**Freescale Semiconductor**
Release Notes

Document Number: KSDK110KV4XRN
Rev. 0, 2/2015

# Kinetis SDK Release Notes for TWR-KV46F150M Freescale Tower System Development Platform

## 1 Overview

These are the release notes for the standalone release supporting the KV4x MCU devices, which is based on Kinetis SDK (KSDK) 1.1.0 release. The core of the Kinetis SDK is a set of peripheral drivers architected in two layers: the Hardware Abstraction Layer (HAL) and the Peripheral Driver layer.

The HAL abstracts the hardware 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.

The Kinetis SDK includes a set of example applications demonstrating the use of drivers and other integrated software.

Contents

# 2 Development Tools

The software modules were compiled and tested with these development tools:

- IAR embedded Workbench version 7.20.2 or later
- ARM® GCC 4.8.3 2014q1 or later
- Keil MDK 5.12.0 or later
- Atollic TrueSTUDIO 5.2.1 or later
- Kinetis Design Studio (KDS) IDE v2.0

**Table 1 List of Default Debugger Configurations**

| Development System | |
|---|---|
| **IDE** | **Tower** |
| IAR Embedded Workbench for ARM version 7.20.2 | P&E Micro |
| MDK-ARM Microcontroller Development Kit (Keil) 5.11 | P&E Micro |
| Atollic TrueSTUDIO for ARM version 5.2.1 | P&E Micro |
| ARM GCC | J-Link |
| Kinetis Design Studio (KDS) IDE | P&E Micro |

# 3 Supported Development Systems

Freescale Kinetis SDK supports these evaluation platforms. Boards and devices in boldface were tested in Kinetis SDK 1.1.0 release.

| Boards | MCU |
|---|---|
| **TWR-KV46F150M** | **MKV46F256VLL15**, MKV46F128VLL15<br>MKV45F256VLL15, MKV45F128VLL15<br>MKV44F128VLL15, MKV43F128VLL15<br>MKV40F256VLL15, MKV40F128VLL15 |

## 3.1 KV4x

KV4x software development is supported through the TWR-KV46F150M development board. See the *Getting Started with Kinetis SDK (KSDK)* (Document KSDKGSUG) for hardware and software environment setup for the Kinetis SDK (KSDK).

Software developed for KV46F256 is compatible with other KV4x devices. Take into account that memory, peripheral and header files are targeted for specific devices and some peripherals may not be available on all variants of Kv4x devices.

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

# 4 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/... |
| Prebuilt 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/… |
| Linker control files for each supported toolchain | <install_dir>/platform/linker/... |

# 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
- RTOS support components
- Documentation (Kinetis SDK reference manual and various user's guides).

## 5.1 Kinetis platform support

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

### 5.1.1 Startup code

The Kinetis SDK includes a set of simple CMSIS-compliant startup code which efficiently delivers 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.

### 5.1.2 Operating system abstraction

The drivers are designed to work with or without an operating system through the OSA. The 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 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.

### 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 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 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 future releases.

## 5.2 Board configuration

The board directory in the Kinetis SDK is mainly used for the board-specific configuration and pin muxing.

## 5.3 Demo 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 v1.1 Demo Applications User's Guide* (Document KSDK11KV4XDEMOUG).

## 5.4 Other integrated software solutions

The Kinetis SDK is designed for easy integration with other software solutions such as OS kernels.

# 6 Known Issues

## 6.1 Maximum file path length in Windows® OS 7

Windows® OS 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 OS. 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 Installer Issue

Note that the Linux® operating system installer was tested only on a host with Ubuntu 14.04.

When uninstalling the Kinetis SDK, the system variable KSDK_PATH will remain set in the Windows Registry until the next PC reboot. If you attempt to install the Kinetis SDK before rebooting the PC, the installer will think 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 the revisions made to this document.

| Revision History | | |
|---|---|---|
| **Revision number** | **Date** | **Substantive changes** |
| 0 | 2/2015 | Initial release |

**How to Reach Us:**

**Home Page:**
www.freescale.com

**Web Support:**
www.freescale.com/support



Document Number: KSDK110KV4XRN
Rev. 0
2/2015