Freescale Semiconductor, Inc.
Release Notes

Document Number: KSDK110MK21DA5MKW24D5RN
Rev. 0, 02/2015

Kinetis SDK v.1.1.0 Release Notes for the MK21DA5 and MKW24D5 Devices

1 Overview

These are the release notes for the Freescale Kinetis Software Development Kit (KSDK) 1.1.0 for the MK21DA5 and MKW24D5 devices. 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

Contents

easy-to-use Software Development Kit for Kinetis microcontroller (MCU) product families.

2 Development Tools

The Kinetis SDK 1.1.0 for MK21DA5 and MKW24D5 devices was compiled and tested with these development tools:

- Kinetis Design Studio IDE v2.0
- IAR Embedded Workbench for ARM® version 7.20.2
- ARM® Keil® development kit 5.11
- Make files support with GCC revision 4.8.3 from ARM Embedded
- Atollic® TrueSTUDIO® 5.2.0

This table provides a list of default debugger configurations for both the Freescale Tower System modules (TWR) and USB development systems. See Section 3, Supported Development Systems, for a list of Freescale development systems supported by this release of the Kinetis SDK.

Table 1. List of Default Debugger Configurations

	Development System	
IDE	**TWR-K21D50M, TWR-KW24D512**	**USB-KW24D512**
IAR Embedded Workbench for ARM version 7.20.2	P&E Micro	J-Link
ARM Keil 5.11	P&E Micro	J-Link
Kinetis Design Studio IDE v2.0	P&E Micro	J-Link
Makefiles support with GCC revision 4.8.3 from ARM Embedded	J-Link	J-Link
Atollic TrueSTUDIO 5.2.0	J-Link	J-Link

3 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
TWR-K21D50M	MK11DN512AVLK5 MK11DN512AVMC5 MK11DX128AVLK5 MK11DX128AVMC5 MK11DX256AVLK5 MK11DX256AVMC5 MK21DN512AVLK5 **MK21DN512AVMC5** MK21DX128AVLK5 MK21DX128AVMC5 MK21DX256AVLK5 MK21DX256AVMC5
TWR-KW24D512 **USB-KW24D512**	MKW21D256VHA5 MKW21D512VHA5 MKW22D512VHA5 **MKW24D512VHA5**

4 Release Contents

This table describes the release contents.

Table 3. Release Contents

Deliverable	Location
Specific content for the evaluation boards	<install_dir>/boards/...
Demo applications	<install_dir>/demos/...
Documentation	<install_dir>/doc/...
File System	<install_dir>/filesystem/...
Projects to build libraries	<install_dir>/lib/...
Common Make files	<install_dir>/mk/...
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/…
Linker control files for each supported toolchain	<install_dir>/platform/linker/…
OS Abstraction for Bare Metal and RTOS	<install_dir>/platform/osa/…
CMSIS-compliant Startup Code	<install_dir>/platform/startup/…
System Services such as clock manager, interrupt manager, unified hardware timer, and low power manager	<install_dir>/platform/system/…
Utilities such as debug console	<install_dir>/platform/utilities/…
RTOS Kernel Code, RTOS abstraction implementations, and RTOS kernel folders	<install_dir>/rtos/...
A Processor Expert service pack and MQX RTOS task-aware plugins for tool chains.	<install_dir>tools
TCP/IP stacks	<install_dir>/tcpip/...
USB stack and demos	<install_dir>/usb/...

5 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
- Board configuration support
- Demo applications
- The FatFs FAT File System
- USB Host and Device stacks
- RTOS support components
- Documentation (Kinetis SDK reference manual and various user's guides)

5.1 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.

5.1.1 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 space or include a prebuilt startup code library for a cleaner project space.

5.1.2 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.

5.1.3 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.

5.1.4 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*.

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.

5.1.5 Header files

The Kinetis SDKCMSIS directory contains CMSIS-compliant 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 for each peripheral instantiated on the Kinetis MCU. Along with the SoC header files and peripheral extension header files, the Kinetis SDK also includes common CMSIS header files for the ARM Cortex-M core and DSP library from the ARM CMSIS version 4.0 release.

5.1.6 Linker files

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

5.1.7 Utilities

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

5.2 Board configuration

The board directory in the 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.

5.3 Demo applications

The example applications demonstrate the usage of the driver libraries and other integrated software solutions on supported development systems. For details, see the *Kinetis SDK Demo Applications User's Guide* (document KSDK11DEMOUG).

5.4 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.

5.4.1 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*.

5.4.2 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.

5.4.3 RTOS

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

6 Known Issues

6.1 Maximum file path length in Windows® 7

Windows 7 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.

6.2 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.

6.3 USB HUB Power Supply

The external power supply of the USB HUB must be provided before it can be used.

6.4 TWR-K21D50M USB OTG hardware configuration

See the TWR-K21D50M Rev. B Schematics document:

- If using the K21 micro USB connector, leave the 33 ohm resistors on nets K21_MICRO_USB_DP and K21_MICRO_USB_DN and do not populate the resistors on USB0_DP and USB0-_DN.
- If using the Tower USB, install the 33 ohm resistors on nets USB0_DP and USB0_DN and remove the resistors on nets K21_MICRO_USB_DP and K21_MICRO_USB_DN.

6.5 TWR-KW24D512 and USB-KW24D512 LEDs

All TWR-KW24D512 and USB-KW24D512 LEDs are blue. As a result, some demos such as adc_low_power, i2c_comm, do not display LED color as expected.

6.6 USB-KW24D512 ADC hardware configuration

Currently, on USB-KW24D512 SCH-28057 REV A3 (and older), the MCU voltage reference for ADC components SH1 and SH2, are not connected. To enable, zero ohm resistors need to be installed in those locations.

6.7 USB-KW24D512 CDC Serial port

The USB-KW24D512 does not have a default CDC Serial port. If the application loaded to the USB-KW24D512 board supports the USB CDC Virtual Com, it is installed instead.

The driver for this device is located in the *usb/example/device/cdc/virtual_com/inf*

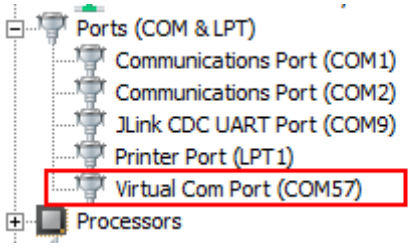

Figure 1 USB serial port

Because USB CDC Virtual Com on USB-KW24D512 is a part of the application loaded to SoC, before flashing a new application to the board, the USB CDC Virtual Com must be disconnected from the terminal first.

To use the USB CDC Virtual Com on USB-KW24D512 in the debugger, initialize it successfully and connect the Virtual Com to the terminal.

6.8 J-Link Lite issue on TWR-K21D50M

On TWR-K21D50M, the board can't be powered up if the J-Link Lite is plugged in first.

6.9 Bubbling noise issue in the dev_audio_speaker USB example

Because of poor clock accuracy of the CSTCE8M00G55-R0 on the TWR-K21D50M, a bubbling noise occurs when the USB audio example is running.

6.10 Installer Issue

The Linux® OS installer was tested only on a host with Ubuntu 12.04 and 14.04 LTS. When uninstalling the Kinetis SDK, the system variable KSDK_PATH remains set in the Windows Registry until the next PC reboot. If you attempt to install the Kinetis SDK before rebooting the PC, the installer thinks that the previous instance is still valid and may not set the KSDK_PATH variable correctly. Reboot the PC after uninstalling the Kinetis SDK to avoid this issue.

7 Revision History

This table summarizes revisions to this document.

Revision History		
Revision number	**Date**	**Substantial changes**
0	02/2015	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.

Document Number: KSDK110MK21DA5MKW24D5RN
Rev. 0
02/2015