuration shoud be 0; for enableInDoze = false, the configuration should be 1.

FLEXIO_UART

Current FLEXIO_UART driver version is 2.1.3

- 2.1.0
    - New Features:
        - Add Transfer prefix in transactional APIs
        - Add txSize/rxSize in handle structure to record the transfer size
    - Bug Fix:
        - Add error handle to handle the data count is zero or data buffer is NULL situation
- 2.1.1
    - Bug Fix:

        - Change the API name FLEXIO_UART_StopRingBuffer to FLEXIO_UART_TransferStopRingBuffer to align with the definition in C file
- 2.1.2
    - Bug Fix:
        - Fix the transfer count calculation issue in FLEXIO_UART_TransferGetReceiveCount, FLEXIO_UART_TransferGetSendCount, FLEXIO_UART_TransferGetReceiveCountDMA, FLEXIO_UART_TransferGetSendCountDMA, FLEXIO_UART_TransferGetReceiveCountEDMA and FLEXIO_UART_TransferGetSendCountEDMA
        - Fix the Dozen mode configuration error in FLEXIO_UART_Init API. For enableInDoze = true, the configuration shoud be 0; for enableInDoze = false, the configuration should be 1.
        - Report error when set baudrate too low and FLEXIO cannot reach that baudrate.
        - Disable FLEXIO_UART receive interrupt instead of disable all NVIC when read data from ring buffer. Because with ring buffer used, receive nonblocking will disable all NVIC interrupts to protect the ring buffer, this will have negative effect to other IPS which are using interrupt.
- 2.1.3
    - Bug Fix: Following modification to support FlexIO using multiple instances.
        - Remove FLEXIO_Reset API in module Init APIs
        - Update module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock
        - Update module Enable APIs to only support enable operation

FLEXIO_I2C

Current FLEXIO_I2C driver version is 2.1.4

- 2.1.0
    - New Features:
        - Add Transfer prefix in transactional APIs
        - Add transferSize in handle structure to record the transfer size
- 2.1.1
    - Bug Fix:
        - Implement the FLEXIO_I2C_MasterTransferBlocking API which defined in header flie but has no implementation in C file
- 2.1.2
    - Fix the FLEXIO I2C master can not receive data from i2c slave in high baudrate issue
    - Fix the FLEXIO I2C master can not receive NAK when master send non exist addr issue
    - Fix the FLEXIO I2C master can not get transfer count successfully issue
    - Fix the FLEXIO I2C master can not receive data successfully when send data first issue
    - Fix the Dozen mode configuration error in FLEXIO_I2C_MasterInit API. For enableInDoze = true, the configuration shoud be 0; for enableInDoze = false, the configuration should be 1.
    - Fix the FLEXIO_I2C_MasterTransferBlocking API calls FLEXIO_I2C_MasterTransferCreateHandle issue, this leads the s_flexioHandle/s_flexioIsr/s_flexioType variable written, then if call FLEXIO_I2C_MasterTransferBlocking API multiple times, the s_flexioHandle/s_flexioIsr/s_flexioType variable cannot be written anymore due to out of range, this will lead the following NonBlocking transfer APIs can not work due to register IRQ failed.
- 2.1.3
    - Change the prototype of FLEXIO_I2C_MasterInit to return kStatus_Success if initialization successfully and return kStatus_InvalidArgument if "(srcClock_Hz / masterConfig->baudRate_Bps) / 2 - 1" exceeds 0xFFU.
- 2.1.4
    - Bug Fix: Following modification to support FlexIO using multiple instances.
        - Remove FLEXIO_Reset API in module Init APIs
        - Update module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock
        - Update module Enable APIs to only support enable operation

FLEXIO_SPI

Current FLEXIO_SPI driver version is 2.1.2

- 2.1.0

  - New Features:
    - Add Transfer prefix in transactional APIs
    - Add transferSize in handle structure to record the transfer size
  - Bug Fix:
    - Fix the error register address return for 16-bit data write in FLEXIO_SPI_GetTxDataRegisterAddress
    - Provide independent IRQHandler/transfer APIs for Master and slave to fix the baudrate limit issue
- 2.1.1
  - Bug Fix:
    - Fix the bug when FLEXIO SPI transfer data in 16 bit per frame mode with edma
    - Fix the bug when FLEXIO SPI transfer data in 16 bit per frame and direction is Lsbfirst mode with edma and interrupt
    - Fix the Dozen mode configuration error in FLEXIO_SPI_MasterInit/FLEXIO_SPI_SlaveInit API. For enableInDoze = true, the configuration shoud be 0; for enableInDoze = false, the configuration should be 1.
  - Optimization:
    - Add #ifndef/#endif to allow user to change the default tx value at compile time.
- 2.1.2
  - Bug Fix: Following modification to support FlexIO using multiple instances.
    - Remove FLEXIO_Reset API in module Init APIs
    - Update module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock
    - Update module Enable APIs to only support enable operation

FLEXIO_I2S

Current FLEXIO_I2S driver version is 2.1.3

- 2.1.0
  - New Features:
    - Add Transfer prefix in transactional APIs
    - Add transferSize in handle structure to record the transfer size
- 2.1.1
  - Bug Fix:
    - Fix flexio i2s rx data read error and edma address error.
    - Fix flexio i2s slave timer compare setting error.
- 2.1.2
  - New Features:
    - Add configure items for all pin polarity and data valid polarity.
    - Add default configure for pin polarity and data valid polarity.
- 2.1.3
  - Bug Fix: Following modification to support FlexIO using multiple instances.
    - Remove FLEXIO_Reset API in module Init APIs
    - Update module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock
    - Update module Enable APIs to only support enable operation

FLEXIO_MCU_LCD

Current FLEXIO_MCU_LCD driver version is 2.0.1

- 2.0.0
  - Initial version.
- 2.0.1
  - Bug Fix: Following modification to support FlexIO using multiple instances.
    - Remove FLEXIO_Reset API in module Init APIs
    - Update module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock
    - Update module Enable APIs to only support enable operation

FLEXIO_CAMERA

Current FLEXIO_CAMERA driver version is 2.0.1

- 2.0.0
  - Initial version.
- 2.0.1
  - Bug Fix: Following modification to support FlexIO using multiple instances.
    - Remove FLEXIO_Reset API in module Init APIs
    - Update module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock
    - Update module Enable APIs to only support enable operation

FLEXSPI

Current FLEXSPI driver version is 2.0.1

- 2.0.0
  - Initial version.
- 2.0.1
  - Bug Fix:
    - Fix the flag clear issue and AHB read Command index configuration issue in FLEXSPI_SetFlashConfig
    - Update FLEXSPI_UpdateLUT function to update LUT table from any index instead of previous command index
    - Add bus idle wait in FLEXSPI_SetFlashConfig and FLEXSPI_UpdateLUT to ensure bus is idle before any change to flexspi controller
    - Update interrupt API FLEXSPI_TransferNonBlocking and interrupt handle flow FLEXSPI_TransferHandleIRQ.
    - Update edma API FLEXSPI_TransferEDMA

GPC

Current GPC driver version is 2.0.0

- 2.0.0
  - Initial version.

GPIO

Current GPIO driver version is 2.0.0

- 2.0.0
  - Initial version.
- 2.0.1:
  - API Interface Change
    - Refine naming of API while keep all original APIs with marking them as deprecated. Original API will be removed in next release. The mainin change is update API with prefix of _PinXXX() and _PorortXXX

GPT

Current gpt driver version is 2.0.0

- 2.0.0
  - Initial version.

KPP

Current kpp driver version is 2.0.0

- 2.0.0
  - Initial version.

LPI2C

Current LPI2C driver version is 2.1.1

- 2.0.0

    - Initial version.
- 2.1.0
    - API name change:
        - LPI2C_MasterTransferCreateHandle -> LPI2C_MasterCreateHandle
        - LPI2C_MasterTransferGetCount -> LPI2C_MasterGetTransferCount
        - LPI2C_MasterTransferAbort -> LPI2C_MasterAbortTransfer
        - LPI2C_MasterTransferHandleIRQ -> LPI2C_MasterHandleInterrupt
        - LPI2C_SlaveTransferCreateHandle -> LPI2C_SlaveCreateHandle
        - LPI2C_SlaveTransferGetCount -> LPI2C_SlaveGetTransferCount
        - LPI2C_SlaveTransferAbort -> LPI2C_SlaveAbortTransfer
        - LPI2C_SlaveTransferHandleIRQ -> LPI2C_SlaveHandleInterrupt
- 2.1.1
    - Bug fix:
        - Disable auto stop feature in EDMA driver, previously the autostop feature is enabled at transfer when transfer with stop flag. If previous transfer without stop flag, then when start a new transfer with stop flag, because the auto stop feature is enabled, so the stop flag will be sent before starting the new transfer and the start flag can not successfully sent, so the transfer can not start.
        - Change default slave configuration with address stall false.
- 2.1.2
    - Bug fix:
        - In LPI2C_SlaveTransferHandleIRQ, reset the slave status to idle when stop flag is detected.
- 2.1.3
    - Improvement:
        - Add LPI2C_WATI_TIMEOUT macro to allow user to specify the timeout times for waiting flags in functional API and blocking transfer API.
        - Add LPI2C_MasterTransferBlocking API
- 2.1.4
    - Bug fix:
        - Fix the LPI2C_MasterTransferEDMA receive issue when LPI2C share same request source for tx/rx dma request. In previous way the API uses scatter gather method, handle command transfer first, then handle the linked tcd which preset with the receive data transfer. The issue is that tx DMA request and rx DMA request are both enabled, when DMA finished the first command tcd transfer and handle the receive data tcd, the tx DMA request still happens due to tx fifo empty, this results the rx DMA transfer starts, without waiting on the expected rx DMA request. Fix the issue by enable IntMajor interrupt for the command tcd and check if there's linked tcd to disable the tx DMA request in LPI2C_MasterEDMACallback API.

LPSPI

Current lpspi driver version is 2.0.2

- 2.0.0
    - Initial version.
- 2.0.1
    - Bug Fix:
        - The clock source should divided by PRESCALE setting in LPSPI_MasterSetDelayTimes function.
        - Fix the bug that LPSPI_MasterTransferBlocking function would hang in some corner cases.
    - Optimization:
        - Add #ifndef/#endif to allow user to change the default tx value at compile time.
- 2.0.2
    - New feature:
        - Add dummy data setup API to allow users to configure the dummy data to be transferred.
        - Enable the 3-wire mode, SIN and SOUT pins can be configured as input/output pin.

LPUART

Current LPUART driver version is 2.2.5

- 2.1.0

    - Update transactional APIs.
- 2.1.1
    - Remove needless check of event flags and assert in LPUART_RTOS_Receive.
    - Wait always for rx event flag in LPUART_RTOS_Receive.
- 2.2.0
    - Add seven data bits and msb support
- 2.2.1
    - Add seprarte rx,tx irq number support
- 2.2.2
    - Add software reset feature support.
    - Add software reset API to LPUART_Init().
- 2.2.3
    - Change parameter type in LPUART_RTOS_Init() struct rtos_lpuart_config –> lpuart_rtos_config_t.
    - Bug fixed:
        - Disable LPUART receive interrupt instead of disable all NVIC when read data from ring buffer. Because with ring buffer used, receive nonblocking will disable all NVIC interrupts to protect the ring buffer, this will have negative effect to other IPS which are using interrupt.
- 2.2.4
    - Add hardware flow control function support.
    - Add idle line detected feature in LPUART_TransferNonBlocking function, if an idle line was detected, a callback will be triggered with status kStatus_LPUART_IdleLineDetected returned. This feature may be usefull when the received bytes is less than the expected receive data size. Before triggering the callback, data in the FIFO will be read out(if has FIFO), and all interrupt will not be disabled except the receive data size reach 0.
    - Enable the RX FIFO watermark function, with the idle line detected feature enabled, you can set the watermark value to whatever you want(should less than the RX FIFO size), data will be received and a callback will be triggered when data receive is end.
- 2.2.5
    - not set or clear the TIE/RIE bits when using LPUART_EnableTxDMA() and LPUART_EnableRxDMA().

PIT

Current PIT driver version is 2.0.0

- 2.0.0
    - Initial version.

PMU

Current PMU driver version is 2.1.0

- 2.0.0
    - Initial version.
- 2.1.0
    - Add feature macros for low power control APIs to support to conditional compile.
    - Renames "PMU_2P1EnablePullDown" to "PMU_2P5EnablePullDown"

PWM

Current PWM driver version is 2.0.0

- 2.0.0
    - Initial version.

PXP

Current PXP driver version is 2.0.0

- 2.0.0
    - Initial version.

RTWDOG

Current RTWDOG driver version is 2.0.0

- 2.0.0
  - Initial version.

QTMR

Current QTMR driver version is 2.0.0

- 2.0.0
  - Initial version.

SAI

Current SAI driver version is 2.1.2

- 2.0.0
  - Initial version.
- 2.1.0
  - API name change:
    - SAI_GetSendRemainingBytes -> SAI_GetSentCount
    - SAI_GetReceiveRemainingBytes -> SAI_GetReceivedCount
    - All transcational API name add "Transfer" prefix.
    - All transactional API use base and handle as input parameter.
    - Unify the parameter names.
  - Bug fix:
    - Fix w1c bug while reading TCSR/RCSR registers.
    - Fix MOE enable flow issue, move MOE enable after MICS settings in SAI_TxInit/SAI_RxInit.
- 2.1.1
  - Optimization:
    - Reduce code size while not using transactional API.
- 2.1.2
  - Bug fix:
    - Add 24-bit support for SAI EDMA transfer. All data shall be 32 bits for send/receive, as EDMA cannot directly handle 3 byte transfer.
- 2.1.3
  - New feature:
    - Add feature to make I2S frame sync length configurable according to bitWidth.

SEMC

Current semc driver version is 2.0.0

- 2.0.0
  - Initial version.

SPDIF

Current SPDIF driver version is 2.0.0

- 2.0.0
  - Initial version.

SRC

Current SRC driver version is 2.0.0

- 2.0.0
  - Initial version.

TSC

Current TSC driver version is 2.0.0

- 2.0.0
    - Initial version.
    - This module is developed firstly on I.mx6ull.

USDHC

Current USDHC driver version is 2.2.1

- 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
- 2.1.2
    - Fix Coverity issue.
    - Add base address and userData parameter for all callback function.
- 2.1.3
    - Fix MISRA issue.
- 2.2.0 -Improve usdhc to support mmc boot feature.
- 2.2.1 -Disable the invalidate cache operation for tuning.

WDOG

Current WDOG driver version is 2.0.0

- 2.0.0
    - Initial version.

XBARA

Current XBARA driver version is 2.0.3

- 2.0.0
    - Initial version.
- 2.0.1
    - Bug Fix:
        - Fix w1c bits for XBARA_SetOutputSignalConfig function.
- 2.0.2
    - Other changes:
        - Change array clock name.
- 2.0.3
    - Bug Fix:
        - Correct configuration for function XBAR_SetOutputSignalConfig.

XBARB

Current XBARB driver version is 2.0.1

- 2.0.0

    - Initial version.
- 2.0.1
    - Bug Fix:
        - Correct XBARB_SetSignalsConnection function.
    - Other changes:
        - Change array clock name.

8 Change Log - Middleware

emWin library for KSDK

Currently supported version is 5.38a

FatFs for KSDK

Current version is FatFs R0.12c.

- R0.12c_rev0
    - Upgrade to version 0.12c and apply patches ff_12c_p1.diff and ff_12c_p2.diff.
- R0.12b_rev0
    - Upgrade to version 0.12b.
- R0.11a
    - Add glue functions for low level drivers (SDHC, SDSPI, RAM, MMC). Modified diskio.c
    - Add RTOS wrappers to make FatFs thread safe. Modified syscall.c
    - Rename of ffconf.h to ffconf_template.h. Each application should contain its own ffconf.h.
    - Include ffconf.h into diskio.c to enable selection of physical disk from ffconf.h by macro definition.
    - Conditional compilation of physical disk interfaces in diskio.c

LigJpeg for KSDK

Current version is LigJpeg 9b.

lwIP for KSDK

Current version of lwIP for KSDK is based on lwIP 2.0.2 (2017-03-13, SHA-1: c0862d60746e2d1ceae69af4c6f24e469570ecef).

- 2.0.2_rev1
    - New Features:
        - Ported lwIP 2.0.2 (2017-03-13, SHA-1: c0862d60746e2d1ceae69af4c6f24e469570ecef) to KSDK 2.0.0.
- 2.0.0_rev3
    - New Features:
        - Ported lwIP 2.0.0 (2016-11-10, SHA-1: 216bf89491815029aa15463a18744afa04df58fe) to KSDK 2.0.0.
- 2.0.0_rev2
    - New Features:
        - Ported lwIP 2.0.0 RC2 (2016-08-08, SHA-1: b1dfd00f9233d124514a36a8c8606990016f2ad4) to KSDK 2.0.0.
- 2.0.0_rev1
    - New Features:
        - Ported lwIP 2.0.0 RC0 (2016-05-26) to KSDK 2.0.0.
        - Changed lwIP bare-metal examples to use poll-driven approach instead of interrupt-driven one.
- 1.4.1_rev2
    - New Features:
        - Enabled critical sections in lwIP.
    - Bug Fix:

        - Fixed default lwIP packet-buffer size to be able to accept a maximum size frame from the ENET driver.
        - Fixed possible drop of multi-frame packets during transmission.
- 1.4.1_rev1
    - New Features:
        - Ported lwIP 1.4.1 to KSDK 2.0.0.

mbedTLS for KSDK

Current version of mbedTLS for KSDK is based on mbedTLS 2.6.0 released 2017-Aug-10

- 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:
        - Add support for DCP driver.
- 2.5.1
    - New Features:
        - Ported mbedTLS 2.5.1 to KSDK
- 2.4.2_rev2
    - New Features:
        - Add Curve25519 support for CAU3.
        - Added MBEDTLS_ECP_MUL_MXZ_ALT configuration parameter enabling overloading of ecp_mul_mxz().
- 2.4.2_rev1
    - New Features:
        - Add 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:
        - Add 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 for KSDK

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 maximum data bus width for MMC card
- 2.1.1

    - Bug Fix:
        - Fix the block range boundary 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 include different adaptor code to make application simpler.
        - Adaptor code provides HOST_Init/HOST_Deinit/CardInsertDetect APIs to do host controller initialize and transfer function configuration. SDMMC card stack uses adaptor code inside stack to wait card insert and configure host when calling card init APIs (SD_Init/MMC_Init/SDIO_Init).
        - So this change requires user to include host adaptor code into application. If not, link errors for cannot find the definition of HOST_Init/HOST_Deinit/CardInsertDetect will appear.
    - New feature: Improve sdmmc to support SD v3.0 and emmc v5.0
    - Bug Fix:
        - Fix Wrong comparison between count and length in MMC_ReadBlocks/MMC_WriteBlocks
- 2.1.3
    - Bug fix:
        - Non high speed 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 improve compatibility.
    - Bug fix:
        - Fix card cannot detect dynamically.
- 2.1.5
    - Fix coverity issue.
    - Fix SD v1.x card write fail issue, it was caused by the block length set error.
    - Improve SDIO card init sequence and add retry option for SDIO_SwitchToHighSpeed function.
- 2.1.6
    - Enhance SD IO default driver strength.
- 2.2.0
    - New feature:
        - Separate the SD/MMC/SDIO init API to xxx_CardInit/xxx_HostInit.
        - Allow user register card detect callback, select card detect type, determine the card detect timeout value.
        - Allow user register the power on/off function, determine the power on/off delay time.
        - SD_Init/SDIO_Init will be deprecated in next version.
        - Add write complete wait operation for MMC_Write to fix command timeout issue.
- 2.2.1
    - Improve mmc boot feature.
    - Keep SD_Init/SDIO_Init function for forward compatibility.
- 2.2.2
    - Move set card detect priority operation before enable IRQ.

USB stack for KSDK

Current version of USB stack is 1.7.0

- 1.0.0
    - New Features:
        - Supported roles
            - Device
            - Host
        - Supported controllers
            - KHCI (Full Speed)
            - EHCI (High Speed)
        - Supported classes
            - AUDIO
            - CCID
            - CDC
            - HID
            - MSC
            - PHDC
            - VIDEO
        - Examples
            - usb_device_audio_generator
            - usb_device_audio_speaker
            - usb_device_ccid_smart_card
            - usb_device_cdc_vcom
            - usb_device_cdc_vnic
            - usb_device_composite_cdc_msc
            - usb_device_composite_hid_audio
            - usb_device_composite_hid_mouse_hid_keyboard
            - 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
- 1.0.1
    - Bug Fix:
        - Improve the efficiency of device audio speaker by changing the transfer mode from interrupt to dma thus can eliminate the periodic noise.
- 1.1.0
    - Bug Fix:
        - Fix some issues in USB certification.
        - Change 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
- 1.2.0
    - New Features:
        - Supported controllers
            - LPC IP3511 (Full Speed, Device mode)
- 1.3.0
    - New Features:
        - Supported roles
            - OTG
        - Supported classes
            - CDC RNDIS
        - Examples
            - usb_otg_hid_mouse
            - usb_device_cdc_vnic
            - usb_suspend_resume_device_hid_mouse
            - usb_suspend_resume_host_hid_mouse
- 1.4.0
    - New Features:
        - Examples
            - usb_device_hid_mouse/freertos_static
            - usb_suspend_resume_device_hid_mouse_lite
- 1.5.0
    - New Features:
        - Supported controllers
            - OHCI (Full Speed, Host mode)
            - IP3516 (High Speed, Host mode)
            - IP3511 (High Speed, Device mode)
        - Examples
            - usb_lpm_device_hid_mouse
            - usb_lpm_device_hid_mouse_lite
            - usb_lpm_host_hid_mouse
- 1.6.0
    - New Features:
        - Supported Device Charger Detect feature on usb_device_hid_mouse
- 1.6.1

    - New Features:
        - Change the struct variable address method for device_video_virtual_camera and host_phdc_manager

  1.6.2

    - New Features:
        - Multi instance support

  1.6.3

    - Bug Fix: -IP3511_HS driver control transfer sequence issue, enbale 3511 ip cv test.

  1.7.0

    - New Features:
        - USB PD stack support.
    - Examples
        - usb_pd
        - usb_pd_battery
        - usb_pd_source_charger

QCA WiFi for KSDK

Current version is 2.0.0.

- 2.0.0
  - Notes:
    - Add 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.

wolfSSL for KSDK

Current version is 3.9.8_rev3, based on Release 3.9.8 of wolfSSL.

- 3.8.0
  - New Features:
    - Added support for Kinetis LTC hardware acceleration module. Accelerates AES, 3DES, TFM module (modular integer arithmetic) and ECC wolfSSL modules.
    - Added support for Kinetis random number generator modules TRNG and RNGA.
  - Other changes:
    - The Kinetis MMCAU acceleration now uses "fsl_mmcau.h" instead of "cau_api.h".
    - In DSA, wc_dsaSign() changed to repeate wc_RNG_GenerateBlock() until k is less than q.
    - wolfssl/wolfcrypt/settings.h is changed to remove unused macros and add support for KSDK 2.0.
    - In wolfcrypt/src/asn.c, ksdk_time(time_t) changed to extern, to be defined by application.
- 3.9.0
  - New Features:
    - Added more LTC public key acceleration (curve25519, ed25519 and RSA4096)
    - FREESCALE_LTC_TFM_RSA_4096_ENABLE macro added to enable RSA4096 on K8x/KL8x LTC
    - LTC_MAX_ECC_BITS increased to 384 to enable ECC-384 curve acceleration on LTC
    - FREESCALE_LTC_SHA added for KL8x SHA-1 and SHA-256 hardware acceleration
  - Other changes:
    - wolfssl/wolfcrypt/settings.h is changed to remove unused macros and add support for KSDK 2.0.
    - LTC public key acceleration is implemented in separate source file ksdk_port.h and ksdk_port.c
- 3.9.8
  - New Features:
    - Added support for AES and SHA acceleration modules of LPC devices. Accelerates AES and SHA wolfSSL modules.
    - LTC acceleration for AES CBC now updates IV
  - Bug fixes:
    - fixed K8x/KL8x LTC RSA sign when FREESCALE_LTC_TFM_RSA_4096_ENABLE macro is enabled.
- 3.9.8_rev1
  - New Features:
    - Add support for CAAM driver.
    - Add FREESCALE_ALT macros.
- 3.9.8_rev2
  - New Features:
    - Add support for CAU3 driver.
- 3.9.8_rev3
  - New Features:
    - Add support for DCP driver.

9 Change Log - RTOS

FreeRTOS for KSDK

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.
    - Enable float context saving in IAR for Cortex-A7. Add configUSE_TASK_FPU_SUPPORT macros. Modify port.c and portmacro.h in portable/IAR/ARM_CA9 folder.
  - Other changes:
    - Transform ARM_CM core specific tickless low power support into generic form under freertos.
- 9.0.0_rev2
  - New Features:
    - Enable 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:
    - Enable -flto optimization in GCC by adding

      attribute

      ((used)) for vTaskSwitchContext.
    - Enable KDS Task Aware Debugger. Apply FreeRTOS patch to enable configRECORD_STACK_HIGH_ADDRESS macro. Modified files are task.c and FreeRTOS.h.
- 9.0.0_rev0
  - New Features:
    - Example freertos_sem_static.
    - Static allocation support RTOS driver wrappers.
  - Other changes:
    - Tickless idle rework. Support for different timers is in separated files (fsl_tickless_systick.c, fsl_tickless_lptmr.c).
    - Remove configuration option configSYSTICK_USE_LOW_POWER_TIMER. Low power timer is now selected by linking of apropriate file fsl_tickless_lptmr.c.
    - Remove configOVERRIDE_DEFAULT_TICK_CONFIGURATION in RVDS port. Use of

      attribute

      ((weak)) is preffered solution. Not same as _week!
- 8.2.3
  - New Features:
    - Tickles idle mode support.
    - Add 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

Document Number MCUXSDKMIMXRT105XRN
Revision 0, 11/2017