Freescale Semiconductor
Release Notes

Document Number: KSDK200RN
Rev. 0, 01/2016

Kinetis SDK v.2.0.0 Release Notes

Contents

1 Overview

The Kinetis Software Development Kit (KSDK) 2.0.0 is a collection of software enablement for Kinetis Microcontrollers that includes peripheral drivers, high level stacks including 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 and µC/OS. In addition to the base enablement, the KSDK is augmented with demo applications and driver example projects, and API documentation to help the customers quickly leverage the support of the Kinetis SDK.

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

2 What is New

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

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

At the middleware level, RTCS and MFS have been removed, and the USB stack has been replaced with a BSD licensed solution. KSDK 2.0.0 has also aligned with ARM® architecture through the integration of mbed TLS with our accelerated cryptography drivers. This integration ensures the highest level of performance from our on-chip security peripherals.

The existing MQX$^{\text{TM}}$ RTOS support has been deprecated to focus on support of FreeRTOS OS and μC/OS-II and μC/OS-III.

3 Development Tools

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

- Kinetis Design Studio IDE v3.0
- IAR Embedded Workbench for ARM version 7.50.0
- MDK-ARM Microcontroller Development Kit (Keil)® 5.17
- Makefiles support with GCC revision 4.9-2015-q3-update from ARM Embedded
- Atollic® TrueSTUDIO® 5.4.0

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	Kinetis MCU devices
MAPS-KS22 (MKS22FN256VLL12)	**MKS22FN256VLL12**, MKS22FN256VLH12, MKS22FN128VLL12, MKS22FN128VLH12
TWR-K80F150M (MK80FN256VDC15)	**MK80FN256VDC15**, MK80FN256VLL15, MK80FN256VLQ15, MK80FN256CAx15
FRDM-K82F (MK82FN256VLL15)	MK82FN256VDC15, **MK82FN256VLL15**, MK82FN256VLQ15, MK82FN256CAx15
FRDM-K64F (MK64FN1M0VLL12)	MK24FN1M0VDC12, MK24FN1M0VLL12, MK24FN1M0VLQ12, MK63FN1M0VLQ12, MK63FN1M0VMD12, MK64FN1M0VDC12, **MK64FN1M0VLL12**, MK64FN1M0VLQ12, MK64FN1M0VMD12, MK64FX512VDC12, MK64FX512VLL12, MK64FX512VLQ12, MK64FX512VMD12
TWR-K21F120M (MK21FN1M0AVMC12)	MK21FN1M0VLQ12, **MK21FN1M0VMC12**, MK21FN1M0VMD12, MK21FX512VLQ12, MK21FX512VMC12, MK21FX512VMD12, MK21FN1M0AVLQ12, MK21FN1M0AVMC12, MK21FN1M0AVMD12, MK21FX512AVLQ12, MK21FX512AVMC12, MK21FX512AVMD12, MK22FN1M0VLH12, MK22FN1M0VLK12, MK22FN1M0VLL12, MK22FN1M0VLQ12, MK22FN1M0VMC12, MK22FN1M0VMD12, MK22FX512VLH12, MK22FX512VLK12, MK22FX512VLL12, MK22FX512VLQ12, MK22FX512VMC12, MK22FX512VMD12, MK22FN1M0AVLH12,

Table continues on the next page...

Table 1. Supported MCU devices and development boards (continued)

	MK22FN1M0AVLK12, MK22FN1M0AVLL12, MK22FN1M0AVLQ12, MK22FN1M0AVMC12, MK22FN1M0AVMD12, MK22FX512AVLH12, MK22FX512AVLK12, MK22FX512AVLL12, MK22FX512AVLQ12, MK22FX512AVMC12, MK22FX512AVMD12
TWR-K22F120M (MK22FN512VDC12)	MK02FN128VFM10, MK02FN128VLF10, MK02FN128VLH10, MK02FN64VLH10, MK02FN64VLF10, MK02FN64VFM10, MK22FN128VDC10, MK22FN128VLL10, MK22FN128VLH10, MK22FN128VMP10, MK22FN128CAH12, MK22FN256CAH12, MK22FN256VDC12, MK22FN256VLH12, MK22FN256VLL12, MK22FN256VMP12, **MK22FN512VDC12**, MK22FN512VLL12, MK22FN512VLH12, MK22FN512CAP12, MK22FN512VMP12
FRDM-KL43Z (MKL43Z256VLH4)	MKL13Z32VMP4, MKL13Z32VLK4, MKL13Z32VLH4, MKL13Z32VFT4, MKL13Z32VFM4, MKL13Z64VLH4, MKL13Z64VFT4, MKL13Z64VMP4, MKL13Z64VFM4, MKL13Z64VLK4, MKL17Z128VFT4, MKL17Z128VLH4, MKL17Z128VFM4, MKL17Z128VMP4, MKL17Z256VFM4, MKL17Z256VMP4, MKL17Z256VFT4, MKL17Z256VLH4, MKL27Z128VLH4, MKL27Z128VFM4, MKL27Z128VMP4, MKL27Z128VFT4, MKL27Z256VMP4, MKL27Z256VFT4, MKL27Z256VFM4, MKL27Z256VLH4, MKL33Z128VLH4, MKL33Z128VMP4, MKL33Z256VLH4, MKL33Z256VMP4, MKL33Z32VLK4, MKL33Z32VFT4, MKL33Z32VMP4, MKL33Z32VLH4, MKL33Z64VLH4, MKL33Z64VLK4, MKL33Z64VMP4, MKL33Z64VFT4, MKL43Z128VLH4, MKL43Z128VMP4, MKL43Z256VMP4, **MKL43Z256VLH4**
TWR-K65F180M (MK65FN2M0VMI18)	MK26FN2M0VLQ18, MK26FN2M0VMD18, MK26FN2M0CAC18, MK26FN2M0VMI18, MK65FN2M0VMI18, MK65FN2M0CAC18, MK65FX1M0CAC18, **MK65FX1M0VMI18**
FRDM-K22F (MK22FN512VLH12)	MK02FN128VFM10, MK02FN128VLF10, MK02FN128VLH10, MK02FN64VLH10, MK02FN64VLF10, MK02FN64VFM10, MK22FN128VDC10, MK22FN128VLL10, MK22FN128VLH10, MK22FN128VMP10, MK22FN128CAH12, MK22FN256CAH12, MK22FN256VDC12, MK22FN256VLH12, MK22FN256VLL12, MK22FN256VMP12, MK22FN512VDC12, MK22FN512VLL12, **MK22FN512VLH12**, MK22FN512CAP12, MK22FN512VMP12
TWR-KL43Z48M (MKL43Z256VLH4)	MKL17Z128VFT4, MKL17Z128VLH4, MKL17Z128VFM4, MKL17Z128VMP4, MKL17Z256VFM4, MKL17Z256VMP4, MKL17Z256VFT4, MKL17Z256VLH4, MKL27Z128VLH4, MKL27Z128VFM4, MKL27Z128VMP4, MKL27Z128VFT4, MKL27Z256VMP4, MKL27Z256VFT4, MKL27Z256VFM4, MKL27Z256VLH4, MKL33Z128VLH4, MKL33Z128VMP4, MKL33Z256VLH4, MKL33Z256VMP4, MKL43Z128VLH4, MKL43Z128VMP4, MKL43Z256VMP4, **MKL43Z256VLH4,** MKL13Z32VFM4, MKL13Z32VFT4, MKL13Z32VLH4, MKL13Z32VLK4, MKL13Z32VMP4, MKL13Z64VFM4, MKL13Z64VFT4, MKL13Z64VLH4, MKL13Z64VLK4, MKL13Z64VMP4, MKL33Z32VFT4, MKL33Z32VLH4, MKL33Z32VLK4, MKL33Z32VMP4, MKL33Z64VFT4, MKL33Z64VLH4, MKL33Z64VLK4, MKL33Z64VMP4

Table continues on the next page...

Table 1. Supported MCU devices and development boards (continued)

FRDM-KL27Z (MKL27Z64VLH4)	MKL17Z32VDA4, MKL17Z32VFT4, MKL17Z32VLH4, MKL17Z32VFM4, MKL17Z32VMP4, MKL17Z64VDA4, MKL17Z64VMP4, MKL17Z64VFM4, MKL17Z64VFT4, MKL17Z64VLH4, MKL27Z32VDA4, MKL27Z32VLH4, MKL27Z32VFM4, MKL27Z32VMP4, MKL27Z32VFT4, MKL27Z64VDA4, MKL27Z64VFT4, MKL27Z64VMP4, MKL27Z64VFM4, **MKL27Z64VLH4**
FRDM-K66F (MK66FN2M0VMD18)	**MK66FN2M0VMD18**, MK66FN2M0VLQ18, MK66FX1M0VLQ18, MK66FX1M0VMD18
TWR-K64F120M (MK64FN1M0VMD12)	MK24FN1M0VDC12, MK24FN1M0VLL12, MK24FN1M0VLQ12, MK63FN1M0VLQ12, MK63FN1M0VMD12, MK64FN1M0VDC12, MK64FN1M0VLL12, MK64FN1M0VLQ12, **MK64FN1M0VMD12**, MK64FX512VDC12, MK64FX512VLL12, MK64FX512VLQ12, MK64FX512VMD12

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
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
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>
Multicore stack	<install_dir>/middleware/multicore_<version>
SDMMC card driver	<install_dir>/middleware/sdmmc_<version>
USB stack	<install_dir>/middleware/usb_<version>
WolfSSL stack	<install_dir>/middleware/wolfssl_<version>

Table continues on the next page...

Table 2. Release contents (continued)

Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source	<install_dir>/CMSIS
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools

6 Kinetis SDK Release Package

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

6.1 Kinetis device support

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

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

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

6.1.1 Kinetis board support

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

6.1.2 Demo applications and other examples

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

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

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

6.2 Middleware

6.2.1 USB stack

See the *USB Stack User's Guide* (document KSDKUSBSUG) 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
HUB	BELKIN F5U233 BELKIN F5U304 BELKIN F5U307 BELKIN F4U040 UNITEK Y-2151 Z-TEK ZK032A HYUNDAI HY-HB608
USB flash driver	ADATA C008 32 GB ADATA S102 8 G ADATA S102 16 G Verbatim STORE N GO USB Device 8 G Kingston DataTraveler DT101 G2 SanDisk Cruzer Blade 8 GB Unisplendour 1 G Imation 2 GB V-mux 2 GB Sanmina-SCI 128 M Corporate Express 1 G TOSHIBA THUHYBS-008G 8 G Transcend JF700 8 G Netac U903 16 G SSK SFD205 8 GB Rex 4 GB SAMSUNG USB3.0 16GB
USB card reader/adapter	SSK TF adapter

Table continues on the next page...

Table 3. Peripheral devices (continued)

	Kawau Multi Card Reader Kawau TF adapter Kawau SDHC card
Mouse	DELL MS111-P DELL M066U0A DELL MUAVDEL8 TARGUS AMU76AP DELL MD56U0 DELL MS111-T RAPOO M110
Keyboard	DELL SK8135 DELL SK8115

6.2.2 TCP/IP stack

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

6.2.3 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.

6.2.4 RTOS

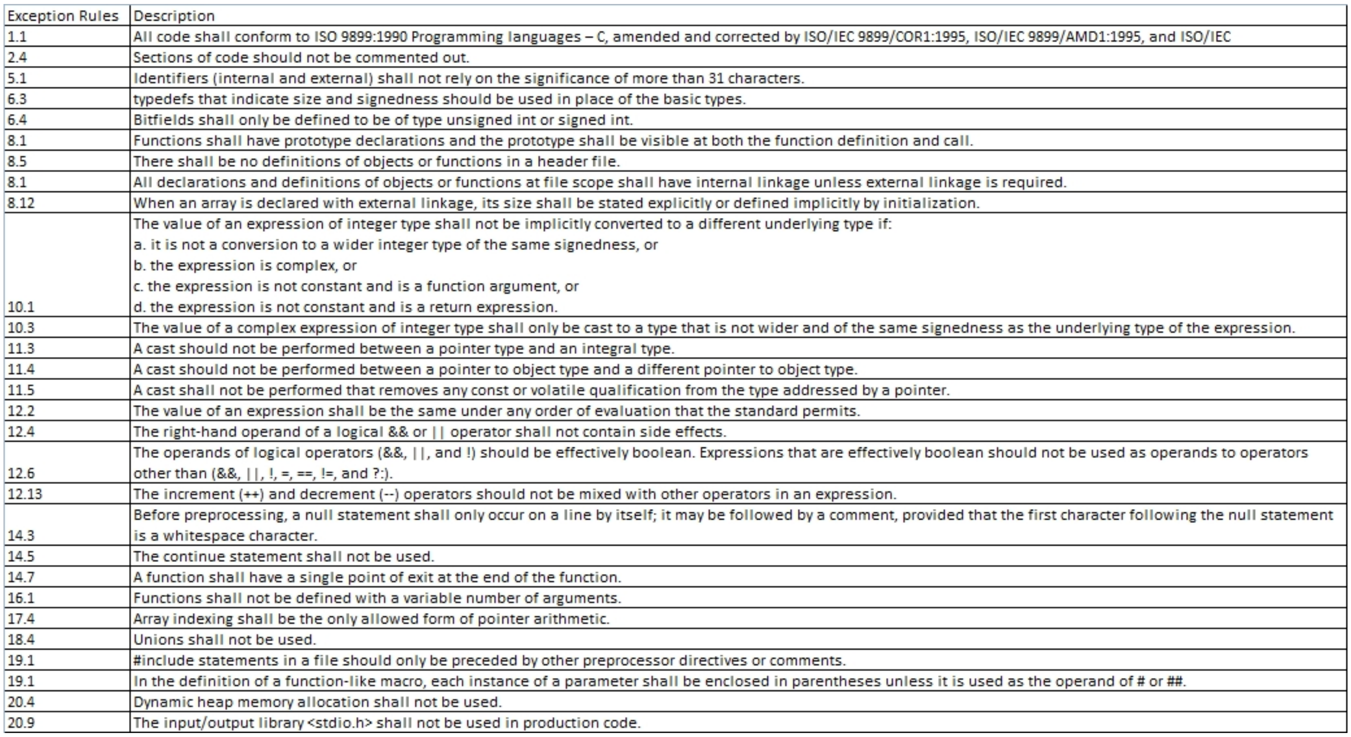
The Kinetis SDK is preintegrated with FreeRTOS OS, µC/OS-II OS, and µC/OS-III OS.

6.2.5 CMSIS

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

7 MISRA Compliance

All KSDK drivers and USB stack comply to MISRA 2004 rules with below exceptions.



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

Figure 1. MISRA exceptions

8 Known Issues

8.1 Maximum file path length in Windows® 7 Operating System

Windows 7 operating system imposes a 260 character maximum length for file paths. When installing the Kinetis SDK, place it in a directory close to the root to prevent file paths from exceeding the maximum character length specified by the Windows operating system. The recommended location is the `C:\nxp` folder.

8.2 USB HUB power supply

The external power supply of the USB HUB must be provided before it can be used. This is the result of the development board which is not designed to power a USB HUB and the devices connected to the HUB. Therefore, the external USB HUB that is connected to the development board should have its own power supply.

8.3 USB audio noise on the TWR-K22F120M and TWR-K21F120M NXP Tower System modules

A noise occurs when running the USB audio example on the TWR-K22F120M, TWR-K21F120M Tower System modules as a result of poor clock accuracy of the CSTCE8M00G55-R0 crystal oscillator.

8.4 HS USB device MSD demo issue

If the SD card is used as the storage medium, the functionality of USB might not work correctly if the USB hot plug action is performed while transferring files.

8.5 USB noise issue

A noise occurs when a song is playing on FreeRTOS USB host speaker example. The issue is caused by the software scheduler not being aligned with the hardware schedule as a result of the dynamic SOF threshold enablement on some boards.

8.6 FOPT programming for the FRDM-KL43Z NXP Freedom Development Platform

For the FRDM-KL43Z NXP Freedom platform, program a new FOPT value to the flash to ensure that SoC boots from flash. To enable the FOPT programming for IAR projects, provide the "--enable_config_write" parameter for the flash loader and save it as a new .board file. The FlashKLxx256ROM_with_config_write_enabled.board file in <install_dir>/examples/frdmkl43z is created for this purpose for all FRDM-KL43Z projects.

8.7 UART Peripheral driver support

Two kinds of UART modules are available for the MKL16, MKL26, MKL34, MKL36, MKL46, MKL25, MKL14, MKL15, MKL24, and MKW01 derivatives. However, these modules are supported by different drivers as follows:

- The UART0 module is supported by the LPSCI driver using UART0_IDX of 0 as the instance and UART0 for the register base address pointer
- The other UART module is supported by the UART driver using UARTn_IDX as the instance and UARTn for the register base address pointer

9 Revision History

This table summarizes revisions to this document.

Table 4. Revision history

Revision number	Date	Substantive changes
0	01/2016	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.