Freescale Semiconductor
User's Guide

Document Number: KBTLDRUG
Rev. 1, 07/2015

Getting Started with the Kinetis ROM Bootloader

Contents

1 Introduction

This document describes how to interface with the Freescale Kinetis ROM-based bootloader to program a user application image into the on-chip flash. The Kinetis bootloader application resides in the Kinetis ROM and enables flash programming without the need for a debugger.

2 Overview

This guide describes the steps required to program a user application image into the Kinetis on-chip flash memory utilizing the standardized Kinetis bootloader command interface. From the factory, the device boots from ROM and executes the Kinetis bootloader which has access to the entire flash array for placement of the user application.

2.1 Kinetis bootloader

The Kinetis bootloader is a standard bootloader for all Kinetis devices. It provides a standard interface to the device using any of the peripherals supported by the bootloader on a given Freescale Kinetis device.

The Kinetis ROM bootloader is a specific implementation of the Kinetis bootloader. For the ROM use case, the Kinetis bootloader command interface is packaged as an executable that is part of the Kinetis ROM. This configuration allows the user application to be placed at the beginning of the on-chip flash where it is automatically launched upon boot from flash. Additionally, the ROM implementation can never be erased and thus can be used to program the Kinetis flash memory at manufacturing time and to update the user application image in the field.

Other Kinetis bootloader implementations include the Kinetis flashloader used as a one-time programming aid for manufacturing and a flash-resident bootloader used for field updates. The Kinetis bootloader is available as source code for custom, flash-resident bootloader implementations. Example applications are provided to demonstrate how to interface with the bootloader.

Many methods are available to execute the Kinetis ROM bootloader. On a blank flash, the Kinetis bootloader executes. After the flash is programmed, the value of the FOPT field at offset 0x40D in the flash memory space determines if the device boots the ROM bootloader or the user application in flash.

See the *Kinetis Bootloader v1.2.0 Reference Manual* (document KBTLDR120RM) for your particular device to determine the correct FOPT value for the user application.

2.2 Host utility

The blhost utility is an example host program used to interface with devices running the Kinetis bootloader. It can list and request execution of all of the commands supported by a given Kinetis device running the bootloader.

NOTE

The blhost application is released as part of Kinetis bootloader release package available on www.freescale.com/KBOOT . The blhost application is available in the `<install_dir>/bin` folder of the release package.

3 The Kinetis ROM bootloader application

The Freescale Kinetis platform must be connected to a host computer to interface with the Kinetis bootloader application. After the platform is connected, use the blhost application to program a user application into the Kinetis flash memory.

The startup process documented in Kinetis ROM bootloader chapter of the device reference manual shows conditions that force the hardware to start the Kinetis ROM bootloader.

3.1 Connecting to the Kinetis platform

The Kinetis ROM bootloader supports UART and USB connections to a computer. See the Kinetis Reference Manual for the specific device to determine which peripherals are supported by the bootloader application and how the signals are routed to the pins of the Kinetis platform. After the Kinetis platform is powered up and there is a physical serial/USB connection between the Kinetis platform and host, the Kinetis device should be ready to receive commands.

For this example, a Kinetis device is connected to a serial-to-USB converter that enumerates on a Windows® operating systems PC as a Serial Port on COMxx.

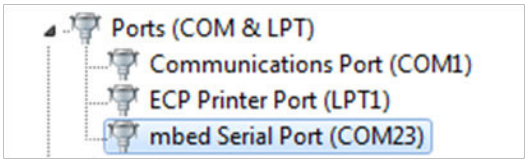

Figure 1. UART connection to Kinetis platform

Figure 2. Alternate UART connection to Kinetis platform

3.2 The host utility application - blhost

This section describes a simple usage of the blhost host utility program to demonstrate communication with the Kinetis bootloader.

- Open a command prompt in the directory containing blhost. For Windows OS, it is `<install_dir>/bin/win`.
- Type *blhost --help* to see the complete usage of the blhost utility.

For this step, it is important to verify that the Kinetis device is properly connected and is running the bootloader firmware application.

- It is assumed that the Kinetis platform is fresh out of reset.
- Note the COM port that the Kinetis platform is connected to as we did earlier. For this example, the device is connected to COMxx.
- Type *blhost -p COMxx -- get-property 1* to get the bootloader version from the Kinetis bootloader .
- An image similar to this screen shot shows that blhost is successfully communicating with the Kinetis platform.

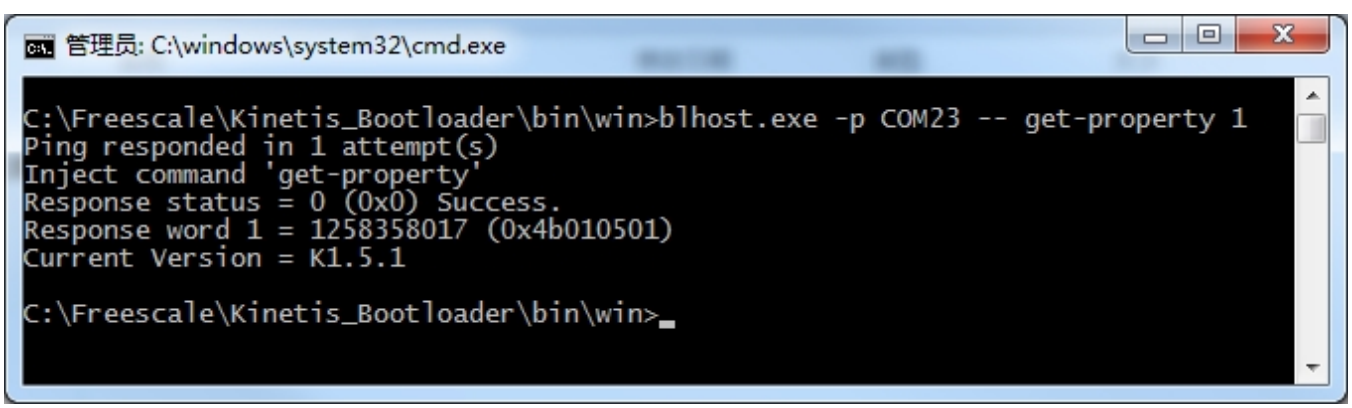

Figure 3. Host communication with Kinetis ROM bootloader

3.3 Flashing the user application

Now that communications have been established between the Kinetis bootloader and the host, issue two commands to program the Kinetis flash memory with a user application.

- *blhost -p COMxx -- flash-erase-all* - Erases the entire flash array.
- *blhost -p COMxx -- write-memory 0 myApp.bin* – Writes the myApp.bin binary image to address 0 of the Kinetis flash memory.
- [Optional] *blhost -p COMxx -- reset* – Resets the Kinetis platform and launches the user application. Note the Kinetis bootloader is no longer running on the device, so further commands issued from the blhost utility fail.
- After issuing the reset command, allow 5 seconds for the user application to start running.
- A screen shot similar to this image shows the successful completion of the above commands.

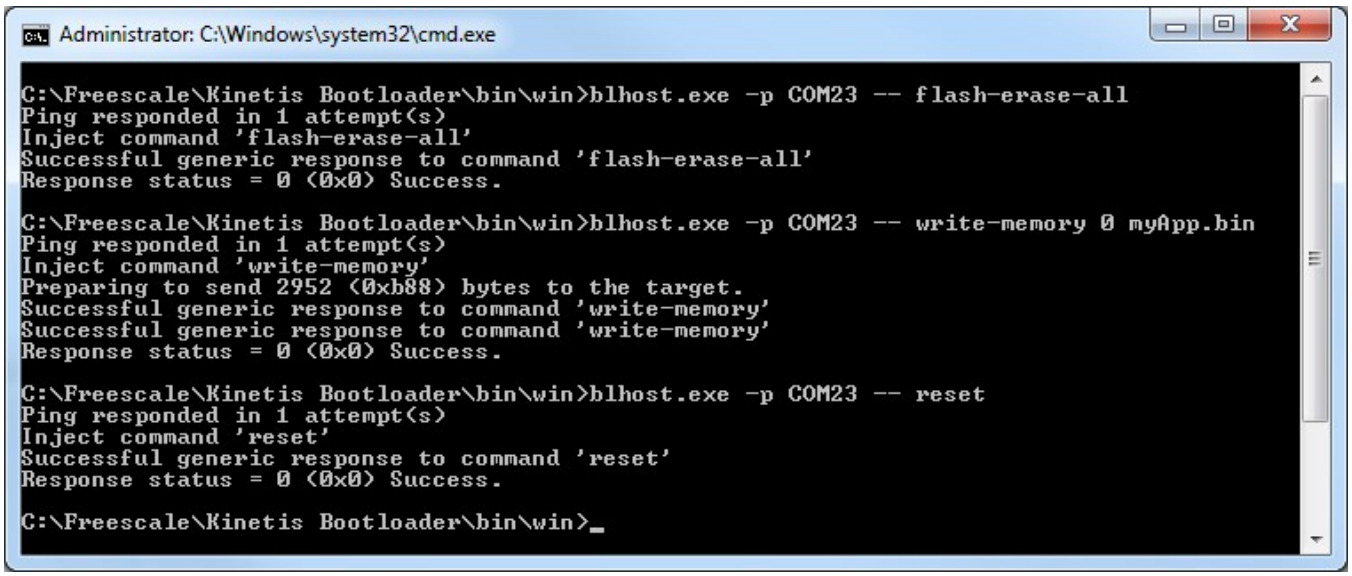

Figure 4. Programming a user application using the Kinetis ROM bootloader

4 User application: vector table offset

Section 3 discusses how to program the Kinetis flash memory with the user application, myApp.bin. When creating the user application, the vector table of the application must be located at the beginning address of the flash memory region.

When booting from flash, the Kinetis device considers offset 0 the initial stack pointer and offset 4 the entry point for the application.

5 User application: Flash Configuration Area

The Flash Configuration Area (0x400-0x40F) should be carefully populated with known values according to the Reference Manual for the specific Kinetis platform. In particular, values for the FSEC (0x40C) and FOPT (0x40D) locations may prevent future writes to the Kinetis flash. Extra attention to ensure the correct values in your application image at these offsets is highly recommended. If your code (other than the vector table) is linked to begin at offset 0x410, then the default erased value (0xFF) of these locations makes the device secure, but mass erase is then enabled.

6 Revision History

This table summarizes revisions made to this document.

Table 1. Revision History

Revision number	Date	Substantive changes
0	07/2014	Initial release
0.1	12/2014	• Updated Figure 3 title to include "Kinetis ROM bootloader" • Updated Figure 4 title to include "Kinetis ROM bootloader"
1	07/2015	• Added content to Section 3.3 • Updates for Kinetis Bootloader 1.2.0

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number KBTLDRUG
Revision 1, 07/2015

freescale™