

# MCUXpresso SDK Release Notes

## Supporting MKV10Z64/128 and MKV11Z64/128

### Contents

## 1 Overview

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

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

|    |                                      |    |
|----|--------------------------------------|----|
| 1  | Overview.....                        | 1  |
| 2  | MCUXpresso SDK.....                  | 1  |
| 3  | Development Tools.....               | 2  |
| 4  | Supported Development Systems.....   | 2  |
| 5  | Release Contents.....                | 2  |
| 6  | MCUXpresso SDK Release Package.....  | 3  |
| 7  | MISRA Compliance.....                | 4  |
| 8  | Known Issues.....                    | 5  |
| 9  | Change Log - Peripheral drivers..... | 6  |
| 10 | Change Log - Middleware.....         | 13 |
| 11 | Change Log - RTOS.....               | 15 |

## 2 MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpressoSDK is the evolution of Kinetis SDK v2.3.0, includes support for both LPC and i.MX System-on-Chips (SoC). The same drivers, APIs, and middleware are still available with support for Kinetis, LPC, and i.MX silicon. The MCUXpresso SDK adds support for the MCUXpresso IDE, a new Eclipse-based toolchain that works with all MCUXpresso



## Development Tools

SDKs. Easily import your SDK into the new toolchain to have access to all of the available components, examples, and demos for your target silicon. In addition to the MCUXpresso IDE, support for the MCUXpresso Config Tools allows for easy cloning of existing SDK examples and demos, allowing users to easily leverage the existing software examples provided by the SDK for their own projects.

### NOTE

In order to maintain compatibility with legacy FSL code, the filenames and source code in MCUXpresso SDK containing the legacy Freescale prefix 'FSL' has been left as is. The 'FSL' prefix has been redefined as the NXP Foundation Software Library.

## 3 Development Tools

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

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

## 4 Supported Development Systems

This release supports boards and devices listed in this table. Boards and devices in boldface were tested in this release:

**Table 1. Supported MCU devices and development boards**

| Development boards              | MCU devices                                                                                                                                                                                                                                    |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TWR-KV11Z75M, FRDM-KV11Z</b> | <b>MKV11Z128VLH7</b> , MKV11Z64VLH7, MKV11Z128VLC7, MKV11Z64VLC7, MKV11Z128VFM7, MKV11Z64VFM7, MKV11Z128VLF7, MKV11Z64VLF7, MKV10Z128VLH7, MKV10Z64VLH7, MKV10Z128VLC7, MKV10Z64VLC7, MKV10Z128VFM7, MKV10Z64VFM7, MKV10Z128VLF7, MKV10Z64VLF7 |

## 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       |
| 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)**

|                                                                                                           |                                               |
|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| Documentation                                                                                             | <install_dir>/docs                            |
| Middleware                                                                                                | <install_dir>/middleware                      |
| DMA manager                                                                                               | <install_dir>/middleware/dma_manager          |
| FatFS stack                                                                                               | <install_dir>/middleware/fatfs                |
| Motor Control libraries                                                                                   | <install_dir>/middleware/motor_control        |
| RTCESL libraries                                                                                          | <install_dir>/middleware/rtcesl               |
| SDMMC card driver                                                                                         | <install_dir>/middleware/sdmmc                |
| 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 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 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.

### 6.2.2 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

### 6.2.3 CMSIS

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

## 7 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.                                                                                    |
|                 | 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                                                                                                                                                 |
| 10.1            | 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 MCUXpresso SDK, place it in a directory close to the root to prevent file paths from exceeding the maximum character length specified by the Windows operating system. The recommended location is the C:\nxp folder.

### 8.2 USBFS controller issue

Because of the USBFS controller design issues, the USB host suspend/resume demos (usb\_suspend\_resume\_host\_hid\_mouse) of the full speed controller do not support the low speed device directly.

### 8.3 USB PID issue

Because the PID of all USB device examples is updated, uninstall the device drivers and then reinstall when the device (with new PID) is plugged in the first time.

## 8.4 New project in MCUXpresso sometimes cannot be built

In KDS 3.2, a new project is created by selecting 'File' -> 'New' -> 'Kinetis SDK 2.x Project'. If you select 'All drivers' but do not select RTOS, the project will contain the `freertos/portable/heap_4.c` file, which cannot be built. As a workaround, remove the FreeRTOS folder with all content from your project, because these files are only needed for FreeRTOS.

## 9 Change Log - Peripheral drivers

### ADC16

The current ADC16 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

### CRC

The current CRC driver version is 2.0.1

- 2.0.0
  - Initial version
- 2.0.1
  - Bug fix:
    - DATA and DATALL macro definition moved from header file to source file

### 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 2.2.0

- 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
- 2.1.3
  - Bug Fix:
    - DSPI eDMA driver doesn't support the odd transfer data size and the bitsPerFrame greater than 8.
  - Optimization:
    - Added the `#ifndef/#endif` to allow users to change the default tx value at compile time.
- 2.1.4
  - Bug fix:
    - DSPI EDMA driver: The DSPI instance that has separated DMA request source can transfer up to 32767 byte data in one DSPI\_MasterTransferEDMA() transfer now
- 2.2.0
  - New features:
    - Add gasket feature for the SPI EDMA driver, which reduces one channel used in the EDMA master transfer. With this feature, only two channels are needed. For example, if the gasket feature is supported, one could use the DSPI\_MasterTransferCreateHandleEDMA function like below:  
`DSPI_MasterTransferCreateHandleEDMA(EXAMPLE_DSPI_MASTER_BASEADDR, &g_dspi_edma_m_handle, DSPI_MasterUserCallback, &userData, &dspiEdmaMasterRxRegToRxDataHandle, NULL, &dspiEdmaMasterIntermediaryToTxRegHandle);`
    - Add dummy data setup API to allow users to configure the dummy data to be transferred.
    - Add new APIs for half-duplex transfer function. Users can send and receive data by one API in polling/interrupt/EDMA way, and users can choose either transmit first or receive first. The PCS pin can be configured as assert status in transmission (between transmit and receive) by setting the `isPcsAssertInTransfer` to true.

## eDMA

The current eDMA driver version is 2.1.2

- 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:

## Change Log - Peripheral drivers

- 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
- 2.0.4
  - Improvement:
    - Added support for SoCs with multiple eDMA instances.
    - Added the pubweak DriverIRQHandler for the KL28T DMA1 and MCIMX7U5\_M4.
- 2.0.5
  - Improvement:
    - Added the pubweak DriverIRQHandler for the K32H844P (16 channels shared).
- 2.1.0
  - Improvement:
    - Changed the EDMA\_GetRemainingBytes API to EDMA\_GetRemainingMajorLoopCount because of the eDMA IP limitation (see API comments/note for details).
- 2.1.1
  - Improvement:
    - Added documentation of the eDMA data flow when scatter/gather is implemented for the EDMA\_HandleIRQ API.
    - Updated and corrected 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.

## EWM

The current EWM driver version is 2.0.1

- 2.0.0
  - Initial version
- 2.0.1
  - Fix EWM\_Deinit hardfault issue

## Flash

The current Flash driver version is 2.3.1

- 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
- 2.2.0
  - New features:
    - Support FTFL device in FLASH\_Swap API
    - Support various pflash start addresses
    - Add support for KV58 in cache clear function
    - Add support for device with secondary flash (KW40)
  - Bug fix
    - Compiled execute-in-ram functions as PIC binary code for driver use
    - Added missed flexram properties
    - Fixed unaligned variable issue for execute-in-ram function code array



- 2.3.0
  - New features:
    - Add support for device with LP flash (K3S/G)
    - Add flash prefetch speculation APIs
  - Improvement
    - Refine flash\_cache\_clear function
    - Reorganize the member of flash\_config\_t struct
- 2.3.1
  - Bug fix
    - Unified Flash IFR design from K3
    - New encoding rule for K3 flash size
- 2.0.2
  - Improvement:
    - Split the FlexIO component which combines all FlexIO/FlexIO UART/FlexIO I2C/FlexIO I2S drivers into several components: the flexio component, flexio\_uart component, flexio\_i2c\_master component and flexio\_i2s component.

## FlexCAN

The 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

## FTM

The current FTM driver version is 2.0.2

- 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
- 2.0.2
  - Feature:
    - Add support to Quad Decoder feature with new APIs:
      - FTM\_GetQuadDecoderFlags()
      - FTM\_SetQuadDecoderModuloValue()
      - FTM\_GetQuadDecoderCounterValue()
      - FTM\_ClearQuadDecoderCounterValue()

### GPIO

The current GPIO driver version is 2.2.1

- 2.1.0
  - API Interface Change
    - Added "pins" or "pin" to some API names
    - Renamed the "GPIO\_PinConfigure" to "GPIO\_PinInit"
- 2.1.1
  - API Interface Change
    - Added API to check attribute bytes
- 2.2.1
  - API Interface Change
    - Refined naming of API while keeping all original APIs, marking them as deprecated. The original API will be removed in the next release. The main change is the updated API with prefix of \_PinXXX() and \_PorortXXX.

### I2C

The current I2C driver version is 2.0.5

- 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
- 2.0.2
  - Bug Fix:
    - Fixed the issue that occurs in master receive and slave transmit mode with no stop flag and master can't start a next transfer because it can't send out restart signal
    - Fixed a data transfer out of order issue which occurs because of a memory barrier
  - New Features:
    - Added an address nak event for the master.
    - Added a general call event for the slave.
- 2.0.3
  - Bug fix
    - Removed enableHighDrive member in the master/slave configuration structure because the user needs to use the DSE bit in the port register to configure the high drive strength capability.
    - Added reset registers operation in I2C\_MasterInit and I2C\_SlaveInit APIs, and fix the issue that I2C could not switch between master and slave mode.
    - Improved the slave IRQ handler to handle the corner case that stop flag and address match flag come synchronously.
- 2.0.4
  - Bug fix
    - Added proper handle for transfer config flag kI2C\_TransferNoStartFlag to support transmit with kI2C\_TransferNoStartFlag flag. Only supports write only or write+read with no start flag. It does not support read only with no start flag.
- 2.0.5
  - Improvement
    - Added the I2C\_WATL\_TIMEOUT macro to allow users to specify the timeout times for waiting flags in functional API and blocking transfer API.

### 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 2.0.1

- 2.0.0
  - Initial version
- 2.0.1
  - Driver update
    - Update the LPTMR driver due to the register LPTMRx\_CMR/CNR in some devices that have become 32-bit, so that the updated LPTMR driver supports the 32 bit CNR and CMR register.

## **MMDVSQ**

The current MMDVSQ driver version is 2.0.2

- 2.0.0
  - Initial version
- 2.0.1
  - Other changes:
    - Change name of MMDVSQ\_GetDivideRemainder and MMDVSQ\_GetDivideQuotient functions
- 2.0.2
  - Bug fix:
    - Fix MMDVSQ\_GetExecutionStatus function get execution status wrong

## **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

## **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

### 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 2.1.0

- 2.0.0
  - Initial version
- 2.1.0
  - Add new APIs of SIM\_GetRfAddr() and SIM\_EnableSystickClock()

### SMC

The current SMC driver version is 2.0.3

- 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\_SetPowerModeVlppw implementation
- 2.0.3
  - Add APIs SMC\_PreEnterStopModes, SMC\_PreEnterWaitModes, SMC\_PostExitWaitModes, and SMC\_PostExitStopModes

### UART

The current UART driver version is 2.1.5

- 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
- 2.1.2
  - Fix baud rate fine adjust bug to make the computed baud rate more accurate
- 2.1.3
  - Add rx framing error and parity error status check when use interrupt transfer
- 2.1.4
  - Change parameter type in UART\_RTOS\_Init() struct rtos\_uart\_config -> uart\_rtos\_config\_t
  - Bug fixed:

- Disable UART receives interrupts instead of disabling all NVIC when reading data from a ring buffer. When a ring buffer is used, receive non-blocking disables all NVIC interrupts to protect the ring buffer, which will have a negative effect on other IPS using the interrupts.
- 2.1.5
  - Add hardware flow control function support.
  - Add idle line detected feature in UART\_TransferNonBlocking function. If an idle line was detected, a callback is triggered with status kStatus\_UART\_IdleLineDetected returned. This feature may be useful when the received bytes are less than the expected receive data size. Before triggering the callback, data in the FIFO will be read out (if has FIFO), and all interrupts will not be disabled except when the receive data size reaches 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 not be bigger than the RX FIFO size). The data is received and a callback is triggered when data receive ends.

## WDOG

The current WDOG driver version is 2.0.0

- 2.0.0
  - Initial version

## CLOCK

The current CLOCK driver version is 2.2.1

- 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
- 2.2.1
  - Bug fix:
    - Fix the issue that MCG could not switch to FEE/FBE/PBE modes when the OSCERCLK clock is not enabled

# 10 Change Log - Middleware

## DMA Manager

The current DMA Manager driver version is 2.1.0

- 2.0.0
  - Initial version
- 2.1.0

## Change Log - Middleware

- Update DMA manager interface to support dynamic configuration of the managed area. This is used for platforms with multiple cores

### FatFs

The current FatFs driver version is R0.12c

- R0.11a
  - Added glue functions for low level drivers (SDHC, SDSPI, RAM, and MMC) and modified the diskio.c file
  - Added RTOS wrappers to make FatFs thread-safe. Modified the syscall.c file
  - Renamed ffconf.h file to ffconf\_template.h file. Each application should contain its own ffconf.h file
  - Include ffconf.h into diskio.c to enable selection of physical disk from ffconf.h by macro definition
  - Conditional compilation of physical disk interfaces in diskio.c
- R0.12b\_rev0
  - Upgrade to version 0.12b
- R0.12c\_rev0
  - Upgrade to version 0.12c and apply patches ff\_12c\_p1.diff and ff\_12c\_p2.diff

### SDMMC

The current SDMMC driver version is 2.2.1

- 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
- 2.1.2
  - New feature
    - Add fsl\_host.h to provide prototype to adapt different controller IPs(SDHC/SDIF)
    - Add adaptor code in sdmmc/port folder to adapt different host controller IPs with different transfer modes(int/polling/freertos). Application include different adaptor code to make application simpler
    - Adaptor code provides HOST\_Init/HOST\_Deinit/CardInsertDetect APIs to do host controller initialize and transfer function configuration. SDMMC card stack uses adaptor code inside stack to wait card insert and configure host when calling card init APIs (SD\_Init/MMC\_Init/SDIO\_Init)
    - So this change requires user to include host adaptor code into application. If not, link errors for cannot find the definition of HOST\_Init/HOST\_Deinit/CardInsertDetect will appear
  - New feature
    - Improve SDMMC to support SD v3.0 and EMMC v5.0
  - Bug fix:
    - Fix wrong comparison between count and length in MMC\_ReadBlocks/MMC\_WriteBlocks
- 2.1.3
  - Bug fix:
    - Non high-speed SD Card init fail at switch to high speed
  - Miscellaneous changes:

- Optimized tuning/mmc switch voltage/mmc select power class/mmc select timing function
  - Added strobe dll for mmc HS400 mode
  - Added Delay for SD Card power up
- 2.1.4
  - Miscellaneous changes:
    - Added Host reset function for card re-initialization
    - Added Go\_Idle function for SDIO card
    - Added Host\_ErrorRecovery function for host error recovery procedure
    - Added cache maintain operation
    - Added HOST\_CARD\_INSERT\_CD\_LEVEL to improve compatibility
  - Bug fix:
    - Fixed issue where card cannot detect dynamically
- 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 features
    - 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.

## 11 Change Log - RTOS

### FreeRTOS OS

The current version is FreeRTOS OS 9.0.0. The original package is available at [freertos.org](http://freertos.org).

- 8.2.3
  - New features:
    - Added tickless idle mode support
    - Added a template application for Kinetis Expert (KEx) tool (template\_application)
  - Changes:
    - Reduced the folder structure to keep only Kinetis-related information
- 9.0.0\_rev0
  - New features:
    - Example freertos\_sem\_static
    - Static allocation support RTOS driver wrappers
  - Other changes:

## Change Log - RTOS

- Tickless idle rework. Support for different timers is in separated files (fsl\_tickless\_systick.c, fsl\_tickless\_lptmr.c)
- Remove configuration option configSYSTICK\_USE\_LOW\_POWER\_TIMER. Low power timer is now selected by linking of appropriate file fsl\_tickless\_lptmr.c
- Remove configOVERRIDE\_DEFAULT\_TICK\_CONFIGURATION in RVDS port. Use of **attribute((weak))** is preferred solution. Not same as **\_weak**
- 9.0.0\_rev1
  - New features:
    - Enable -flto optimization in GCC by adding **attribute((used))** for vTaskSwitchContext
    - Enable KDS Task Aware Debugger. Apply FreeRTOS patch to enable configRECORD\_STACK\_HIGH\_ADDRESS macro. Modified files are task.c and FreeRTOS.h
- 9.0.0\_rev2
  - New features:
    - Enabled MCUXpresso thread aware debugging. Added freertos\_tasks\_c\_additions.h and configINCLUDE\_FREERTOS\_TASK\_C\_ADDITIONS\_H and configFRTOS\_MEMORY\_SCHEME macros
- 9.0.0\_rev3
  - New features:
    - Tickless idle mode support for Cortex-A7. Add fsl\_tickless\_epit.c and fsl\_tickless\_generic.h in portable/IAR/ARM\_CA9 folder.
    - Enable float context saving in IAR for Cortex-A7. Add configUSE\_TASK\_FPU\_SUPPORT macros. Modify port.c and portmacro.h in portable/IAR/ARM\_CA9 folder.
  - Other changes:
    - Transform ARM\_CM core specific tickless low power support into generic form under FreeRTOS.



**How to Reach Us:****Home Page:**[nxp.com](http://nxp.com)**Web Support:**[nxp.com/support](http://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](http://nxp.com/SalesTermsandConditions).

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

© 2017 NXP B.V.

Document Number MCUXSDKKV11RN  
Revision 0, 11/2017

