NXP Semiconductors
Release Notes

Document Number: MCUXSDKKV31RN
Rev. 0, 03/2017

MCUXpresso SDK Release Notes Supporting TWR-KV31F120M and FRDM-KV31F

Contents

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 6
8 Known Issues 6
9 Change Log - Peripheral drivers 7
10 Change Log - Middleware 14
11 Change Log - RTOS 15
12 Revision History 16

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.

2 MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpressoSDK is the evolution of Kinetis SDK v2.0.0 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 v10.0.0, a new Eclipse-based toolchain that works with all

MCUXpresso SDKs. Easily import your SDK into the new toolchain to have access to all of the available components, examples, and demos for your target silicon. In addition to the MCUXpresso IDE v10.0.0, support for the MCUXpresso Project Generator 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 make the life of our customers a bit easier, 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 7.80.4
- MDK-ARM Microcontroller Development Kit (Keil)® 5.21a
- Makefiles support with GCC revision v5-2016-q3 from ARM Embedded
- MCUXpresso IDE v10.0.0

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
TWR-KV31F120M and FRDM-KV31F	**MKV31F512VLL12**, MKV31F512VLH12, MKV30F128VFM10, MKV30F128VLF10, MKV30F128VLH10, MKV30F64VFM10, MKV30F64VLF10, MKV30F64VLH10, MKV31F128VLH10, MKV31F128VLL10, MKV31F256VLH12, MKV31F256VLL12

5 Release Contents

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

Table 2. Release contents

Deliverable	Location
Boards	<install_dir>/boards
Demo applications	<install_dir>/boards/<board_name>/demo_apps
USB demo applications	<install_dir>/boards/<board_name>/usb_examples
Driver examples	<install_dir>/boards/<board_name>/driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples

Table continues on the next page...

Table 2. Release contents (continued)

Multicore examples	<install_dir>/boards/<board_name>/multicore_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
lwIP Documentation	<install_dir>/docs/lwip
Middleware	<install_dir>/middleware
lwIP stack	<install_dir>/middleware/lwip_<version>
DMA manager	<install_dir>/middleware/dma_manager_<version>
EMV stack	<install_dir>/middleware/emv_<version>
FatFS stack	<install_dir>/middleware/fatfs_<version>
mmCAU	<install_dir>/middleware/mmcau_<version>
Motor Control libraries	<install_dir>/middleware/motor_control_<version>
Multicore stack	<install_dir>/middleware/multicore_<version>
RTCESL libraries	<install_dir>/middleware/rtcesl_<version>
SDMMC card driver	<install_dir>/middleware/sdmmc_<version>
USB stack	<install_dir>/middleware/usb_<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 MCUXpresso SDK Release Package

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

6.1 Device support

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

The device-specific header files provide a direct access to the MCU peripheral registers. The device header file provides an overall SoC memory mapped register definition. In addition to the overall device memory mapped header file, the MCUXpresso SDK also includes the feature header file for each peripheral instantiated on the SoC.

The toolchain folder contains the startup code and linker files for each supported toolchain. The startup code is a CMSIS-compliant startup that efficiently transfers the code execution to the main() function.

6.1.1 Board support

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

6.1.2 Demo applications and other examples

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

The driver examples demonstrate the capabilities of the peripheral drivers. Each example implements a common use case to help demonstrate the driver functionality.

The RTOS and middleware folders each contain examples demonstrating the use of the included source.

6.2 Middleware

6.2.1 USB stack

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

6.2.1.1 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

Table continues on the next page...

Table 3. Peripheral devices (continued)

	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

6.2.2 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).

6.2.3 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.

For details, see the FatFs documentation installed at <install_dir>/middleware/fatfs_<version>/doc.

6.2.4 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.5 CMSIS

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

7 MISRA Compliance

All MCUXpresso SDK drivers and USB stack 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 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 Program flash issue

The last 8 KB flash segment is restricted to execute only by default, and users cannot break the restriction. This means that there is 512 KB flash in KV31, but only the first 504 KB flash that is accessible as normal flash for users.

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 2.1.4

- 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

eDMA

The current eDMA driver version is 2.1.1

- 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
- 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.

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

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 Wuad Decoder feature with new APIs:
      - FTM_GetQuadDecoderFlags()
      - FTM_SetQuadDecoderModuloValue()
      - FTM_GetQuadDecoderCounterValue()
      - FTM_ClearQuadDecoderCounterValue()

GPIO

The current GPIO driver version is 2.1.1

- 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 for the check attribute bytes

I2C

The current I2C driver version is 2.0.3

- 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
        - Remove enableHighDrive member in the master/slave configuration structure because the operation to HDRS bit is useless. The user needs to use DSE bit in port register to configure the high drive capability.
        - Add reset registers operation in I2C_MasterInit and I2C_SlaveInit APIs, and fix the issue that I2C could not switch between master and slave mode.
        - Improve slave IRQ handler to handle the corner case that stop flag and address match flag come synchronously.

LLWU

The current LLWU driver version is 2.0.1

- 2.0.0
    - Initial version
- 2.0.1
    - Changes:
        - Updated for KL8x

- 2.1.1
    - Bug fix:
        - Disable the auto stop feature in the EDMA driver. Previously the autostop feature is enabled at transfer when transfer with stop flag. If the previous transfer is without stop flag, then when starting a new transfer with stop flag, because the auto stop feature is enabled, the stop flag sends before starting the new transfer and the start flag can not successfully sent, so the transfer cannot start.
        - Change default slave configuration with address stall false

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.

LPUART

The current LPUART driver version is 2.2.3

- 2.1.0
    - Updated transactional APIs
- 2.1.1
    - Removed the needless check of event flags and assert in LPUART_RTOS_Receive
    - Wait always for RX event flag in LPUART_RTOS_Receive
- 2.2.0
    - Added seven data bits and MSB support
- 2.2.1

        - Added a separate RX and TX IRQ number support
- 2.2.2
    - Added software reset feature support.
    - Added software reset API to LPUART_Init().
- 2.2.3
    - Changed the parameter type in the LPUART_RTOS_Init() struct rtos_lpuart_config --> lpuart_rtos_config_t.

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

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 2.0.3

- 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_SetPowerModeVlpw implementation
- 2.0.3
  - Add APIs SMC_PreEnterStopModes, SMC_PreEnterWaitModes, SMC_PostExitWaitModes, and SMC_PostExitStopModes

UART

The current UART driver version is 2.1.4

- 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.

VREF

The current VREF driver version is 2.1.0

- 2.1.0
  - Added new functions:
    - Added VREF_SetTrim2V1Val() and VREF_GetTrim2V1Val() functions to supply 2V1 output mode.

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
    - Update DMA manager interface to support dynamic configure the manage area. This is used for platform with multiple cores

EMVL1

The current EMVL1 driver version is 2.1.0

- 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
- 2.0.2
    - Re-implemented a function for sending commands in T=0
    - Bug Fix:
        - Fixed the incorrect size of response in T=0 (KPSDK-11248)
        - Fixed a problem with command cases 3 in T=1; expected incorrect length of response (KPSDK-11335)
        - Fixed an incorrect length of response in T=1 (KPSDK-11868)
        - Fixed the usage application buffer for the data payload and overhead associated with T=1 protocol (KPSDK-11336)
- 2.1.0
    - Added abort transfer functionality

FatFs

The current FatFs driver version is R0.12b

- R0.12b_rev0
- 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
  - Conditional compilation of physical disk interfaces in diskio.c

SDMMC

The current SDMMC driver version is 2.1.2

- 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 inlude different adpator code to make application simplier
    - Adaptor code provides HOST_Init/HOST_Deinit/CardInsertDetect APIs to do host controller initialize and transfer function configuration. SDMMC card stack uses adaptor code inside stack to wait card insert and configure host when calling card init APIs (SD_Init/MMC_Init/SDIO_Init)
    - So this change requires user to include host adaptor code into application. If not, link errors for cannot find the definition of HOST_Init/HOST_Deinit/CardInsertDetect will appear
  - New feature
    - Improve SDMMC to support SD v3.0 and EMMC v5.0
  - Bug fix:
    - Fix wrong comparison between count and length in MMC_ReadBlocks/MMC_WriteBlocks

11 Change Log - RTOS

FreeRTOS OS

The current version is FreeRTOS OS 9.0.0. The original package is available at freertos.org.

- 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_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 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
- 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

12 Revision History

This table summarizes revisions to this document.

Table 4. Revision history

Revision number	Date	Substantive changes
0	03/2017	Initial release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number MCUXSDKKV31RN
Revision 0, 03/2017