

Kinetis SDK Release Notes

1 Overview

These are the release notes for the Freescale Kinetis SDK (KSDK) 1.0.0-beta. The core of the Kinetis SDK is a set of peripheral drivers architected in two layers: Hardware Abstraction Layer (HAL) and the Peripheral Driver layer.

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

Kinetis SDK includes a set of example applications demonstrating the use of drivers and other integrated software modules such as an RTOS through an RTOS abstraction layer. Kinetis SDK also integrates middle-ware such as the Freescale USB stack to provide a true easy-to-use Software Development Kit for Kinetis product families.

Contents

1	Overview	1
2	Development Tools	2
3	System Requirements.....	2
4	Target Requirements	2
5	Release Contents	2
6	Release Overview	3
6.1	Kinetis platform support	3
6.2	Board configuration.....	4
6.3	Example applications	4
6.4	Other integrated software solutions	4
7	Known Issues.....	5
7.1	IAR Debugger Setting.....	5
7.2	Maximum file path length in Windows® 7	5
7.3	No spaces in the KSDK installation	5
8	Revision History	5

2 Development Tools

Freescale Kinetis SDK 1.0.0-beta was compiled and tested with these development tools:

- IAR Embedded Workbench for ARM version 6.70.3
- Makefiles support with GCC revision 4.7.4 from ARM Embedded

3 System Requirements

System requirements are based on the requirements for development tools. There are no special host requirements for hosting the Freescale Kinetis SDK distribution itself.

Minimum PC configuration is determined by the development tools.

Recommended PC configuration is 2 GHz processor, 2 GB RAM and 2 GB free disk space.

4 Target Requirements

Freescale Kinetis SDK supports these evaluation boards:

- TWR-K64F120M
- FRDM-K64F

There are no special requirements for the hardware other than what each board requires to operate.

5 Release Contents

This table describes the release contents.

Table 1. Release Contents

Deliverable	Location
Example applications	<install_dir>/apps/...
Specific content for the evaluation boards	<install_dir>/boards/...
Documentation	<install_dir>/doc/...
Pre-built libraries and projects to build libraries	<install_dir>/lib/...
Driver library, startup code and utilities	<install_dir>/platform/...
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source, and IP extension header files	<install_dir>/platform/CMSIS/...
Peripheral Drivers	<install_dir>/platform/drivers/...
Hardware Abstraction Layer	<install_dir>/platform/hal/...
CMSIS Compliance Startup Code	<install_dir>/platform/startup/...
Utilities such as debug console and Bare Metal OS Abstraction	<install_dir>/platform/utilities/...

Deliverable	Location
Linker control files for each supported toolchain	<install_dir>/platform/linker/...
USB stack	<install_dir>/usb/...
RTOS Kernel Code, RTOS abstraction implementations, and RTOS kernel folders. RTOS kernel code requires a separate download.	<install_dir>/rtos/...

6 Release Overview

The Freescale Kinetis SDK is intended for Freescale microcontrollers with ARM Cortex-M architectures.

The core of the Freescale Kinetis SDK is the peripheral driver library. It does not require an OS, but can be adapted to use various OSES through the OS abstraction layer. It allows easy integration of other software solutions such as OS kernels and the Freescale USB stack.

6.1 Kinetis platform support

The platform directory contains the startup code, driver libraries for peripherals, utilities such as software timer and the OS abstraction implementation for bare metal cases.

6.1.1 Startup code

Freescale Kinetis SDK includes a set of simple CMSIS-compliant startup code which efficiently deliver the code execution to the `main()` function. An application can either include the startup code directly in the workspace or include a prebuilt startup code library for a cleaner project space.

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). OSA defines a common set of services that abstract most of the OS kernel functionality. 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 Freescale Kinetis SDK implements the OS Abstraction layer for Freescale MQX™ RTOS, FreeRTOS, μC/OS-II, μC/OS-III, and for OS-less “bare metal” usage. The bare metal OS abstraction implementation is selected as the default option.

6.1.3 Driver library

Freescale Kinetis SDK provides a set of drivers for the peripherals found on Kinetis product families. The drivers are designed and implemented around the peripheral hardware blocks rather than for a specific Kinetis SoC, and work with or without an OS through the OS Abstraction layer. The drivers are architected into two layers: the Hardware Abstraction Layer (HAL) and the Peripheral Driver layer (PD).

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

The PD is built on top of HAL to provide a set of easy-to-use interfaces that handle high-level data and stateful transactions. PD is designed for the most common use cases identified for the underlying hardware block. The drivers are written in C language and are reasonably efficient in terms of memory and performance. The drivers are also designed to be initialized at runtime based on the driver configuration so that the drivers can be easily ported from product to product and can be used in ROM with minimum effort when necessary. In most cases, the drivers can be used as they are. However, if the PD does not address the target use cases, it can either be modified/enhanced or completely rewritten to meet the target functionality and other requirements. The existing peripheral drivers can be used as references to build the custom drivers based on the HAL.

Detailed implementation of IP functionality, for both HAL and peripheral drivers, 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 will be implemented in the future releases.

6.2 Board configuration

The board directory in the Freescale Kinetis SDK is mainly used for the board-specific configuration and pin muxing. The board directory also contains software components specific to the boards such as Ethernet PHY, Accelerometer, and SPI Flash implementations.

6.3 Example applications

The example applications demonstrate the usage of the driver libraries and other integrated software solutions on supported evaluation boards. For details, see the *Kinetis SDK Demo Applications User Guide*.

6.4 Other integrated software solutions

The Freescale Kinetis SDK is designed for easy integration with other software solutions such as OS, kernel, USB stack, TCP/IP stack, file systems, etc. In this release, the USB stack is integrated with the Freescale Kinetis SDK.

6.4.1 USB stack

A Freescale USB stack is integrated with the KSDK and was tested both with and without an OS through the OS abstraction layer. For details, see the USB stack documentation.

6.4.2 RTOS

The KSDK is pre-integrated with Freescale MQX RTOS, FreeRTOS, μ C/OS-II, and μ C/OS-III. OS abstraction layers are implemented for the RTOSes. The KSDK release package, however, does not contain any RTOS kernel code. The specific kernel code must be downloaded separately from the same location where you downloaded the KSDK and placed in a designated folder in the KSDK. For details, see readme.txt files in the RTOS folders.

7 Known Issues

7.1 IAR Debugger Setting

KSDK 1.0.0-beta is developed and tested with P&E Micro USB OpenSDA as the debugger interface. All IAR projects in KSDK 1.0.0-beta use the P&E Micro USB OpenSDA as the default debugger interface. Therefore, the FRDM-K64F IAR projects will not be able to connect to the boards if the factory preprogrammed debugger interface is still installed on the FRDM-K64F. To fix this issue, reprogram the FRDM-K64F OpenSDA debugger firmware back to the P&E Micro. See the *Kinetis SDK K64 User's Guide* (document KSDKK64UG) for more details.

Limited testing was performed with the CMSIS-DAP debugger interface and an issue was found. The issue occurs when running a demo application that exits the main function, which causes the CMSIS-DAP to stop functioning. The only way to make the CMSIS-DAP functional is to reprogram the CMSIS-DAP firmware. If you want to work around this issue, ensure that the application never returns or exits the main function. This issue is currently under investigation.

7.2 Maximum file path length in Windows® 7

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

7.3 No spaces in the KSDK installation

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

8 Revision History

Revision/Date	Description
1.0.0 Beta 2014	Initial release

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

www.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 and ARM Cortex-M are the registered trademarks of ARM Limited.

© 2014 Freescale Semiconductor, Inc. All rights reserved.

