NXP Semiconductors
User's Guide

Document identifier: MBOOTFLASHGS
Rev. 6, 31 December 2020

Getting Started with the Flashloader for Kinetis

1 Introduction

This document describes how to interface with the flashloader to program a user application image into the on-chip flash for Kinetis devices. The flashloader is pre-programmed into the Kinetis device during manufacturing and enables flash programming without a debugger.

2 Overview

This document describes the steps required to program a user application image into Kinetis flash memory, utilizing the standardized MCU bootloader command interface. In the factory, the device boots from flash memory and loads the flashloader into RAM. Running from RAM, the flashloader has access to the entire flash array for placement of the user application. After the user application is programmed into flash memory, the flashloader is no longer available.

2.1 Flashloader

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

The flashloader is a specific implementation of the MCU bootloader. For the flashloader implementation, the MCU bootloader command interface is packaged as an executable. This executable is loaded from flash and executed from RAM. This configuration allows the user to place the application at the beginning of the on-chip flash where it automatically launches when the device boots from flash.

Other MCU bootloader implementations include a ROM-based bootloader and a flash-resident bootloader. The MCU bootloader is available as source code for custom flash-based implementations. The package includes source code and example applications to demonstrate how to interface with the bootloader.

Using the flashloader to program a user application to the beginning of the Kinetis flash makes this implementation of the bootloader a one-time programming aid.

The developers creating a manufacturing flow for their hardware and software implementations may restore flashloader so that the device works as it did in the NXP factory. To accomplish this, use an external debugger to program the `flashloader_loader.bin` file built from flashloader_loader project to the Kinetis on-chip flash. The procedure for programming `flashloader_loader.bin` varies depending on hardware design and available tools.

2.2 Host utilities

The blhost utility is a command-line host program used to interface with devices running the MCU 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

blhost is released on MCUBOOT webpage nxp.com/MCUBOOT and Kinetis Flash Tool is available in the SDK package: `<sdk_package>/middleware/mcu-boot/bin/Tools`.

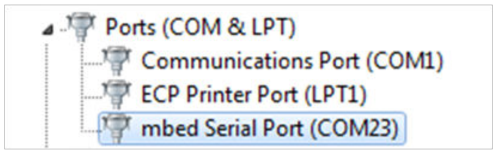
3 Flashloader application

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

3.1 Connecting to the Kinetis platform

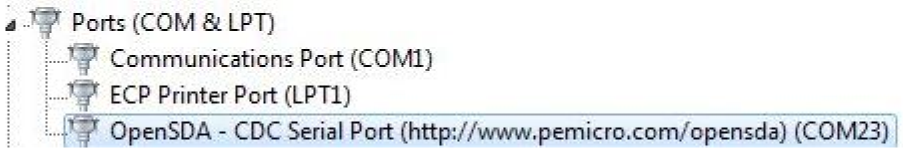
The flashloader supports UART and USB connections to a computer. See the Reference Manual for a specific device to determine the peripherals supported by the flashloader application and the signals route to the pins of the Kinetis platform. After the Kinetis platform is powered up, there is a physical serial/USB connection between the Kinetis platform and host. The Kinetis device is 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 COMx.



Ports (COM & LPT)
Communications Port (COM1)
ECP Printer Port (LPT1)
mbed Serial Port (COM23)

Figure 1. UART connection to MCU platform



Ports (COM & LPT)
Communications Port (COM1)
ECP Printer Port (LPT1)
OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM23)

Figure 2. Alternate UART connection to MCU platform

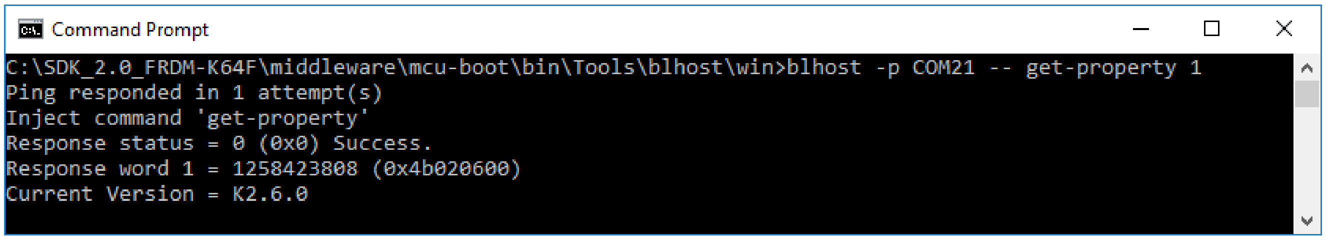
3.2 The host utility application - blhost

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

- Open a command prompt in the directory containing blhost.
- Type *blhost --help* to see the complete usage of the blhost utility.

Verify if the Kinetis device is connected and is running the flashloader firmware application.

- Assumption: The Kinetis platform just came out of reset.
- Check under the COM port in Device Manager that the Kinetis platform is connected. In this example, let us say that the device is connected to COMx.
- Type *blhost -p COMx -- get-property 1* to get the flashloader version from the flashloader.
- Below screenshot shows that blhost is successfully communicating with the Kinetis platform.



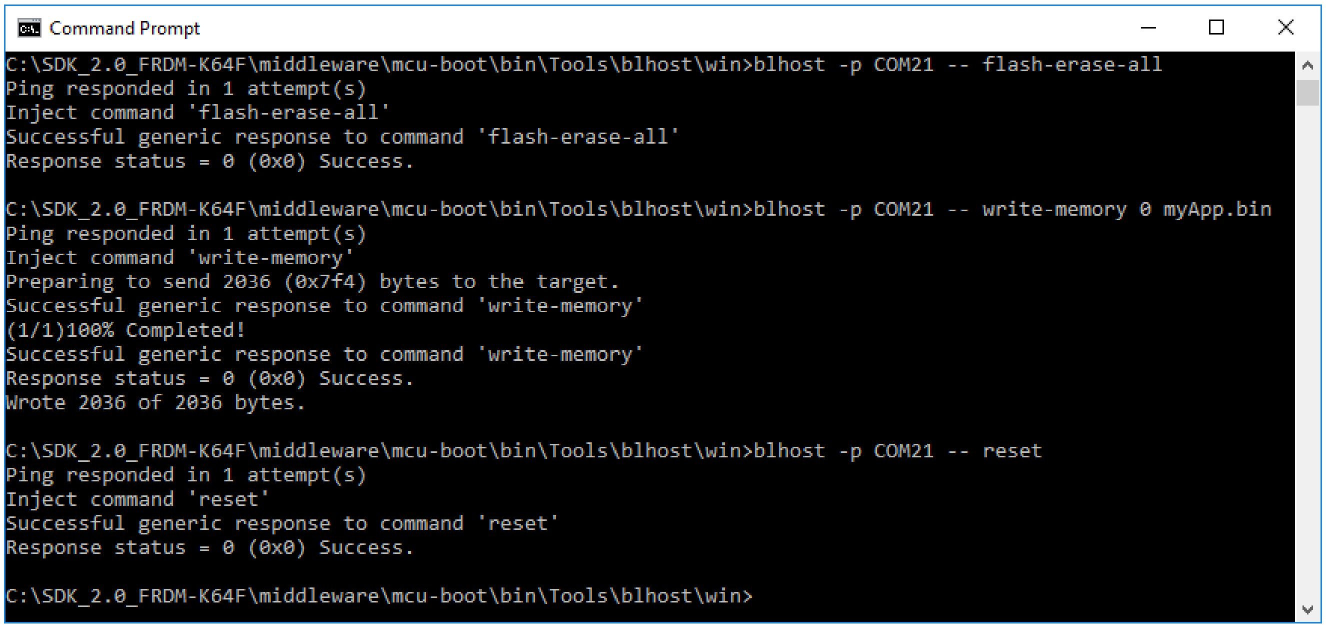
```
C:\SDK_2.0_FRDM-K64F\middleware\mcu-boot\bin\Tools\blhost\win>blhost -p COM21 -- get-property 1
Ping responded in 1 attempt(s)
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258423808 (0x4b020600)
Current Version = K2.6.0
```

Figure 3. Host communication with MCU flashloader

3.3 Flashing the user application

The flashloader issue two commands to program the Kinetis flash memory with a user application once the communication establishes between the flashloader and the host.

- *blhost -p COMx -- flash-erase-all* - Erases the entire flash array.
- *blhost -p COMx -- write-memory 0 myApp.bin* – Writes the myApp.bin binary image to address 0 of the Kinetis flash memory.
- [Optional] *blhost -p COMx -- reset* – Resets the Kinetis platform and launches the user application. Note: the flashloader 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.
- Below screenshot shows the successful completion of the above commands.



```
C:\SDK_2.0_FRDM-K64F\middleware\mcu-boot\bin\Tools\blhost\win>blhost -p COM21 -- 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:\SDK_2.0_FRDM-K64F\middleware\mcu-boot\bin\Tools\blhost\win>blhost -p COM21 -- write-memory 0 myApp.bin
Ping responded in 1 attempt(s)
Inject command 'write-memory'
Preparing to send 2036 (0x7f4) 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 2036 of 2036 bytes.

C:\SDK_2.0_FRDM-K64F\middleware\mcu-boot\bin\Tools\blhost\win>blhost -p COM21 -- reset
Ping responded in 1 attempt(s)
Inject command 'reset'
Successful generic response to command 'reset'
Response status = 0 (0x0) Success.

C:\SDK_2.0_FRDM-K64F\middleware\mcu-boot\bin\Tools\blhost\win>
```

Figure 4. Programming a user application using the MCU flashloader

3.4 The KinetisFlashTool application

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

4 User application: vector table offset

" Chapter 3, MCU flashloader application discusses how to program the Kinetis flash memory with the *myApp.bin* user application. When creating the user application, the vector table of the application must be 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) must be populated with known values as per the specific Kinetis platform reference manual. In particular, values of the FSEC (0x40C) and FOPT (0x40D) locations may prevent future writes to the Kinetis flash. In the user application code, the vector table should begin at offset 0x410. If any other code 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 to this document.

Table 1. Revision history

Revision number	Date	Substantive changes
0	12/2014	Initial release
1	07/2015	Kinetis Bootloader 1.2.0 updates
2	04/2016	Kinetis Bootloader 2.0 updates
3	05/2018	MCU Bootloader v2.5.0 release
4	09/2018	MCU Bootloader v2.6.0 release
5	11/2018	MCU Bootloader v2.7.0 release
6	31/2020	MCU Bootloader v2.7.1 release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 31 December 2020

Document identifier: MBOOTFLASHGS