Freescale Semiconductor
Release Notes

Document Number: KBTLDR120RN
Rev. 0, 07/2015

Kinetis Bootloader v1.2.0 Release Notes

1 Overview

These are the release notes for the Kinetis bootloader v1.2.0. This release of the bootloader supports the Freescale platforms identified in the Target section of this document. For more information and getting started instructions, see the files mentioned in the Getting Started section of this document.

The Kinetis bootloader is an application that you program into the internal flash memory of a Kinetis device. The bootloader is designed to detect communication traffic on one of the supported peripherals (USB-HID, UART, SPI, I2C, and CAN), download a user application, and write the application to internal flash. The bootloader stays resident on flash along with your application.

This release includes the PC-hosted Kinetis Updater application. This application allows to you choose a device application image and send it to the bootloader over USB-HID or UART.

Contents

2 Development tools

The Kinetis bootloader 1.2.0 was compiled and tested with these development tools.

Firmware projects:

- Kinetis Design Studio (KDS) IDE v3.0.0
- IAR Embedded Workbench for ARM® v7.40.2

NOTE

> The IAR tool binary path must be added to the system environment path. For example, C:\Program Files (x86)\IAR Systems\Embedded Workbench 7.4\arm\bin.

Host projects:

- Microsoft® Visual Studio® Professional 2013 for Windows® OS Desktop
- Microsoft® NET Framework 4.5 (included in Windows® OS 8)
- Microsoft Visual Studio® C++ Redistributable for Visual Studio 2013 (vcredist_x86.exe)
- Python v2.7 (www.python.org)

NOTE

> The Python path must be added to the system environment path. For example, C:\Python27.

- Apple Xcode® v6.3.1 (for the blhost tool)
- GNU Compiler (GCC) v4.8.1 (for the blhost tool)

3 System requirements

System requirements are based on the requirements for the development tools and the Kinetis Updater application.

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

4 Target requirements

This release of the Kinetis bootloader supports the following platforms:

- FRDM-K22F Freescale Freedom Development platform

- FRDM-K64F Freescale Freedom Development platform
- FRDM-KL25Z Freescale Freedom Development platform
- TWR-K22F120M Freescale Tower System Module platform
- TWR-K64F120M Freescale Tower System Module platform
- TWR-KV46F150M Freescale Tower System Module platform
- TWR-KV31F120M Freescale Tower System Module platform
- TWR-K24F120M Freescale Tower System Module platform
- TWR-K65F180M Freescale Tower System Module platform
- TWR-KV11Z75M Freescale Tower System Module platform

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

5 Release contents

This table describes the release contents.

Table 1. Release Contents

Deliverable	Location
Host updater and demo applications	<install_dir>/apps/...
Host binaries and utilities	<install_dir>/bin/...
Documentation	<install_dir>/doc/...
Bootloader and host tools source code	<install_dir>/src/...
Tool chain build projects	<install_dir>/targets/...
Host validation tools build projects	<install_dir>/validation/…

6 Getting started

See the *Demo Applications User’s Guide for the Freescale Platforms* (document KBTLDRDEMOUG) for a description of how to use the Kinetis bootloader to load a user application on the Kinetis MCUs.

See the *Getting Started with Kinetis ROM Bootloader* (document KBTLDRUG) for Kinetis ROM-specific information.

See the *Getting Started with Kinetis Flashloader* (document KFLLDRUG) for Kinetis Flashloader-specific information.

For porting information, see Chapter 10, “Kinetis bootloader porting” in the *Kinetis Bootloader v1.2.0 Reference Manual* (document KBTLDR120RM).

For customization, see Chapter 11, “Create a custom flash-resident bootloader” in the *Kinetis Bootloader v1.2.0 Reference Manual* (document KBTLDR120RM).

7 Features

The bootloader release contains source code and tool chain projects for building flash-resident bootloaders, and flashloaders for the supported platforms (see Section 4, “Target Requirements”). A flash-resident bootloader stays resident in flash along with the user application. The flash-resident bootloader can be used to download and program an initial application image into a blank area on the flash, and to later update the application. In contrast, a flashloader gets replaced in flash by the user application and thus is a one-time programming aid.

The Kinetis bootloader supports the following communication interfaces for downloading an application. Not all interfaces are supported on all platforms. See the individual platform reference manual for supported interfaces.

- USB-HID
- UART
- I2C
- SPI
- CAN

Usually, USB-HID and UART connections are made directly to a PC, whereas I2C and SPI require additional hardware. The bootloader, running on the target platform, acts as a communication slave. The bootloader can automatically detect which peripheral is being used to download the application and, in the case of UART, automatically detect the baud rate.

The application image is downloaded to the target through a series of command and data packets sent from a host PC or embedded host platform.

8 Host tools

The bootloader release contains source code and build projects for the following PC-based host tools:

- blhost - command line debug tool to send individual commands to the bootloader
- Kinetis Updater - GUI application to download and flash an application image

For more information, see the *Kinetis blhost User’s Guide* (document KBLHOSTUG) and *Kinetis Updater User’s Guide* (document KUPDTRUG).

9 New features

The following new features were introduced in this release:

- Addition of new supported platforms as indicated in Section 4, “Target Requirements”.
- Addition of BusPal example to demonstrate I2C and SPI communication.
- Support for the latest version of IAR tool chain.
- Support for KDS tool chain.

- Addition of Flash Driver API example.
- Improved USB stack with high-speed support.
- Support for the blhost application on Windows® OS XP, Linux® OS, and Mac® OS X.
- Support of CAN peripheral.

10 Fixed issues

The following issues were fixed since the previous release:

- The blhost.exe application now issues a warning message if terminated with Ctrl + C.
- The BusPal example has been added to demonstrate I2C and SPI communication.
- The IAR debug device name has been fixed in the K22F512 projects.

11 Known issues

The following are known issues with this release:

- Due to limited available RAM space, the MKV11Z7 target build for the KDS tool chain does not support the flashloader or flashloader _loader projects.
- Compiler warning messages for the KDS tool chain builds have been disabled. The warnings will be fixed in a future release.
- The Mac OS build of the blhost tool has not been validated with the UART interface. Use the USB interface.

12 Tool notes

- When changing an IAR project to generate additional output (Options->Output Converter->Generate additional output), the output filename extension (Linker->Output->Output filename) must be changed to '.out'.
- The ReceiveSBFile bootloader command has been removed from the *Kinetis Bootloader v1.2.0 Reference Manual* (document KBTLDR120RM). It will be fully supported in an upcoming release.
- The blhost tool accepts any speed setting in the "--buspal" option for the SPI and I2C peripherals. However, the maximum effective speed settings when using the BusPal example are approximately 300 kHz for I2C and 8000 kHz for SPI.
- When using KDS, load projects using File->Import->Existing Projects into Workspace. Do not use File->Import->Projects of Projects.

Each bootloader target supports one or more of the following project types:

1. tower_bootloader – bootloader designed to execute from target flash memory on the Tower platform.
2. freedom_bootloader – bootloader designed to execute from target flash memory on the Freedom platform.
3. flashloader – bootloader designed to execute from target RAM memory on either the Freedom or Tower platform.
4. flashloader_loader – bootstrap loader designed to execute from flash memory on either the Freedom or Tower platform. This loader copies an image of the flashloader into RAM, then executes the flashloader from RAM.

The flashloader_loader project uses the output of the flashloader build to create the flashloader image to load into RAM. For this reason, the flashloader project must be built before building the flashloader_loader project.

13 Platform notes

1. TWR-KV11Z75M: When setting up the board, first power on Elevator board, then plug in the OpenSDA line so that the TWR-KV11Z75M Tower System board can startup successfully.
2. TWR-KV31F120M: When using the Elevator board on KV31, first power on the Elevator board. Then, plug in the OpenSDA line so that the TWR-KV31F120M Tower System board can startup successfully.

14 Revision history

This table summarizes revisions to this document.

Revision history		
Revision number	**Date**	**Substantive changes**
0	07/2015	Kinetis Bootloader 1.2.0 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: KBTLDR120RN
Rev. 0
07/2015