

Kinetis SDK v.2.1.0 Release Notes Supporting FRDM-KL43Z

Contents

1 Overview

The Kinetis Software Development Kit (KSDK) 2.1.0 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 μ C/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](#).

2 KSDK 2.1.0

KSDK 2.1.0 is the evolution of KSDK 1.x into a more optimized software solution. KSDK 2.1.0 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

1	Overview.....	1
2	KSDK 2.1.0.....	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	Change Log - Peripheral drivers.....	6
10	Change Log - Middleware.....	12
11	Change Log - RTOS.....	13
12	Revision History.....	13



Development Tools

2.1.0 also eliminate external software dependencies. The Operating System Abstraction, Power Manager, and Clock Manager are no longer required by the KSDK 2.1.0 drivers.

The existing MQXTM RTOS support has been deprecated to focus on support of FreeRTOS OS, μ C/OS-II, and μ C/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 2.1.0 was compiled and tested with these development tools:

- Kinetis Design Studio IDE v3.2
- IAR Embedded Workbench for ARM version 7.70
- MDK-ARM Microcontroller Development Kit (Keil)[®] 5.20
- Makefiles support with GCC revision v4.9-2015-q3 from ARM Embedded
- Atollic[®] TrueSTUDIO[®] 5.5.2

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
FRDM-KL43Z	MKL27Z128VMP4, MKL27Z128VFT4, MKL27Z256VMP4, MKL27Z256VFT4, MKL27Z256VFM4, MKL27Z256VLH4, MKL13Z32VMP4, MKL13Z32VLK4, MKL13Z32VLH4, MKL13Z32VFT4, MKL13Z32VFM4, MKL13Z64VLH4, MKL13Z64VFT4, MKL13Z64VMP4, MKL13Z64VFM4, MKL13Z64VLK4, MKL33Z128VLH4, MKL33Z128VMP4, MKL33Z256VLH4, MKL33Z256VMP4, MKL33Z32VLK4, MKL33Z32VFT4, MKL33Z32VMP4, MKL33Z32VLH4, MKL33Z64VLH4, MKL33Z64VLK4, MKL33Z64VMP4, MKL33Z64VFT4, MKL43Z256VLH4, MKL43Z128VMP4, MKL43Z256VMP4.

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

Table continues on the next page...

Table 2. Release contents (continued)

Driver examples	<install_dir>/boards/<board_name>/driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Multicore examples	<install_dir>/boards/<board_name>/multicore_examples
Documentation	<install_dir>/docs
Middleware	<install_dir>/middleware
DMA manager	<install_dir>/middleware/dma_manager_<version>
FatFS stack	<install_dir>/middleware/fatfs_<version>
Multicore stack	<install_dir>/middleware/multicore_<version>
SDMMC card driver	<install_dir>/middleware/sdmmc_<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 2.1.0 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 File System

The FatFs file system is integrated with Kinetis 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.

6.2.2 RTOS

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

6.2.3 CMSIS

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

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.
	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 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 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 FreeRTOS issue

When generating a new FreeRTOS project with New Project Wizard tool, the user should deassert the macro option "configUSE_PORT_OPTIMISED_TASK_SELECTION" in the "FreeRTOSConfig.h" file while it is not used in the template project for some Cortex-M0+ devices, or it might not pass the compiling.

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

COP

The current COP 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

DMA

The current DMA driver version is 2.0.1

- 2.0.0
 - Initial version
- 2.0.1
 - Bug fix:
 - Fixed the DMA driver build fail by adding parenthesis. The issue occurs because of the MISRA C 2004 rule 12.5

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

Flash

The current Flash driver version is 2.2.0

- 2.0.0
 - Initial version
- 2.1.0
 - New features:
 - Support for the FTFLL 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:
 - Added support for devices with the secondary flash (KW40)
 - Added support for devices with the LP flash (K3S/G)
 - Added flash pre-fetch speculation APIs

FlexIO UART

The current FlexIO UART driver version is 2.1.4

- 2.1.0
 - New features:
 - Added a transfer prefix to the transactional APIs
 - Added the txSize/rxSize parameters to the handle structure to record the transfer size
 - Bug fix
 - Added an error handle to handle the data count zero or data buffer NULL situation
- 2.1.1
 - Bug fix
 - Renamed FLEXIO_UART_StopRingBuffer to FLEXIO_UART_TransferStopRingBuffer to align with the definition in the C file
- 2.1.2
 - Fixed the baud rate bug to ensure that the computed baud rate is accurate.
- 2.1.3
 - Added the Rx framing error and parity error status check when using an interrupt transfer.
- 2.1.4
 - Changed the parameter type in UART_RTOS_Init() struct rtos_uart_config -> uart_rtos_config_t.

FlexIO I2C

The current FlexIO I2C driver version is 2.1.1

- 2.1.0
 - New features:
 - Added a transfer prefix to the transactional APIs
 - Added the txSize/rxSize parameters to the handle structure to record the transfer size
 - Bug fix

Change Log - Peripheral drivers

- Added an error handle to handle the data count zero or data buffer NULL situation
- 2.1.1
 - Bug fix
 - Implemented the FLEXIO_I2C_MasterTransferBlocking API which is defined in the header file but has no implementation in the C file

FlexIO SPI

The current FlexIO SPI driver version is 2.1.0

- 2.1.0
 - New features:
 - Added a transfer prefix to the transactional APIs
 - Added the txSize/rxSize parameters to the handle structure to record the transfer size
 - Bug fix
 - Fixed the error register address return for 16-bit data write in the FLEXIO_SPI_GetTxDataRegisterAddress

FlexIO I2S

The current FlexIO I2S driver version is 2.1.1

- 2.1.0
 - New features:
 - Added a transfer prefix to the transactional APIs
 - Added the txSize/rxSize parameters to the handle structure to record the transfer size
- 2.1.1
 - Bug fix
 - Fixed the FlexIO I2S RX data read error and eDMA address error
 - Fixed the FlexIO I2S slave timer comparison setting error

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 2.0.2

- 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.

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

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.

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

Change Log - Peripheral drivers

- 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.

RTC

The current RTC driver version is 2.0.0

- 2.0.0
 - Initial version

SAI

The current SAI driver version is 2.1.1

- 2.0.0
 - Initial version
- 2.1.0
 - API name change:
 - SAI_GetSendRemainingBytes -> SAI_GetSentCount
 - SAI_GetReceiveRemainingBytes -> SAI_GetReceivedCount
 - Added "Transfer" prefix to all transactional API names
 - All transactional APIs use a base and a handle as input parameters
 - Unify the parameter names.
 - Bug fix:
 - Fixed the W1C bug while reading TCSR/RCSR registers.
 - Fixed the MOE enable flow issue by moving the MOE enable after MICS settings in SAI_TxInit/SAI_RxInit.
- 2.1.1
 - Optimization:
 - Reduced code size while not using transactional APIs.

SIM

The current SIM driver version is 2.0.0

- 2.0.0
 - Initial version

SLCD

The current SLCD driver version is 2.0.1

- 2.0.0
 - Initial version
- 2.0.1
 - Bug fix:
 - Changed the blink mode start setting flow
 - Miscellaneous changes:
 - Added static to SLCD global variables

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

SPI

The current SPI driver version is 2.0.1

- 2.0.0
 - Initial version
- 2.0.1
 - Bug fix
 - Fixed the SPI_Enable function parameter error
 - Set the s_dummy variable as a static variable in fsl_spi_dma.c
 - Optimization
 - Optimizes the code size when not using the transactional API
 - Improved performance in a polling method

TPM

The current TPM driver version is 2.0.2

- 2.0.0
 - Initial version
- 2.0.1
 - Bug fix:
 - Fixed the TPM_UpdateChnEdgeLevelSelect ACK wait issue.
 - Fixed the TPM_SetupDualEdgeCapture failure to set the FILTER register.
 - Fixed the TPM_UpdateChnEdgeLevelSelect ACK wait issue.
- 2.0.2
 - Bug fix:
 - Fixed issues in TPM_SetupPwm/TPM_UpdateChnEdgeLevelSelect /TPM_SetupInputCapture/TPM_SetupOutputCompare/TPM_SetupDualEdgeCapture functions wait acknowledgement when the channel is disabled.

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

VREF

The current VREF driver version is 2.1.0

- 2.1.0
 - Added new functions:

Change Log - Middleware

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

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 Change Log - Middleware

DMA Manager for KSDK

The current DMA Manager driver version is 2.0.0

- 2.0.0
 - Initial version

FatFs for KSDK

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

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

11 Change Log - RTOS

FreeRTOS OS for KSDK

The current version is FreeRTOS OS 8.2.3. The original package is available at 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

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	09/2016	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, and Kinetis, are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, ARM Powered, Cortex, Keil, and μ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2016 NXP B.V.

Document Number KSDK210KL43ZRN
Revision 0, 09/2016

