

Kinetis SDK v.1.3.0 Release Notes

1 Overview

These are the release notes for the Freescale Kinetis Software Development Kit (KSDK) 1.3.0. The core of the Kinetis SDK is a set of drivers architected in two layers: the Hardware Abstraction Layer (HAL) and the Peripheral Driver Layer.

The HAL abstracts the hardware register accesses into a set of stateless functional primitives which provide the building blocks for high-level Peripheral Drivers or applications. The Peripheral Driver Layer implements use-case driven drivers by utilizing one or more HAL layer components, system services, and possibly other Peripheral Drivers.

The Kinetis SDK includes a set of example applications demonstrating the use of the Peripheral Drivers and other integrated software modules such as a Real-Time Operating System (RTOS) through an RTOS abstraction layer. The Kinetis SDK also integrates middleware such as the Freescale USB stack to provide an easy-to-use Software Development Kit for Kinetis microcontroller (MCU) product families.

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

Contents

1	Overview.....	1
2	What Is New.....	1
3	Development Tools.....	2
4	Supported Development Systems.....	3
5	Release Contents.....	6
6	Kinetis SDK Release Overview.....	7
7	Known Issues.....	11
8	Change Log.....	13
9	Revision History.....	15

Development Tools

This section describes the major changes and new features implemented in this release.

- Added device family support:

Table 1. Added device families

K22FA12	K80F25615	K81F25615
K82F25615	KL13Z644	KL33Z644
KM34Z7	KV10Z1287	KV11Z7
KW20Z4	KW30Z4	KW40Z4

- Added Peripheral support:
 - AFE
 - CMT
 - IRTC
 - LTC
 - MMAU
 - QSPI
 - QUADTMR
 - SDRAMC
 - SLCD
 - SMARTCARD
 - TRNG

These are the updates for middleware:

- FreeRTOS:
 - Updated to 8.2.0
- EMV
 - Added.
- USB
 - Enabled Device Video class.
 - Removed the board-related code implementation from the USB library. The USB library is SoC-specific and not board-specific. See the USB_Class_<Class_name>_Init function for the detailed device stack API change, and the usb_host_init function for the host stack API change.
 - Removed USBCFG_DEV_COMPOSITE from usb_device_config.h. In the previous release, the users had to set this MACRO to 1 to enable the composite device examples. This is no longer needed.

3 Development Tools

The Kinetis SDK 1.3.0 was compiled and tested with these development tools:

- Kinetis Design Studio IDE v3.0
- IAR Embedded Workbench for ARM® version 7.40.3
- MDK-ARM Microcontroller Development Kit (Keil)® 5.15
- Makefiles support with GCC revision 4.9-2015-q1-update from ARM Embedded
- Atollic® TrueSTUDIO® 5.3.1

4 Supported Development Systems

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

Table 2. Supported MCU devices and development boards

Development boards	Kinetis MCU devices
FRDM-KL03Z (Rev. B1)	MKL03Z8VFG4, MKL03Z16VFG4, MKL03Z32VFG4, MKL03Z32CAF4, MKL03Z8VFK4, MKL03Z16VFK4, MKL03Z32VFK4
TWR-K22F120M (Rev. C) FRDM-K22F (Rev. D) FRDM-KV31F (Rev. X1)	MK22FN512VDC12, MK22FN512VLL12, MK22FN512VLH12 , MK22FN512VMP12, MK22FN512CAP12, MK02FN128VLH10, MK02FN128VLF10, MK02FN128VFM10, MK02FN64VFM10, MK02FN64VLF10, MK02FN64VLH10, MK22FN128CAH12, MK22FN128VDC10, MK22FN128VLH10, MK22FN128VLL10, MK22FN128VMP10, MK22FN256CAH12, MK22FN256VDC12, MK22FN256VLH12, MK22FN256VLL12, MK22FN256VMP12
TWR-KV31F120M (Rev. B)	MKV31F512VLL12 , MKV31F512VLH12, MKV30F128VFM10, MKV30F128VLF10, MKV30F128VLH10, MKV30F64VFM10, MKV30F64VLF10, MKV30F64VLH10, MKV31F128VLH10, MKV31F128VLL10, MKV31F256VLH12, MKV31F256VLL12
TWR-K24F120M (Rev. A)	MK24FN256VDC12
TWR-K64F120M (Rev. B1) FRDM-K64F (Rev. C)	MK24FN1M0VDC12, MK24FN1M0VLL12, MK24FN1M0VLQ12, MK63FN1M0VLQ12, MK63FN1M0VMD12, MK64FN1M0VDC12, MK64FN1M0VLL12, MK64FN1M0VLQ12, MK64FN1M0VMD12, MK64FX512VDC12, MK64FX512VLL12, MK64FX512VLQ12, MK64FX512VMD12
TWR-KV10Z32 (Rev. B3) FRDM-KV10Z	MKV10Z32VLF7 , MKV10Z32VLC7, MKV10Z32VFM7, MKV10Z16VLF7, MKV10Z16VLC7, MKV10Z16VFM7
TWR-K60D100M (Rev. B)	MK10DN512VLK10, MK10DN512VLL10, MK10DN512VLQ10, MK10DN512VMC10, MK10DN512VMD10, MK10DX128VLQ10, MK10DX128VMD10, MK10DX256VLQ10, MK10DX256VMD10, MK20DN512VLK10, MK20DN512VLL10, MK20DN512VLQ10, MK20DN512VMC10, MK20DN512VMD10, MK20DX128VLQ10, MK20DX128VMD10, MK20DX256VLK10, MK20DX256VLQ10, MK20DX256VMC10, MK20DX256VMD10, MK20DX256VLL10, MK30DN512VLK10, MK30DN512VLL10, MK30DN512VLQ10, MK30DN512VMC10, MK30DN512VMD10, MK30DX128VLQ10, MK30DX128VMD10, MK30DX256VLQ10, MK30DX256VMD10, MK40DN512VLK10, MK40DN512VLL10, MK40DN512VLQ10, MK40DN512VMC10, MK40DN512VMD10, MK40DX128VLQ10, MK40DX128VMD10, MK40DX256VLQ10, MK40DX256VMD10, MK50DN512CLL10, MK50DN512CLQ10, MK50DN512CMC10, MK50DN512CMD10,

Table continues on the next page...

Table 2. Supported MCU devices and development boards (continued)

	MK50DX256CLK10, MK50DX256CCL10, MK50DX256CMC10, MK50DX256CMD10, MK51DN256CLQ10, MK51DN256CMD10, MK51DN512CCL10, MK51DN512CLQ10, MK51DN512CMC10, MK51DN512CMD10, MK51DX256CLK10, MK51DX256CCL10, MK51DX256CMC10, MK52DN512CLQ10, MK52DN512CMD10, MK53DN512CLQ10, MK53DN512CMD10, MK53DX256CLQ10, MK53DX256CMD10, MK60DN256VLL10, MK60DN256VLQ10, MK60DN256VMC10, MK60DN256VMD10, MK60DN512VLL10, MK60DN512VLQ10, MK60DN512VMC10, MK60DN512VMD10 , MK60DX256VLL10, MK60DX256VLQ10, MK60DX256VMC10, MK60DX256VMD10
FRDM-KL46Z (Rev. C)	MKL16Z256VLH4, MKL16Z256VMP4, MKL26Z256VLH4, MKL26Z256VMP4, MKL26Z128VLL4, MKL26Z256VLL4, MKL26Z128VMC4, MKL26Z256VMC4, MKL34Z64VLH4, MKL34Z64VLL4, MKL36Z64VLH4, MKL36Z128VLH4, MKL36Z256VLH4, MKL36Z256VMP4, MKL36Z64VLL4, MKL36Z128VLL4, MKL36Z256VLL4, MKL36Z128VMC4, MKL36Z256VMC4, MKL46Z128VLH4, MKL46Z256VLH4, MKL46Z256VMP4, MKL46Z128VLL4, MKL46Z256VLL4 , MKL46Z128VMC4, MKL46Z256VMC4
TWR-K21D50M (Rev. B)	MK11DN512AVLK5, MK11DN512AVMC5, MK11DX128AVLK5, MK11DX128AVMC5, MK11DX256AVLK5, MK11DX256AVMC5, MK21DN512AVLK5, MK21DN512AVMC5 , MK21DX128AVLK5, MK21DX128AVMC5, MK21DX256AVLK5, MK21DX256AVMC5
TWR-K21F120M (Rev. A)	MK21FN1M0AVLQ12, MK21FN1M0AVMC12 , MK21FN1M0AVMD12, MK21FX512AVLQ12, MK21FX512AVMC12, MK21FX512AVMD12, MK22FN1M0AVLH12, MK22FN1M0AVLK12, MK22FN1M0AVLL12, MK22FN1M0AVLQ12, MK22FN1M0AVMC12, MK22FN1M0AVMD12, MK22FX512AVLH12, MK22FX512AVLK12, MK22FX512AVLL12, MK22FX512AVLQ12, MK22FX512AVMC12, MK22FX512AVMD12
TWR-KW24D512 (Rev. C) USB-KW24D512 (Rev. A3) FRDM-KW24 (Rev. X1)	MKW21D256VHA5, MKW21D512VHA5, MKW22D512VHA5, MKW24D512VHA5
FRDM-KL43Z (Rev. A) TWR-KL43Z48M (Rev. C)	MKL17Z128VFM4, MKL17Z256VFM4, MKL17Z128VFT4, MKL17Z256VFT4, MKL17Z128VLH4, MKL17Z256VLH4, MKL17Z128VMP4, MKL17Z256VMP4, MKL27Z128VFM4, MKL27Z256VFM4, MKL27Z128VFT4, MKL27Z256VFT4, MKL27Z128VLH4, MKL27Z256VLH4, MKL27Z128VMP4, MKL27Z256VMP4, MKL33Z128VLH4, MKL33Z256VLH4, MKL33Z128VMP4, MKL33Z256VMP4, MKL43Z128VLH4, MKL43Z256VLH4 , MKL43Z128VMP4, MKL43Z256VMP4, MKL13Z32VFM4, MKL13Z32VFT4, MKL13Z32VLH4, MKL13Z32VVK4, MKL13Z32VMP4, MKL13Z64VFM4, MKL13Z64VFT4, MKL13Z64VLH4, MKL13Z64VVK4,

Table continues on the next page...

Table 2. Supported MCU devices and development boards (continued)

	MKL13Z64VMP4, MKL33Z32VFT4, MKL33Z32VLH4, MKL33Z32VLK4, MKL33Z32VMP4, MKL33Z64VFT4, MKL33Z64VLH4, MKL33Z64VLK4, MKL33Z64VMP4
FRDM-KL27Z (Rev. A)	MKL17Z32VFM4, MKL17Z64VFM4, MKL17Z32VDA4, MKL17Z64VDA4, MKL17Z32VFT4, MKL17Z64VFT4, MKL17Z32VMP4, MKL17Z64VMP4, MKL17Z32VLH4, MKL17Z64VLH4, MKL27Z32VFM4, MKL27Z64VFM4, MKL27Z32VDA4, MKL27Z64VDA4, MKL27Z32VFT4, MKL27Z64VFT4, MKL27Z32VMP4, MKL27Z64VMP4, MKL27Z32VLH4, MKL27Z64VLH4
TWR-K65F180M (Rev. C)	MK26FN2M0VMD18, MK26FN2M0VLQ18, MK26FN2M0CAC18, MK26FN2M0VMI18, MK65FN2M0CAC18, MK65FN2M0VMI18 , MK65FX1M0VMI18, MK65FX1M0CAC18, MK66FN2M0VLQ18, MK66FN2M0VMD18, MK66FX1M0VLQ18, MK66FX1M0VMD18
FRDM-KL02Z (Rev. B)	MKL02Z8VFG4, MKL02Z16VFG4, MKL02Z32VFG4, MKL02Z32CAF4, MKL02Z16VFK4, MKL02Z32VFK4, MKL02Z16VFM4, MKL02Z32VFM4
FRDM-KL25Z (Rev. F)	MKL14Z32VFM4, MKL14Z64VFM4, MKL14Z32VFT4, MKL14Z64VFT4, MKL14Z32VLH4, MKL14Z64VLH4, MKL14Z32VLK4, MKL14Z64VLK4, MKL15Z32VFM4, MKL15Z64VFM4, MKL15Z128VFM4, MKL15Z128CAD4, MKL15Z32VFT4, MKL15Z64VFT4, MKL15Z128VFT4, MKL15Z32VLH4, MKL15Z64VLH4, MKL15Z128VLH4, MKL15Z32VLK4, MKL15Z64VLK4, MKL15Z128VLK4, MKL24Z32VFM4, MKL24Z64VFM4, MKL24Z32VFT4, MKL24Z64VFT4, MKL24Z32VLH4, MKL24Z64VLH4, MKL24Z32VLK4, MKL24Z64VLK4, MKL25Z32VFM4, MKL25Z64VFM4, MKL25Z128VFM4, MKL25Z32VFT4, MKL25Z64VFT4, MKL25Z128VFT4, MKL25Z32VLH4, MKL25Z64VLH4, MKL25Z128VLH4, MKL25Z32VLK4, MKL25Z64VLK4, MKL25Z128VLK4
MRB-KW019032NA (Rev. D) MRB-KW019030JA (Rev. B) MRB-KW019032EU (Rev. D)	MKW01Z128CHN4
FRDM-KL26Z (Rev. B)	MKL16Z32VFM4, MKL16Z64VFM4, MKL16Z128VFM4, MKL16Z32VFT4, MKL16Z64VFT4, MKL16Z128VFT4, MKL16Z32VLH4, MKL16Z64VLH4, MKL16Z128VLH4, MKL26Z32VFM4, MKL26Z64VFM4, MKL26Z128VFM4, MKL26Z128CAL4, MKL26Z32VFT4, MKL26Z64VFT4, MKL26Z128VFT4, MKL26Z32VLH4, MKL26Z64VLH4, MKL26Z128VLH4
TWR-K80F150M (Rev. B) TWR-K81F150M (Rev. B) FRDM-K82F (Rev. B)	MK80FN256VLQ15, MK80FN256VLL15 , MK80FN256VDC15 , MK80FN256CAx15, MK81FN256CDC15 , K81FN256CLL15, MK81FN256CAx151, MK81FN256CLQ152, MK82FN256VLQ15, MK82FN256CLL15, MK82FN256CDC15, MK82FN256CAx15
TWR-KM34Z75M (Rev. A)	MKM34Z256VLL7, MKM34Z256VLQ7
TWR-KV11Z75M (Rev. BX3)	MKV11Z128VLH7, MKV11Z128VLF7, MKV11Z128VLC7, MKV11Z128VFM7, MKV11Z64VLH7 , MKV11Z64VLF7, MKV11Z64VLC7, MKV11Z64VFM7, MKV10Z64VLH7,

Table continues on the next page...

Table 2. Supported MCU devices and development boards (continued)

	MKV10Z64VLF7, MKV10Z64VLC7, MKV10Z64VFM7, MKV10Z128VLH7, MKV10Z128VLF7, MKV10Z128VLC7, MKV10Z128VFM7
FRDM-KW40Z (Rev. A) USB-KW40Z USB-KW40Z-K22F	MKW40Z160VHT4 , MKW30Z160VHM4, MKW20Z160VHT4

5 Release Contents

This table describes the release contents.

Table 3. Release contents

Deliverable	Location
Examples	<install_dir>/examples/...
Demo applications	<install_dir>/examples/<board_name>/demo_apps/...
USB Demo applications	<install_dir>/examples/<board_name>/demo_apps/usb/...
Driver examples	<install_dir>/examples/<board_name>/driver_examples/...
Documentation	<install_dir>/doc/...
MQX RTOS Documentation	<install_dir>/doc/rtos/mqx/...
USB Documentation	<install_dir>/doc/usb/...
MQX RTCS Documentation	<install_dir>/doc/tcpip/mqx_rtcs/...
lwIP Documentation	<install_dir>/doc/tcpip/lwip/...
MQX MFS Documentation	<install_dir>/doc/filesystem/mqx_mfs/...
Projects to build libraries	<install_dir>/lib/...
Middleware	<install_dir>/middleware/...
TCP/IP stacks	<install_dir>/middleware/tcpip/...
File system	<install_dir>/middleware/filesystem/...
Driver library, startup code and utilities	<install_dir>/platform/...
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source	<install_dir>/platform/CMSIS/...
Composite drivers for SD-card and Soundcard support	<install_dir>/platform/composite/...
Linker control files for each supported tool chain	<install_dir>/platform/devices/<soc_name>/linker/...
SoC header files, Extension header files and feature header files	<install_dir>/platform/devices/<device_name>/include files
CMSIS-compliant startup code	<install_dir>/platform/devices/<soc_name>/startup/...
Peripheral Drivers	<install_dir>/platform/drivers/...
Hardware Abstraction Layer	<install_dir>/platform/hal/...
OS Abstraction for Bare Metal and RTOS	<install_dir>/platform/osa/...
System Services such as clock manager, interrupt manager, unified hardware timer, and low power manager	<install_dir>/platform/system/...

Table continues on the next page...

Table 3. Release contents (continued)

Utilities such as debug console	<install_dir>/platform/utilities/...
RTOS Kernel Code, RTOS abstraction implementations, and RTOS kernel folders	<install_dir>/rtos/...
Processor Expert service pack and cmake toolchain files	<install_dir>/tools
USB stack and USB projects to build libraries	<install_dir>/usb/...
Utilities such as shell	<install_dir>/utilities/...

6 Kinetis SDK Release Overview

The Kinetis SDK is intended for use with Freescale's Kinetis MCU product family based on the ARM Cortex-M series architectures. The release consists of:

- Kinetis MCU platform support
- Demo applications/Driver examples
- The FatFs FAT File System
- USB Host and Device OTG stacks
- lwIP TCP/IP networking stack
- RTOS support components
- Documentation (Kinetis SDK API Reference Manual and various user's guides)

6.1 Kinetis MCU platform support

The Kinetis SDK platform directory contains the startup code, operating system abstraction, system services, driver libraries for peripherals, header files, linker files, and utilities such as the debug console implementation.

6.1.1 Startup code

The Kinetis SDK includes simple CMSIS-compliant startup code for the supported Kinetis MCUs which efficiently deliver the code execution to the main() function. An application can either include the startup code directly in the project build environment or include a prebuilt startup code library for a cleaner project build environment.

6.1.2 Operating system abstraction

The drivers are designed to work with or without an operating system through the Operating System Abstraction layer (OSA). The OSA defines a common set of services that abstract most of the OS kernel functionalities. The OSA either maps an OSA service to the target OS function, or implements the service when no OS is used (bare metal) or when the service does not exist in the target OS. The Kinetis SDK implements the OSA for Freescale MQX™ RTOS, FreeRTOS, μC/OS-II, μC/OS-III, and for OS-less "bare metal" usage. The bare metal OSA implementation is selected as the default option.

6.1.3 System services

The system services contain a set of software entities that can be used either by the Peripheral Drivers or with the HAL to build either Peripheral Drivers or an application directly. The system services include the interrupt manager, clock manager, low power manager, and the unified hardware timer interface.

6.1.4 Driver library

The Kinetis SDK provides a set of drivers for the Kinetis MCU product family on-chip peripherals. The drivers are designed and implemented around the peripheral hardware blocks rather than for a specific Kinetis MCU, and work with or without an OS through the OS Abstraction layer. The drivers are architected into two layers: the Hardware Abstraction Layer and the Peripheral Driver Layer.

The HAL is designed to abstract hardware register accesses into functional accesses. It is stateless and is intended to cover the entire hardware functionality.

The Peripheral Drivers are built on top of the HAL to provide a set of easy-to-use interfaces that handle high-level data and stateful transactions. They are designed for the most common use cases identified for the underlying hardware block and are reasonably efficient in terms of memory and performance. They are written in C language and can be easily ported from product to product as they are designed to be initialized at runtime based on the driver configuration passed in by the user. In most cases, the Peripheral Drivers can be used as is. However, if the Peripheral Driver does not address a particular target use case, it can either be modified/enhanced or completely rewritten to meet the target functionality and other requirements. In this case, the existing Peripheral Driver can be used as a reference to build a custom driver based on the HAL. For more details, see the Architectural Overview chapter in the Kinetis SDK API Reference Manual (document KSDK13APIRM).

Detailed implementation of hardware peripheral functionality, for both the HAL and Peripheral Driver, is implemented in stages. For example, the current version of the UART driver does not support modem control and smart card features. Likewise, the current version of the I2C driver does not support the SMBUS feature. The features which are missing from the current driver versions may be implemented in future releases.

6.1.5 Header files

The Kinetis SDK devices directory contains device-specific header files which provide direct access to the Kinetis MCU peripheral registers. Each supported Kinetis MCU device in the Kinetis SDK has an overall System-on-Chip (SoC) memory-mapped header file. In addition to the overall SoC memory-mapped header file, the Kinetis SDK includes extension header files and feature header files for each peripheral instantiated on the Kinetis MCU. Along with the SoC header files, peripheral extension header files, and feature header files the Kinetis SDK CMSIS directory includes common CMSIS header files for the ARM Cortex-M core and DSP library from the ARM CMSIS version 4.2 release.

6.1.6 Linker files

The Kinetis SDK devices directory contains linker control files (or simply linker files) for each supported tool chain and Kinetis MCU device.

6.1.7 Utilities

The utilities directory contains useful software utilities such as a debug console.

6.2 Demo applications

The demo applications demonstrate the usage of the driver libraries and other integrated software solutions on supported development systems.

For details, see the *Kinetis SDK v. 1.3 Demo Applications User's Guide* (document KSDK13DEMOUG).

6.3 Driver examples

The driver examples demonstrate configuring drivers by passing configuration data to the API functions.

For details, see the *Kinetis SDK v.1.3 Demo Applications User's Guide* (document KSDK13DEMOUG).

6.4 Other integrated software solutions

The Kinetis SDK is designed for easy integration with other software solutions such as OS kernels, USB stack, TCP/IP stack, and file systems.

6.4.1 USB stack

A Freescale USB stack is integrated with the Kinetis SDK and was tested both with and without an OS through the OS abstraction layer. For details, see the *Integration of the USB Stack and Kinetis SDK* (document USBKSDKUG).

6.4.2 Peripheral devices

Peripheral devices that are verified are listed below. If a peripheral device is not in the list, it does not indicate that this device cannot be supported by the USB Host Stack. Instead, it indicates that this device has not been verified.

6.4.2.1 HUB

- TARGUS PAUK10U
- BELKIN F5U233
- BELKIN F5U304
- BELKIN F5U307
- BELKIN F4U022
- BELKIN F4U040
- UNITEK Y-2151
- Z-TEK ZK032A
- HYUNDAI HY-HB608

6.4.2.2 USB flash driver

- ADATA C008 32 GB
- ADATA S102 8 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

6.4.2.3 USB card reader/adapter

- SSK TF adapter
- Kawai Multi Card Reader
- Kawai TF adapter
- Kawai SDHC card

6.4.2.4 Mouse

- DELL MS111-P
- DELL M066U0A
- DELL MUAVDEL8
- TARGUS AMU76AP
- DELL MD56U0
- DELL MS111-T
- RAPOO M110

6.4.2.5 Keyboard

- DELL SK8135
- DELL SK8115

6.4.3 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).

6.4.4 File System

A FAT 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.

6.4.5 RTOS

The Kinetis SDK is pre-integrated with Freescale MQX RTOS, FreeRTOS, μ C/OS-II, and μ C/OS-III. OS abstraction layers are implemented for these RTOSes.

7 Known Issues

7.1 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:\Freescale` folder.

7.2 No spaces in the Kinetis SDK installation

The Freescale MQX RTOS build uses batch files, which do not work when there are spaces in the file path.

7.3 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.

7.4 USB audio noise on the TWR-K22F120M, TWR-K21D50M, TWR-K21F120M, and TWR-K24F120M Freescale Tower System modules

A noise occurs when running the USB audio example on the TWR-K22F120M, TWR-K21D50M, TWR-K21F120M, and TWR-K24F120M Tower System modules as a result of poor clock accuracy of the CSTCE8M00G55-R0 crystal oscillator.

7.5 cdc_serial/OTG example on KSDK MQX RTOS

Follow these steps to run the host `cdc_serial/OTG` example on the KSDK MQX RTOS.

1. Add two lines to the `rtos\mqx\config\mcu<soc_name>\mqx_sdk_config.h`

Known Issues

```
#define BSPCFG_ENABLE_IO_SUBSYSTEM (0)
```

```
#define printf debug_printf
```

2. Re-compile all libraries, such as `mqx_<board_name>`, `mqx_stdlib_<board_name>`, `ksdk_mqx_lib`, and `usbh_sdk_<board_name>_mqx`.

7.6 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.

7.7 Trimming the Clock in software initialization for TWR-KW24D512 Tower System module with the P24K9V N62J1N42H SoC embedded

Run the clock trimming for the TWR-KW24D512 Tower System module with the P24K8V N62J1N42H SoC embedded.

7.8 FOPT programming for the FRDM-KL43Z Freescale Freedom Development Platform

For the FRDM-KL43Z Freescale 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.

7.9 UART HAL/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 HAL/drivers as follows:

- The UART0 module is supported by the LPSCI HAL/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 HAL/driver using `UARTn_IDX` as the instance and `UARTn` for the register base address pointer

7.10 K81F SoCs are not supported by third party tools

Because K81F SoCs are not supported by third party tools, the users need to configure the device type in projects with K82F SoCs.

7.11 Update P&E Micro and J-Link drivers for K82F SoCs

The P&E and J-Link debugger support for K82F is not integrated into the third party toolchains. Users need to download the latest P&E, J-Link software from links below to debug the K82F. J-Link version must be greater than or equal to v5.02. P&E release must be later or equal to 2015/6/9.

- [P&E](#)
- [P&E eclipse update link](#)
- [J-Link](#)

7.12 K80F, K81F, and K82F header file issue

The header files for K80F, K81F, and K82F SoCs do not contain definitions for the PIT0_LTMR64H and PIT0_LTMR64L as specified in the reference manual (document K80P121M150SF5RM). If necessary, users can manually update the header files based on the header files for K65F SoC. To use the related driver/hal functions, set the macro, FSL_FEATURE_PIT_HAS_LIFETIME_TIMER in MK8x_features.h header file, to 1. The header file also contains several OTG-related bitfields not described in the reference manual rev. 3. These bitfields are not used in the KSDK code and can be safely ignored.

7.13 Windows OS installer issue

Windows OS users should use a PC/laptop with at least 8 GB of RAM memory for faster installation results. If the available RAM memory is less than 8 GB, the installation can take approximately 45-90 minutes.

8 Change Log

These are major driver API updates:

- DMA
 - [Added]: dma_status_t DMA_DRV_SetDestTransferSize(dma_channel_t *chn, uint32_t transferSize).
 - [Added]: dma_status_t DMA_DRV_SetSourceTransferSize(dma_channel_t *chn, uint32_t transferSize).
- DSPI
 - [Added]: fsl_dspi_dma_master_driver.c/fsl_dspi_dma_slave_driver.c to support DSPI using DMA on Master/Slave side.
- eDMA
 - [Added]: static inline edma_status_t EDMA_DRV_PrepareDescriptorMinorLink(edma_software_tcd_t *stcd, uint32_t linkChn).
 - [Added]: static inline edma_status_t EDMA_DRV_TriggerChannelStart(edma_chn_state_t *chn).
- PWM
 - [Added]: void PWM_HAL_EnableInterrupts(PWM_Type *base, pwm_module_t subModuleNum, uint32_t eventmask).
 - [Added]: void PWM_HAL_DisableInterrupts(PWM_Type *base, pwm_module_t subModuleNum, uint32_t eventmask).
 - [Added]: void PWM_HAL_ClearStatus(PWM_Type *base, pwm_module_t subModuleNum, uint32_t eventmask).
 - [Removed]: static inline void PWM_HAL_SetFaultIntCmd(PWM_Type *base, pwm_fault_input_t fault, bool val)
 - [Removed]: static inline void PWM_HAL_ClearFaultFlags(PWM_Type *base, pwm_fault_input_t fault).

Change Log

- RNGA
 - [Added]: `rnga_status_t RNGA_DRV_GetRandomData(uint32_t instance, void *data, uint32_t data_size).`
- RTC
 - [Removed]: `void RTC_HAL_Enable(RTC_Type *rtcBase).`
 - [Removed]: `void RTC_HAL_Disable(RTC_Type *rtcBase).`
- SAI
 - [Added]: `void SAI_DRV_TxStopModule(uint32_t instance).`
 - [Added]: `void SAI_DRV_RxStopModule(uint32_t instance).`
 - [Added]: `void SAI_DRV_TxSetIntCmd(uint32_t instance, bool enable).`
 - [Added]: `void SAI_DRV_RxSetIntCmd(uint32_t instance, bool enable).`
 - [Added]: `void SAI_DRV_TxSetDmaCmd(uint32_t instance, bool enable).`
 - [Added]: `void SAI_DRV_RxSetDmaCmd(uint32_t instance, bool enable).`
- TPM
 - [Added]: `void TPM_DRV_SetClock(uint32_t instance, tpm_clock_mode_t clock, tpm_clock_ps_t clockPs):`
Update the prototype.
- I2C
 - [Added]: `static inline void I2C_HAL_SetManualACKCmd(I2C_Type *base, bool enable).`
 - [Added]: SMBUS support.
- LPUART
 - [Added]: `lpuart_status_t LPUART_HAL_SetTxFifoCmd(LPUART_Type * base, bool enable).`
 - [Added]: `lpuart_status_t LPUART_HAL_SetRxFifoCmd(LPUART_Type * base, bool enable).`
 - [Added]: `static inline uint8_t LPUART_HAL_GetTxFifoSize(LPUART_Type * base).`
 - [Added]: `static inline uint8_t LPUART_HAL_GetRxFifoSize(LPUART_Type * base).`
 - [Added]: `lpuart_status_t LPUART_HAL_FlushTxFifo(LPUART_Type * base).`
 - [Added]: `lpuart_status_t LPUART_HAL_FlushRxFifo(LPUART_Type * base).`
 - [Added]: `static inline bool LPUART_HAL_IsTxFifoEmpty(LPUART_Type * base).`
 - [Added]: `lpuart_status_t LPUART_HAL_SetRxIdleEmpty(LPUART_Type * base,`
`lpuart_rx_idle_empty_config_t config).`
 - [Added]: `static inline lpuart_rx_idle_empty_config_t LPUART_HAL_GetRxIdleEmpty(LPUART_Type * base).`
 - [Added]: `lpuart_status_t LPUART_HAL_SetTxFifoWatermark(LPUART_Type * base, uint8_t watermark).`
 - [Added]: `static inline uint8_t LPUART_HAL_GetTxFifoWatermark(LPUART_Type * base).`
 - [Added]: `static inline uint8_t LPUART_HAL_GetTxDatawordCountInFifo(LPUART_Type * base).`
 - [Added]: `lpuart_status_t LPUART_HAL_SetRxFifoWatermark(LPUART_Type * base, uint8_t watermark).`
 - [Added]: `static inline uint8_t LPUART_HAL_GetRxDatawordCountInFifo(LPUART_Type * base).`
 - [Added]: `static inline uint8_t LPUART_HAL_GetRxFifoWatermark(LPUART_Type * base).`
 - [Updated]: Add `txFifoEntryCount` in structure `LpuartState`.
- MCG
 - [Added]: `static inline void CLOCK_HAL_SetClksFrdivIrefs(MCG_Type * base, mcg_clkout_src_t clks, uint8_t frdiv, mcg_fll_src_t irefs).`
 - [Added]: `void CLOCK_HAL_UpdateInternalRefClk(MCG_Type *base, mcg_irc_mode_t ircs, uint8_t frdiv, bool enableInStop).`
 - [Added]: `void CLOCK_HAL_EnablePll0InFillMode(MCG_Type * base, bool enableInStop).`
 - [Added]: `static inline void CLOCK_HAL_SetPllRefSel0Mode(MCG_Type * base, mcg_pll_ref_clock_source_t setting).`
 - [Updated]: Remove parameter of `outClkFreq`. `mcg_mode_error_t CLOCK_HAL_SetFeeMode(MCG_Type * base, mcg_oscsel_select_t oscselVal, uint8_t frdivVal, mcg_dm32_select_t dm32, mcg_dco_range_select_t drs, void (* fllStableDelay)(void)).`
 - [Updated]: Remove parameter of `outClkFreq`. `mcg_mode_error_t CLOCK_HAL_SetFeiMode(MCG_Type * base, mcg_dco_range_select_t drs, void (* fllStableDelay)(void)).`
 - [Updated]: Remove parameter of `outClkFreq`. `mcg_mode_error_t CLOCK_HAL_SetFbiMode(MCG_Type * base, mcg_dco_range_select_t drs, mcg_irc_mode_t ircSelect, uint8_t frdivVal, void (* fllStableDelay)(void)).`
 - [Updated]: Remove parameter of `outClkFreq`. `mcg_mode_error_t CLOCK_HAL_SetBlpiMode(MCG_Type * base, uint8_t frdivVal, mcg_irc_mode_t ircSelect).`
 - [Updated]: Remove parameter of `outClkFreq`. `mcg_mode_error_t CLOCK_HAL_SetBlpeMode(MCG_Type * base, mcg_oscsel_select_t oscselVal).`

- [Updated]: Remove parameter of outClkFreq. mcg_mode_error_t CLOCK_HAL_SetPbeMode(MCG_Type * base, mcg_oscsel_select_t oscselVal, mcg_pll_ref_clock_source_t pll32kRef, uint8_t frdivVal).
- [Updated]: Remove parameter of outClkFreq. mcg_mode_error_t CLOCK_HAL_SetPbeMode(MCG_Type * base, mcg_oscsel_select_t oscselVal, mcg_pll_clk_select_t pllcsSelect, uint8_t prdivVal, uint8_t vdivVal).
- [Updated]: Remove parameter of outClkFreq. mcg_mode_error_t CLOCK_HAL_SetPeeMode(MCG_Type * base).
- [Updated]: Remove parameter of outClkFreq. mcg_mode_error_t CLOCK_HAL_SetPbiMode(MCG_Type * base, mcg_irc_mode_t ircSelect, uint8_t fcrdivVal).
- [Updated]: Remove parameter of outClkFreq. mcg_mode_error_t CLOCK_HAL_SetPeiMode(MCG_Type * base).
- [Updated]: Remove parameter of outClkFreq. mcg_mode_error_t CLOCK_HAL_SetFbeMode(MCG_Type * base, mcg_oscsel_select_t oscselVal, uint8_t frdivVal, mcg_dmx32_select_t dmx32, mcg_dco_range_select_t drs, void (* flIStableDelay)(void)).
- FlexIO
 - [Updated]: Add dma_request_source_t baseSource in structure flexioI2SConfig.
 - [Updated]: Add dma_request_source_t baseSource in structure flexioI2SHandler.
 - [Updated]: Remove edmaTxTcd, edmaRxTcd in structure flexio_spi_state.
 - [Removed]: void FLEXIO_I2C_DRV_Deinit(flexio_i2c_state_t *i2cState).
 - [Added]: void FLEXIO_I2C_DRV_MasterDeinit(flexio_i2c_state_t *i2cState).
 - [Updated]: Add pwidth in structure _flexio_shifter_config_t.
 - [Added]: static inline uint32_t FLEXIO_HAL_GetShifterBufferNibbleByteSwapped(FLEXIO_Type * base, uint32_t shifterIdx).
 - [Added]: static inline void FLEXIO_HAL_SetShifterBufferNibbleByteSwapped(FLEXIO_Type * base, uint32_t shifterIdx, uint32_t value).
 - [Added]: static inline uint32_t FLEXIO_HAL_GetShifterBufferHalfWordSwapped(FLEXIO_Type * base, uint32_t shifterIdx).
 - [Added]: static inline void FLEXIO_HAL_SetShifterBufferHalfWordSwapped(FLEXIO_Type * base, uint32_t shifterIdx, uint32_t value).
 - [Added]: static inline uint32_t FLEXIO_HAL_GetShifterBufferNibbleSwapped(FLEXIO_Type * base, uint32_t shifterIdx).
 - [Added]: static inline void FLEXIO_HAL_SetShifterBufferNibbleSwapped(FLEXIO_Type * base, uint32_t shifterIdx, uint32_t value).

9 Revision History

This table summarizes revisions to this document.

Table 4. Revision history

Revision number	Date	Substantive changes
0	09/2015	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.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Tower is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM, ARM powered logo, Keil, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2015 Freescale Semiconductor, Inc.

Document Number KSDK130RN
Revision 0, 09/2015

