NXP Semiconductors
Release Notes

Document Number: KSDK200KV58FRN
Rev. 0, 07/2016

Kinetis SDK v.2.0.0 Release Notes Supporting TWR-KV58F220M

1 Overview

The Kinetis Software Development Kit (KSDK) 2.0.0 is a collection of software enablement for Kinetis Microcontrollers that includes peripheral drivers, high-level stacks including LWIP, integration with WolfSSL and mbed TLS cryptography libraries, other middleware packages, such as multicore support and FatFs, and integrated RTOS support for FreeRTOS OS and µC/OS. In addition to the base enablement, the KSDK is augmented with demo applications and driver example projects, and API documentation to help the customers quickly leverage the support of the Kinetis SDK.

For the latest version of this and other Kinetis SDK documents, see the Kinetis SDK homepage KINETIS-SDK: Software Development Kit for Kinetis MCUs.

Contents

1 Overview............................. 1
2 KSDK 2.0.0............................ 1
3 Development Tools..................... 2
4 Supported Development Systems.... 2
5 Release Contents..................... 2
6 Kinetis SDK Release Package.................................3
7 MISRA Compliance.................. 4
8 Known Issues... 5
9 Driver Log............................ 6
10 Middleware Log....................... 11
11 RTOS Log............................ 14
12 Revision History.................... 15

2 KSDK 2.0.0

KSDK 2.0.0 is the evolution of KSDK 1.x into a more optimized software solution. KSDK 2.0.0 eliminates the need for a separate HAL and Peripheral Driver, replacing these two layers with a single driver for each peripheral. The single driver provides both the low-level functionality of the HAL and the non-blocking interrupt-based functionality of the Peripheral Driver, enabling customers to select the right level of abstraction for their solution. Peripheral drivers in KSDK

2.0.0 also eliminate external software dependencies. The Operating System Abstraction, Power Manager, and Clock Manager are no longer required by the KSDK 2.0.0 drivers.

The existing MQX™ RTOS support has been deprecated to focus on support of FreeRTOS OS, µC/OS-II, and µC/OS-III.

The Real Time Control Embedded Software Library (RTCESL) and motor control examples for PMSM and BLDC are added to the middleware layer.

3 Development Tools

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

- Kinetis Design Studio IDE v3.2
- IAR Embedded Workbench for ARM version 7.60
- MDK-ARM Microcontroller Development Kit (Keil)® 5.18a
- Makefiles support with GCC revision 5-2015-q3-update from ARM Embedded
- Atollic® TrueSTUDIO® 5.5.1

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	Kinetis MCU devices
TWR-KV58F220M	MKV56F1M0VLL24, MKV56F1M0VLQ24, MKV56F512VLL24, MKV56F512VLQ24, MKV58F1M0VLL24, **MKV58F1M0VLQ24**, MKV58F512VLL24, MKV58F512VLQ24, MKV56F1M0VMD24 MKV56F512VMD24, MKV58F1M0VMD24, MKV58F512VMD24

5 Release Contents

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

Table 2. Release contents

Deliverable	**Location**
Boards	<install_dir>/boards
Demo applications	<install_dir>/boards/<board_name>/demo_apps
Driver examples	<install_dir>/boards/<board_name>/driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Documentation	<install_dir>/docs
lwIP Documentation	<install_dir>/docs/lwip

Table continues on the next page...

Table 2. Release contents (continued)

Middleware	<install_dir>/middleware
lwIP stack	<install_dir>/middleware/lwip_<version>
DMA manager	<install_dir>/middleware/dma_manager_<version>
mmCAU	<install_dir>/middleware/mmcau_<version>
Motor Control libraries	<install_dir>/middleware/motor_control_<version>
RTCESL libraries	<install_dir>/middleware/rtcesl_<version>
SDMMC card driver	<install_dir>/middleware/sdmmc_<version>
WolfSSL stack	<install_dir>/middleware/wolfssl_<version>
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source	<install_dir>/CMSIS
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools

6 Kinetis SDK Release Package

The KSDK 2.0.0 release package contents are aligned with the silicon subfamily it supports. This includes the boards, CMSIS, devices, documentation, middleware, and RTOS support.

6.1 Kinetis device support

The device folder contains all available software enablement for the specific 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 Kinetis MCU peripheral registers. The device header file provides an overall System-on-Chip (SoC) memory mapped register definition. In addition to the overall device memory mapped header file, the Kinetis 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 Kinetis 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 TCP/IP stack

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

6.2.2 RTOS

The Kinetis SDK is preintegrated with FreeRTOS OS, µC/OS-II OS, and µC/OS-III OS.

6.2.3 CMSIS

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

6.2.4 Real Time Control Embedded Software Library (RTCESL)

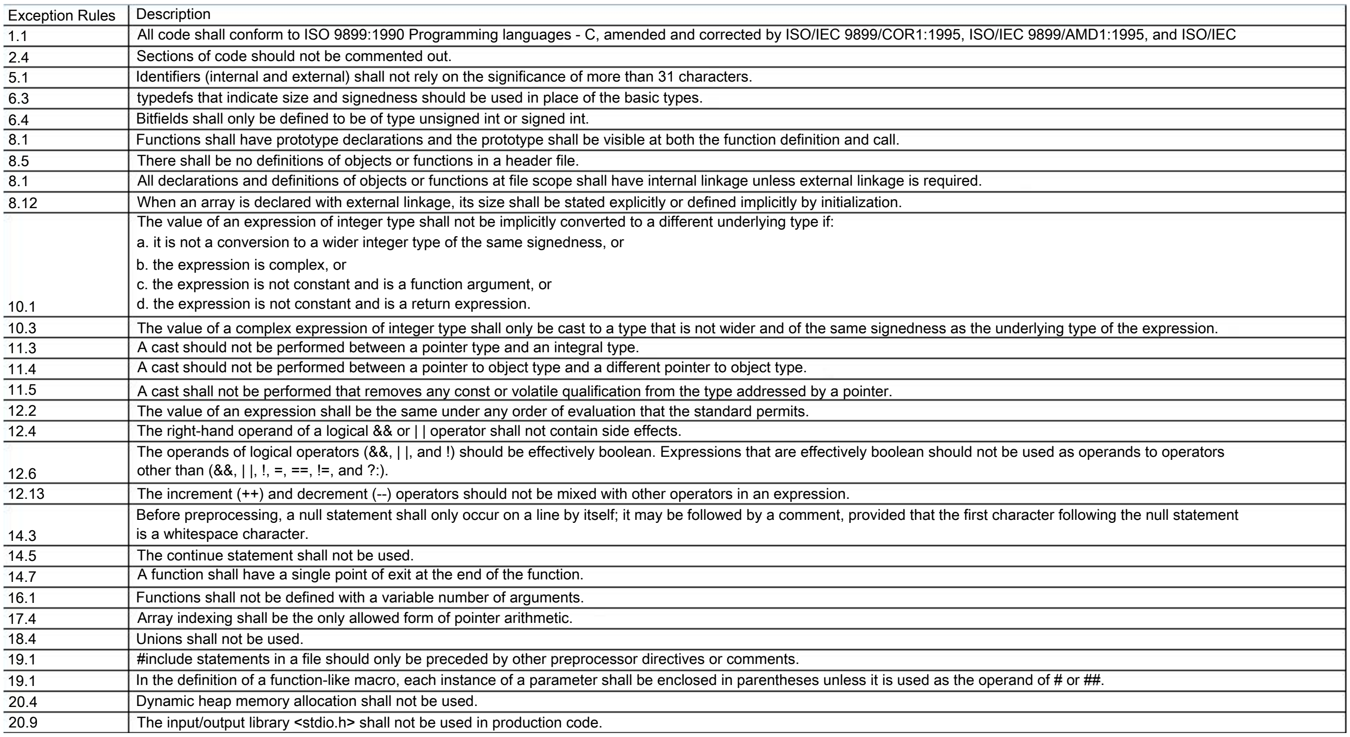
RTCESL contains a set of functions for the real time control. See documentation in <install_dir>/middleware/rtcesl_<version>.

6.2.5 Motor control examples

Motor control examples include examples for PMSM and the BLDC control. See the motor control documentation in <install_dir>/docs/MC.

7 MISRA Compliance

All KSDK 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

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 Kinetis 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 MC examples known issue

The supported IDEs are IAR/KDS/KEIL IDEs only. Attolic TrueSTUDIO and ARMGCC IDEs are not supported.

8.3 Atollic TrueSTUDIO and KDS IDE FPU support limitation

In KDS and Atollic TrueSTUDIO IDEs, the FPU for TWR-KV58F220M does not support selecting FPV5. Because armgcc does not support the *-mfpu* option, use FPV4 to as a workaround.

9 Driver Log

ADC16

The current ADC16 driver version is 2.0.0

- 2.0.0
  - Initial version

AOI

The current AOI driver version is 2.0.0

- 2.0.0
  - Initial version

CMP

The current CMP driver version is 2.0.0

- 2.0.0
  - Initial version

DAC

The current DAC driver version is 2.0.1

- 2.0.0
  - Initial version
- 2.0.1
  - Bug fix:
    - Moved the default DAC_Enable(..., true) from the DAC_Init() to the application code to enable the DAC output

DMAMUX

The current DMAMUX driver version is 2.0.2

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

DSPI

The current DSPI driver version is 2.1.2

- 2.1.0
  - New features
    - Added transfer prefix to transactional APIs
- 2.1.1
  - Bug fix:
    - Set the EOQ (End Of Queue) bit to TRUE for the last transfer in transactional APIs
- 2.1.2

- Bug fix:
    - The DSPI_MasterTransferBlocking function hangs in corner cases, for example, when bitsPerFrame is 4 and 6 and in the kDSPI_MasterPcsContinuous transfer mode

eDMA

The current eDMA driver version is 2.0.3

- 2.0.0
    - Initial version
- 2.0.1
    - Bug fix:
        - Fixed the issue where an eDMA callback does not check a valid status in the EDMA_HandleIRQ API
- 2.0.2
    - Bug fix:
        - Fixed the incorrect minorLoopBytes type definition in the _edma_transfer_config structure. Defined the minorLoopBytes as uint32_t instead of uint16_t
- 2.0.3
    - Bug fix:
        - Fixed the incorrect pubweak IRQHandler name issue, which causes re-definition build errors when a client sets his/her own IRQHandler, by changing the 32-channel IRQHandler name to DriverIRQHandler

ENC

The current ENC driver version is 2.0.0

- 2.0.0
    - Initial version

ENET

The current ENET driver version is 2.0.1

- 2.0.0
    - Initial version
- 2.0.1
    - Bug fix:
        - Used the direct transmit busy check during data transmission
    - Miscellaneous changes:
        - Updated the IRQ handler workflow
        - Changed the TX/RX interrupt macro from kENET_RxByteInterrupt to kENET_RxBufferInterrupt and from kENET_TxByteInterrupt to kENET_TxBufferInterrupt
        - Deleted unnecessary parameters in the ENET handler

Flash

The current Flash driver version is 2.1.0

- 2.0.0
    - Initial version
- 2.1.0
    - New features:
        - Support for the FTFL device in FLASH_Swap API
        - Support for various pflash start addresses
        - Added support for KV58 in the cache clear function
    - Bug fix
        - Compiled execute-in-RAM functions as a PIC binary code for driver use
        - Added missed FlexRAM properties
        - Fixed an unaligned variable issue for the execute-in-RAM function code array

FlexCAN

The current FlexCAN driver version is 2.1.0

- 2.0.0
    - Initial version
- 2.1.0
    - Bug fix:
        - Fixed the incorrect function name spelling FlEXCAN_XXX() -> FLEXCAN_XXX();
        - Moved the Freeze Enable/Disable setting from FLEXCAN_Enter/ExitFreezeMode() to FLEXCAN_Init();
        - Fixed the incorrect helper macro values
    - Miscellaneous changes:
        - Hid FLEXCAN_Reset() from the user
        - Used the NDEBUG macro to wrap the FLEXCAN_IsMbOccupied() function instead of a DEBUG macro

FTM

The current FTM driver version is 2.0.1

- 2.0.0
    - Initial version
- 2.0.1
    - Bug fix:
        - Updated the FTM driver to fix write to ELSA and ELSB bits
        - Set the COMBINE bit before writing to the CnV register

GPIO

The current GPIO driver version is 2.1.0

- 2.1.0
    - API Interface Change:
        - Added "pins" or "pin" to some API names
        - Renamed the "GPIO_PinConfigure" to "GPIO_PinInit"

HSADC

The current HSADC driver version is 2.0.0

- 2.0.0
    - Initial version

I2C

The current I2C driver version is 2.0.1

- 2.0.1
    - New features
        - Added a double buffer enable configuration for SoCs which have the DFEN bit in S2 register
        - Added the flexible transmit/receive buffer size support in I2C_SlaveHandleIRQ
        - Added the start flag clear address match and release bus operation in I2C_SlaveWrite/ReadBlocking API
    - Bug fix:
        - Updated the kI2C_SlaveRepeatedStartEvent to kI2C_SlaveStartEvent

LLWU

The current LLWU driver version is 2.0.1

- 2.0.0
    - Initial version
- 2.0.1
    - Changes:

        - Updated for KL8x

LPTMR

The current LPTMR driver version is 2.0.0

- 2.0.0
    - Initial version

MMDVSQ

The current MMDVSQ driver version is 2.0.2

- 2.0.0
    - Initial version
- 2.0.1
    - Changes:
        - Changed the name of MMDVSQ_GetDivideRemainder and MMDVSQ_GetDivideQuotient functions
- 2.0.2
    - Bug fix:
        - Fixed the MMDVSQ_GetExecutionStatus function related to the incorrect execution status

MPU

The current MPU driver version is 2.1.0

- 2.0.0
    - Initial version
- 2.1.0
    - API updates:
        - Updated the mpu_region_num_t and mpu_master_t to uint32_t
        - Updated the mpu_low_masters_access_rights_t and mpu_high_masters_access_rights_t to mpu_rwxrights_master_access_control_t and mpu_rwrights_master_access_control_t
        - Updated the MPU_SetRegionLowMasterAccessRights() and MPU_SetRegionHighMasterAccessRights() to MPU_SetRegionRwxMasterAccessRights() and MPU_SetRegionRwMasterAccessRights()

PDB

The current PDB driver version is 2.0.1

- 2.0.0
    - Initial version
- 2.0.1
    - Changed the PDB register base array to a constant

PIT

The current PIT driver version is 2.0.0

- 2.0.0
    - Initial version

PMC

The current PMC driver version is 2.0.0

- 2.0.0
    - Initial version

PORT

The current PORT driver version is 2.0.2

- 2.0.1
    - Changes:
        - Added "const" in function parameters
        - Updated enumeration variable names
- 2.0.2
    - Changes:
        - Added feature guard macros in the driver

PWM

The current PWM driver version is 2.0.0

- 2.0.0
    - Initial version

RCM

The current RCM driver version is 2.0.1

- 2.0.0
    - Initial version
- 2.0.1
    - [KPSDK-10249] Fixed the kRCM_SourceSw bit shift issue.

SIM

The current SIM driver version is 2.0.0

- 2.0.0
    - Initial version

SMC

The current SMC driver version is 2.0.2

- 2.0.0
    - Initial version
- 2.0.1
    - Changes:
        - Updated for KL8x
- 2.0.2
    - Bug fix:
        - Added DSB before WFI and ISB after WFI
    - Changes:
        - Updated the SMC_SetPowerModeVlpw implementation

UART

The current UART driver version is 2.1.1

- 2.0.0
    - Initial version
- 2.1.0
    - Add transactional APIs
- 2.1.1
    - Removed needless check of event flags and assert in UART_RTOS_Receive
    - Wait always for RX event flag in UART_RTOS_Receive

WDOG

The current WDOG driver version is 2.0.0

- 2.0.0
  - Initial version

XBARA

The current XBARA driver version is 2.0.3

- 2.0.0
  - Initial version
- 2.0.1
  - Bug fix:
    - Fixed w1c bits for the XBARA_SetOutputSignalConfig function
- 2.0.2
  - Changes:
    - Changed the array clock name
- 2.0.3
  - Bug fix:
    - Corrected a configuration for the function XBAR_SetOutputSignalConfig

XBARB

The current XBARB driver version is 2.0.1

- 2.0.0
  - Initial version
- 2.0.1
  - Bug fix:
    - Corrected the XBARB_SetSignalsConnection function
  - Changes:
    - Changed the array clock name

CLOCK

The current CLOCK driver version is 2.2.0

- 2.0.0
  - Initial version
- 2.1.0
  - Changes:
    - Merged the fsl_mcg and fsl_osc into fsl_clock
- 2.2.0
  - New features:
    - [KPSDK-9157] Updated the CLOCK_SetFeiMode/CLOCK_SetFbiMode/CLOCK_BootToFeiMode() to support set MCG_C4[DMX32]=1 in FEI/FBI modes
  - Bug fix:
    - Updated the IP_CLOCKS array, removed unused gates, and added missing gates

10 Middleware Log

DMA Manager for KSDK

The current DMA Manager driver version is 2.0.0

- 2.0.0
  - Initial version

EMVL1 for KSDK

The current EMVL1 driver version is 2.0.1

- 2.0.0
  - Initial version
- 2.0.1
  - Bug fix:
    - Fixed low level driver protocol timer failures during EMVL1 pre-certification tests (KPSDK-9556)
    - Fixed incorrect T0 command response that causes long command responses (KPSDK-8707). Command case2, case3, and case4 are affected

The current FatFs driver version is R0.11a

- Added glue functions for low level drivers (SDHC, SDSPI, RAM, and MMC) and modified the diskio.c file
- Added RTOS wrappers to make FatFs thread-safe. Modified the syscall.c file
- Renamed ffconf.h file to ffconf_template.h file. Each application should contain its own ffconf.h file
- Included ffconf.h into diskio.c files to enable selection of a physical disk from the ffconf.h by macro definition
- Conditional compilation of physical disk interfaces in diskio.c

lwIP for KSDK

The current lwIP version is based on the lwIP 1.5.0 released on 2016-03-22

- 1.5.0
  - New features:
    - Ported the lwIP 1.5.0 (2016-03-22) to the KSDK 2.0.0
- 1.4.1_rev1
  - New features:
    - Ported the lwIP 1.4.1 to the KSDK 2.0.0
- 1.4.1_rev2
  - New features:
    - Enabled critical sections in lwIP
  - Bug fix
    - Fixed the default lwIP packet buffer size to accept a maximum size frame from the ENET driver
    - Fixed a possible drop of multiframe packets during transmission

mbedTLS for KSDK

The current mbedTLS driver version is based on the mbedTLS 2.2.1 released on 2016-01-05

- 2.2.1
  - New features:
    - Ported mbedTLS 2.2.1 to KSDK 2.0.0
    - Added support for the mmCAU cryptographic acceleration module. Accelerated MD5, SHA, AES, and DES
    - Added support for the 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 the MBEDTLS_GCM_CRYPT_ALT configuration parameter to enablereloading the mbedtls_gcm_crypt_and_tag() function
    - Added the MBEDTLS_ECP_MUL_COMB_ALT to enable an alternate implementation of the ecp_mul_comb() function
    - Added the MBEDTLS_ECP_ADD_ALT configuration parameter to enable reloading the ecp_add() function

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

mmCAU library for KSDK

The current mmCAU driver version is 2.0.0

- 2.0.0
    - New features:
        - Q4/2013 release of the CAU library
        - Added the fsl_mmcau.h/fsl_mmcau.c optional layer between the application and the legacy CAU library (cau_api.h). This API has no alignment requirements

SDMMC for KSDK

The current SDMMC driver version is 2.1.1

- 2.1.0
    - Bug fix:
        - Changed the callback mechanism when sending a command
        - Fixed the performance low issue when transferring data
    - Changes:
        - Changed the name of error codes returned by an internal function
        - Merged all host-related attributes into one structure
        - Optimized the function to set a maximum data bus width for the MMC card
- 2.1.1
    - Bug fix:

    - Fixed the block range boundary error when transferring data to the MMC card
    - Fixed the bit mask error in the SD card when switching to a high-speed function
  - Changes:
    - Added an error code to indicate that SDHC ADMA1 transfer type is not supported
    - Optimized the SD card initialization function

wolfSSL for KSDK

The current wolfSSL is 3.9.0 based on the release 3.9.0 of wolfSSL

- 3.8.0
  - New features:
    - Added support for the Kinetis LTC hardware acceleration module, which accelerates AES, 3DES, TFM module (modular integer arithmetic), and ECC wolfSSL modules
    - Added support for the Kinetis random number generator modules TRNG and RNGA.
  - Miscellaneous changes:
    - The Kinetis mmCAU acceleration now uses the "fsl_mmcau.h" file instead of the "cau_api.h" file
    - In DSA, wc_dsaSign() function is updated to repeat the wc_RNG_GenerateBlock() until k is less than q
    - wolfssl/wolfcrypt/settings.h file is changed to remove the unused macros and add support for the KSDK 2.0.0
    - In the wolfcrypt/src/asn.c file, the ksdk_time(time_t) is changed to external, to be defined by an application
- 3.9.0
  - New features:
    - Added more LTC public key acceleration (curve 25519 and RSA 4096)
    - Added the FREESCALE_LTC_TFM_RSA_4096_ENABLE macro to enable the RSA 4096 on the K8x/KL8x LTC
    - Increased the LTC_MAX_ECC_BITS to 384 to enable the ECC-384 curve acceleration on LTC
    - Added the FREESCALE_LTC_SHA for the KL8x SHA-1 and SHA-256 hardware acceleration
  - Changes:
    - Changed wolfssl/wolfcrypt/settings.h to remove unused macros and add support for KSDK 2.0.0
    - Implemented the LTC public key acceleration in a separate source file ksdk_port.h and ksdk_port.c

11 RTOS Log

µC/OS-II OS for KSDK

The current version is µC/OS-II OS V2.92.11

- 2.92.11
  - New features:
    - Added a template application for the Kinetis Expert (KEx) tool (template_application)
  - Changes:
    - Reduced the folder structure to keep only Kinetis-related information
    - Added wrappers to adaptat PendSV_Handler and SysTick_Handler. Related files are located in rtos\ucosii_<version>\uCOS-II\Ports\ARM-Cortex-Mx\Generic\<compiler>\fsl_isr_wrapper.S

µC/OS-III OS for KSDK

The current version is µC/OS-III OS V3.05.01

- V3.05.01
  - New features:
    - Added a template application for the Kinetis Expert (KEx) tool (template_application)
  - Bug fix:
    - [KPSDK-7247] Downgraded port files from V3.05.01 to V3.05.00 because of the context switch issue
  - Changes:

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

12 Revision History

This table summarizes revisions to this document.

Table 3. Revision history

Revision number	Date	Substantive changes
0	07/2016	Initial release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

NXP, the NXP logo, Freescale, the Freescale logo, Tower, and Kinetis are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, ARM powered logo, Keil, µVision, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. mbed is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

Document Number KSDK200KV58FRN
Revision 0, 07/2016