

Getting Started with the Kinetis ROM Bootloader

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 or external serial 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 Kinetis 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.

Contents

1	Introduction.....	1
2	Overview.....	1
2.1	Kinetis bootloader	1
2.2	Host utilities.....	2
3	The Kinetis ROM bootloader application.....	2
3.1	Connecting to the Kinetis platform.....	2
3.2	The host utility application - blhost.....	3
3.3	Flashing the user application.....	3
3.4	The KinetisFlashTool application	4
4	User application: vector table offset.....	4
5	User application: Flash Configuration Area.....	4
6	Revision History.....	5



The Kinetis ROM bootloader application

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 v2.0.0 Reference Manual* (document KBTLDLDR200RM) for your particular device to determine the correct FOPT value for the user application.

2.2 Host utilities

The blhost utility is a command-line 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. The KinetisFlashTool is a GUI host application that can be used to program an application image.

NOTE

These applications are released as part of Kinetis bootloader release package available on www.nxp.com/KBOOT. They are available in the <install_dir>/bin/Tools 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 or KinetisFlashTool 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 a 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.

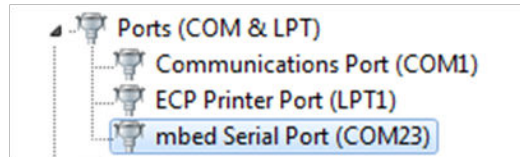


Figure 1. UART connection to Kinetis platform



Figure 2. Alternate UART connection to Kinetis platform

3.2 The host utility application - blhost

This section describes how blhost host utility program is used to communicate with the Kinetis bootloader.

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

For this step, 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. 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.

 A screenshot of a Windows command prompt window titled 'Administrator: C:\windows\system32\cmd.exe'. The command executed is `C:\Freescall\Kinetis Bootloader\bin\Tools\blhost\win>blhost.exe -p COM23 -- get-property 1`. The output shows:


```

C:\Freescall\Kinetis Bootloader\bin\Tools\blhost\win>blhost.exe -p COM23 -- get-property 1
Ping responded in 1 attempt(s)
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258422272 (0x4b020000)
Current Version = K2.0.0
C:\Freescall\Kinetis Bootloader\bin\Tools\blhost\win>
  
```

Figure 3. Host communication with Kinetis ROM bootloader

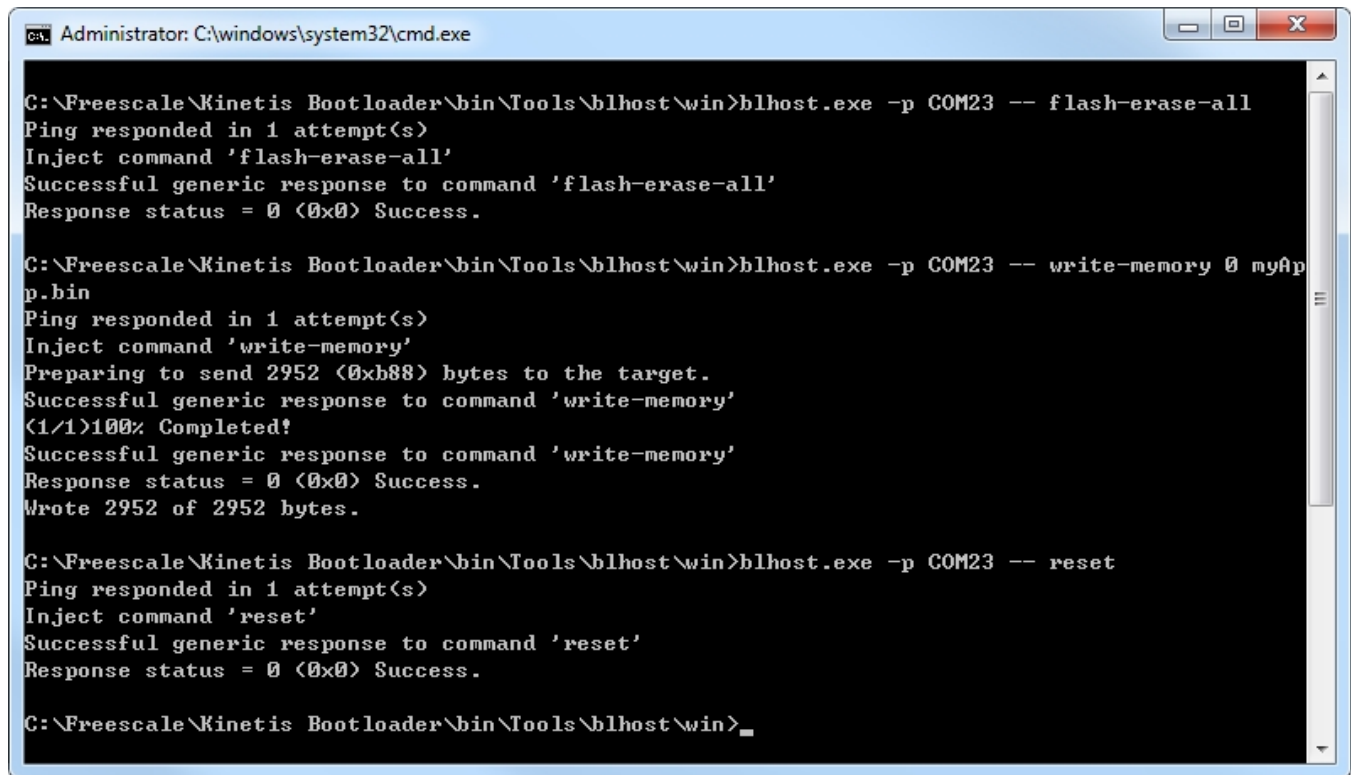
3.3 Flashing the user application

After communication has been established between the Kinetis bootloader and the host, the 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.

User application: vector table offset

- [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.



```
Administrator: C:\windows\system32\cmd.exe
C:\Freescale\Kinetis Bootloader\bin\Tools\blhost\win>blhost.exe -p COM23 -- flash-erase-all
Ping responded in 1 attempt(s)
Inject command 'flash-erase-all'
Successful generic response to command 'flash-erase-all'
Response status = 0 (0x0) Success.

C:\Freescale\Kinetis Bootloader\bin\Tools\blhost\win>blhost.exe -p COM23 -- write-memory 0 myApp.p.bin
Ping responded in 1 attempt(s)
Inject command 'write-memory'
Preparing to send 2952 (0xb88) bytes to the target.
Successful generic response to command 'write-memory'
(1/1)100% Completed!
Successful generic response to command 'write-memory'
Response status = 0 (0x0) Success.
Wrote 2952 of 2952 bytes.

C:\Freescale\Kinetis Bootloader\bin\Tools\blhost\win>blhost.exe -p COM23 -- reset
Ping responded in 1 attempt(s)
Inject command 'reset'
Successful generic response to command 'reset'
Response status = 0 (0x0) Success.

C:\Freescale\Kinetis Bootloader\bin\Tools\blhost\win>
```

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

3.4 The KinetisFlashTool application

The KinetisFlashTool application can also be used to program an application to the device. See the *Kinetis Flash Tool User's Guide* (document KFLASHTOOLUG) for more information.

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, the default erased value (0xFF) of these locations makes the device secure. However, mass erase is 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	<ul style="list-style-type: none"> Updated Figure 3 title to include "Kinetis ROM bootloader" Updated Figure 4 title to include "Kinetis ROM bootloader"
1	07/2015	<ul style="list-style-type: none"> Added content to Section 3.3 Updates for Kinetis Bootloader 1.2.0
2	04/2016	<ul style="list-style-type: none"> Added Section 3.4 Updates for Kinetis Bootloader 2.0

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.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: nxp.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. ARM, ARM Powered Logo, Keil, and Cortex are registered trademarks of ARM limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2016 Freescale Semiconductor, Inc.

