

Demo Application User's Guide for the FRDM-K22F and TWR- K22F120M Freescale Platforms

Contents

1	Introduction.....	1
2	Overview.....	1
3	Hardware configuration.....	2
4	Kinetis bootloader application.....	4
5	The host utility application.....	7
6	Windows GUI updater application.....	8
7	Appendix A Kinetis flash-resident bootloader operation.....	11
8	Revision History.....	14

1 Introduction

This document describes how to use the Kinetis bootloader to load a user application on a Kinetis K22 series MCU.

2 Overview

This guide describes the steps required to use the Freescale-provided Kinetis bootloader utilities to both load the Kinetis bootloader image and use the bootloader to update the user application section of flash. Upon reset, the bootloader detects the presence of the user application and launches it. The bootloader also provides a means to suppress the application launch and remain in the bootloader command processor in order to refresh the user application. This full-circle environment enables application developers to easily install new applications onto Kinetis devices, and provides manufacturers a way to update Kinetis devices in the field without the need for a debugger.

2.1 Kinetis bootloader

Hardware configuration

The Kinetis bootloader serves as the standard bootloader for all Kinetis devices. It provides a standard interface to the device via all of the available peripherals supported on a given Freescale Kinetis device. The Kinetis bootloader interface comes in several forms, ranging from ROM, serial flashloader, or a customized flash-resident bootloader. All future Kinetis devices will arrive with a ROM containing the Kinetis bootloader, while others will arrive pre-programmed from the factory with a one-time-use serial flashloader. For a customized interface, customers can leverage the Kinetis bootloader source code to create a unique flash-resident bootloader that is both compatible with tools that understand the bootloader interface, and are capable of supporting application-specific features. Freescale provides utilities to demonstrate how to interface with the bootloader.

2.2 Host utility

The blhost.exe utility is a cross-platform 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.

2.3 led_demo user application

The led_demo_FRDM-K22F*.bin and led_demo_TWR-K22F120M*.bin programs are example demo firmware applications used to demonstrate how the Kinetis Bootloader can load and launch user applications.

2.4 Host updater

The KinetisUpdater.exe host application is a Windows® GUI program used to update the user application image on the device running the Kinetis bootloader firmware application.

2.5 Toolchain requirement

Firmware projects:

- IAR Embedded Workbench for ARM version 7.20 (or later)

Host projects:

- Microsoft® Visual Studio® Express 2013 for Windows Desktop
- Microsoft® .NET Framework 4.5 (included in Windows 8)
- Microsoft® Visual C++ Redistributable for Visual Studio 2013 (vc redistrib_x86.exe)
- Python v2.7 (www.python.org)

3 Hardware configuration

This section describes how to set up the hardware in order to run the Kinetis bootloader example.

3.1 FRDM-K22F

The bootloader example application assumes that all platform jumpers are set to their default positions.

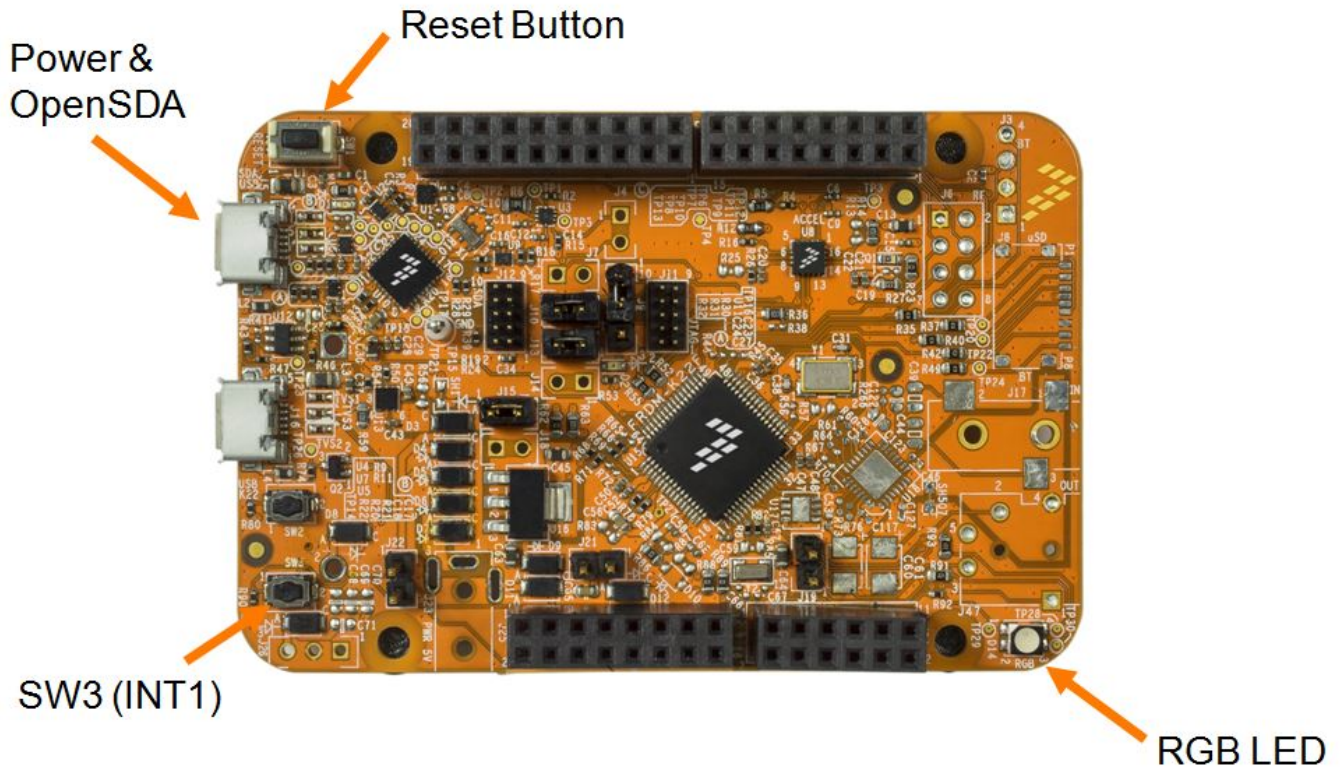


Figure 1. Front side of FRDM-K22F platform

3.2 TWR-K22F120M

The bootloader example application assumes that all module jumpers are set to their default positions.

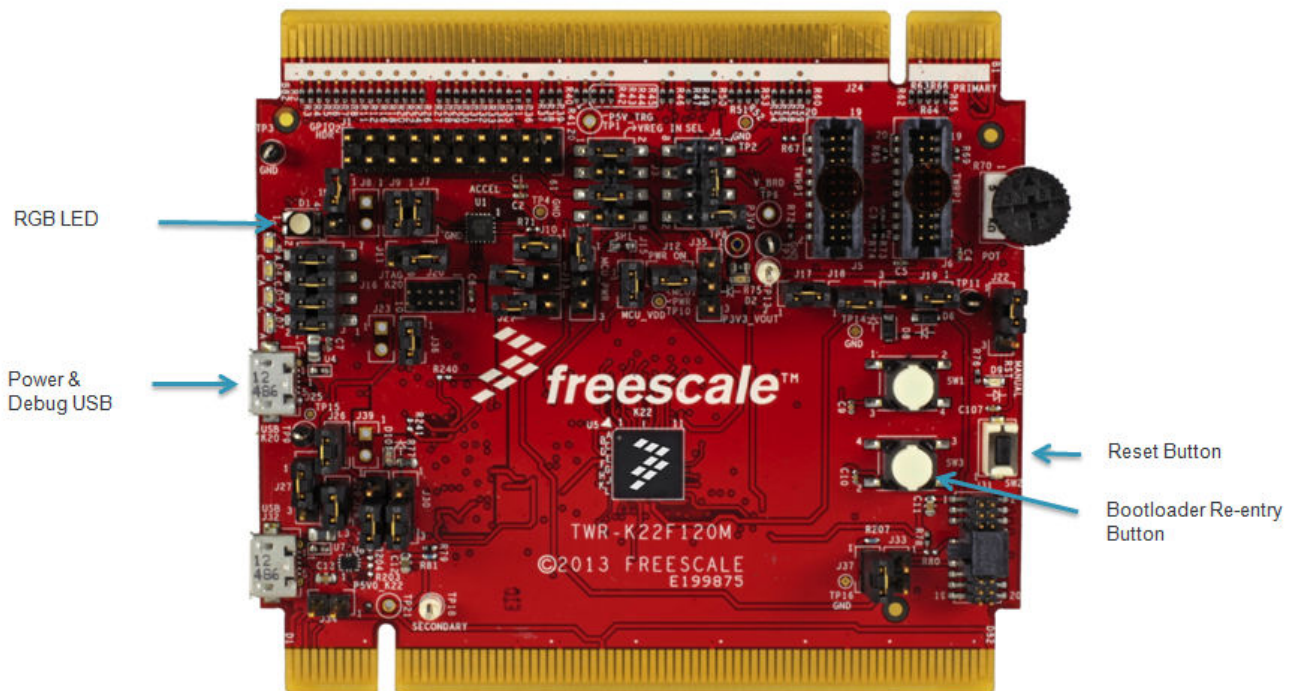


Figure 2. Front side of TWR-K22F120M module

4 Kinetis bootloader application

This section describes how to connect the platform to the computer and download the pre-built Kinetis bootloader application.

4.1 Connect the Freedom platform

The following section assumes the board is set up with the default factory configurations. For more information, visit www.freescale.com/FRDM-K22F.

For Windows PCs, install the mbed serial port driver in order to communicate with the Kinetis device over a serial port.

1. Download and install the latest mbed Windows serial port driver from <http://developer.mbed.org/handbook/Windows-serial-configuration>.
2. Connect the OpenSDA USB connector, J5 for the FRDM-K22F board, to the USB port on a PC.
3. Install the mbed serial port driver.

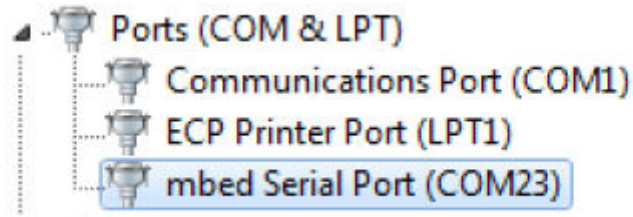


Figure 3. mbed serial port in Windows Device Manager

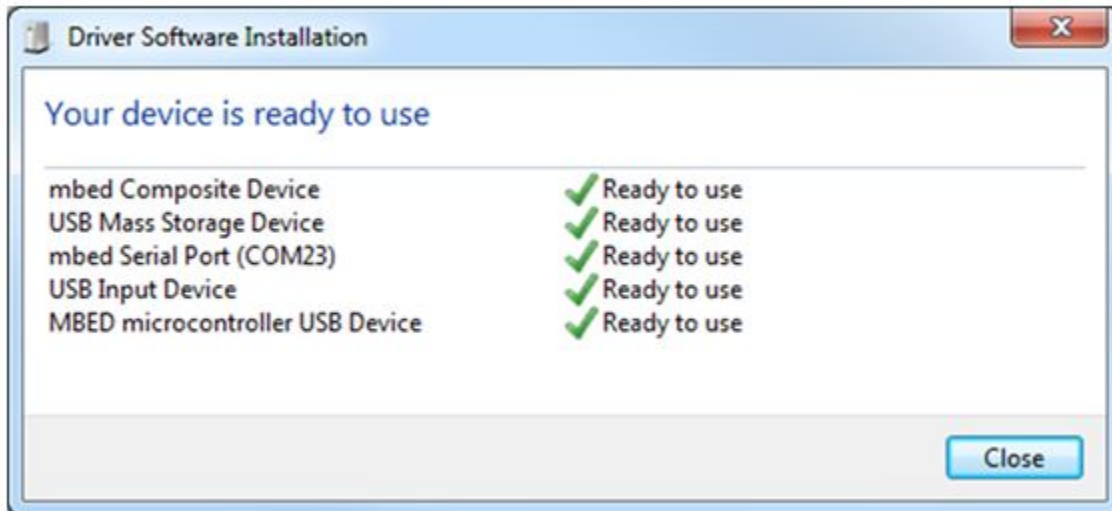


Figure 4. Driver software installation

4.1.1 Install the Kinetis bootloader onto the Freedom Platform

To install the bootloader application, drag or copy and paste the `freedom_bootloader.bin` file from `<install_dir>/targets/MK22F51212/binaries` onto the MBED mass storage device.

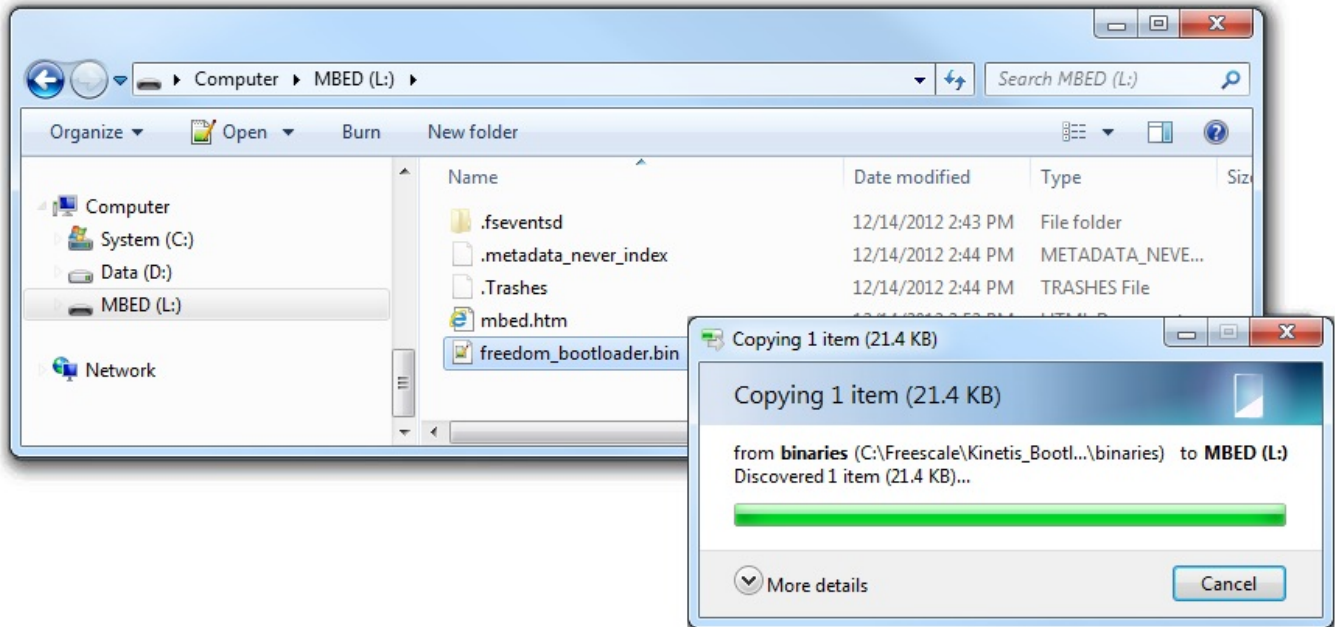


Figure 5. Install the Kinetis bootloader on the FRDM board

4.2 Connect the Tower module

The following section assumes the board is set up with the default factory configurations. For more information, visit www.freescale.com/TWR-K22F120M.

For Windows PCs, install the P&E Micro OpenSDA drivers in order to communicate with the Kinetis device over a serial port.

1. Connect the module to the USB port on a PC using the module's debug USB connector.
2. Download the driver package from the P&E Micro website (www.pemicro.com/opensda/) and run the installer.
3. After the installer finishes, plug in the module and open the Windows Device Manager to show the COM port number assigned to the virtual serial port.



Figure 6. OpenSDA virtual comport in Windows Device Manager

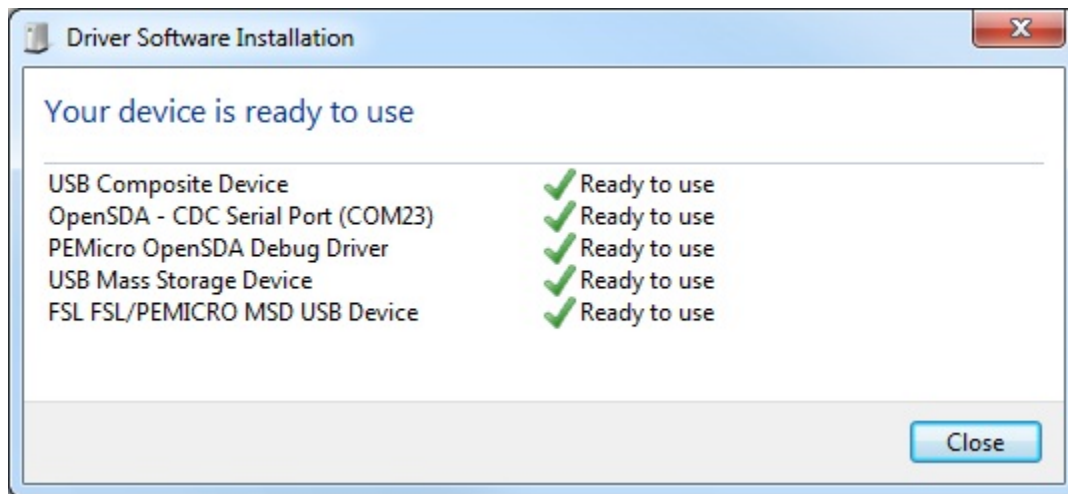


Figure 7. Driver software installation

4.2.1 Install the Kinetis bootloader onto the Tower Module

To install the bootloader application, drag the tower_bootloader.srec file from <install_dir>/targets/MK22F51212/binaries onto the TWR-K22F120 mass storage device.

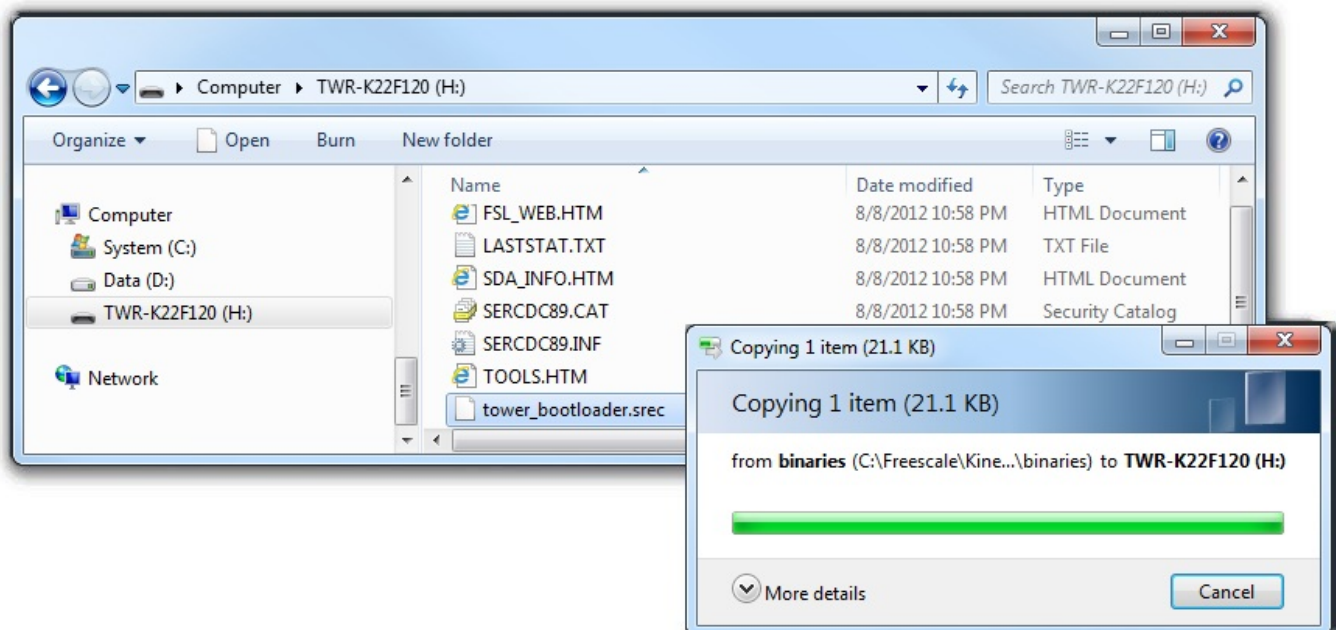


Figure 8. Install the Kinetis bootloader on the TWR board

5 The host utility application

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

Windows GUI updater application

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

For this exercise, verify the Kinetis device is running the bootloader firmware application.

- Press the "Reset" button on the platform.
- Note what the COM port that the platform is connected to. Refer to step 3 of section 4.1. For this guide, the device is connected to COM23.
- Type *blhost -p COM23 -- get-property 1* to get the bootloader version from the Kinetis bootloader.
- Something similar to the screen shot below indicates that blhost.exe is successfully communicating with the Kinetis bootloader on the platform.

```
C:\Freescall\Kinetis_Bootloader\bin\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 = 1258357504 (0x4b010300)
Current Version = K1.3.0
```

Figure 9. Host communication with Kinetis bootloader

6 Windows GUI updater application

This section describes how to use the Windows GUI updater application, KinetisUpdater.exe, to install an example user application onto the platform.

6.1 Installing the user application

The FRDM-K22F platform is used in this example. Similar steps can be used for the TWR-K22F120M module.

1. Click the "Reset" button on the platform.
2. Navigate Windows Explorer to the <install_dir>/bin/win/KinetisUpdater directory.
3. Double-click the KinetisUpdater.exe file to launch the app.

NOTE

If the application fails to launch, check that the .NET Framework 4.5 and the Visual C++ Redistributable 2013 are installed as noted in Toolchain requirements.

4. Start at the orange home screen. Click "Select Device".
 - The blue device configuration page will show.
 - Select the COM23 device from the drop-down box.
 - Click "Home" to return to the home screen.

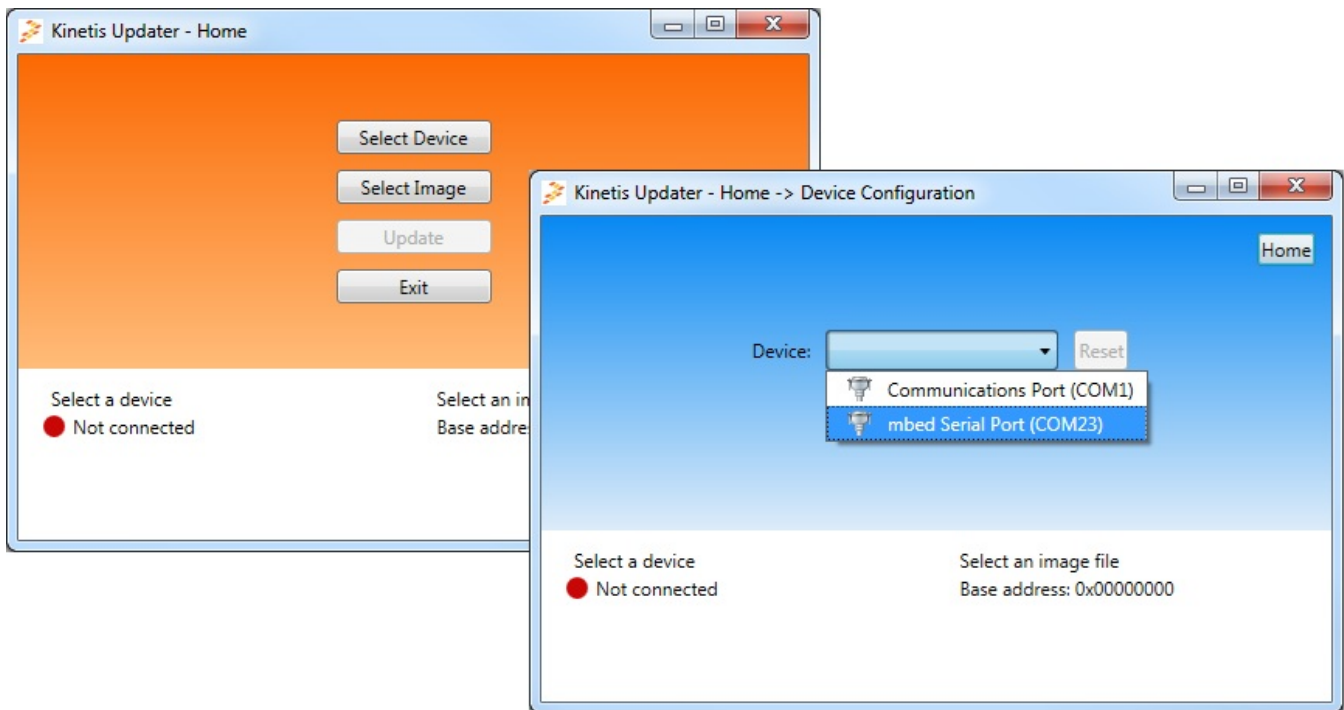


Figure 10. Select the COM port

5. Click "Select Image".

- The blue image configuration page will show.
- Select the led_demo_FRDM-K22F_a000.bin application image from the <install_dir>/apps/led_demo/binaries directory using the "Browse" button.
- Set the base address to 0xA000.
- Click "Home" to return to the home screen.

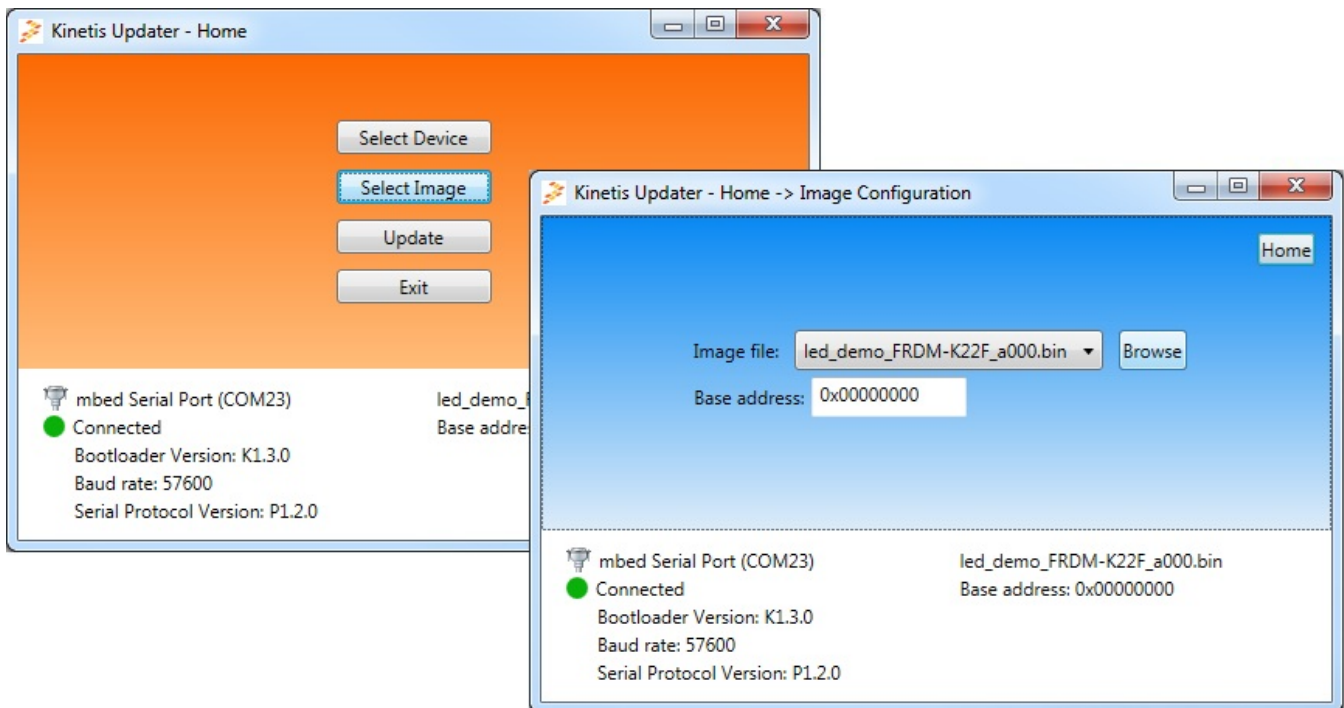


Figure 11. Browse for the user application

Windows GUI updater application

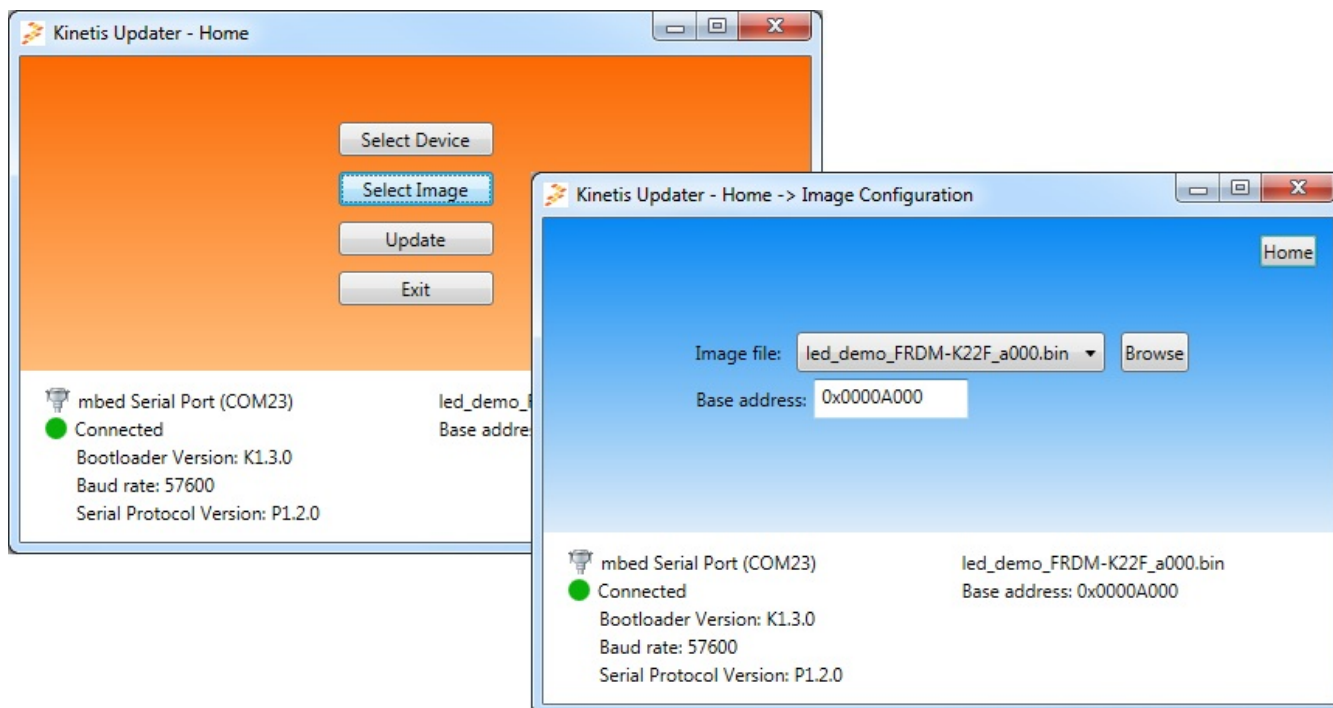


Figure 12. Set base address for application file

7. The "Update" button should now be enabled. Click "Update".
 - The blue update page will now be displayed.
 - Click the "Update" button to write the application image to the device flash.
 - Click "Home" to return to the home screen.

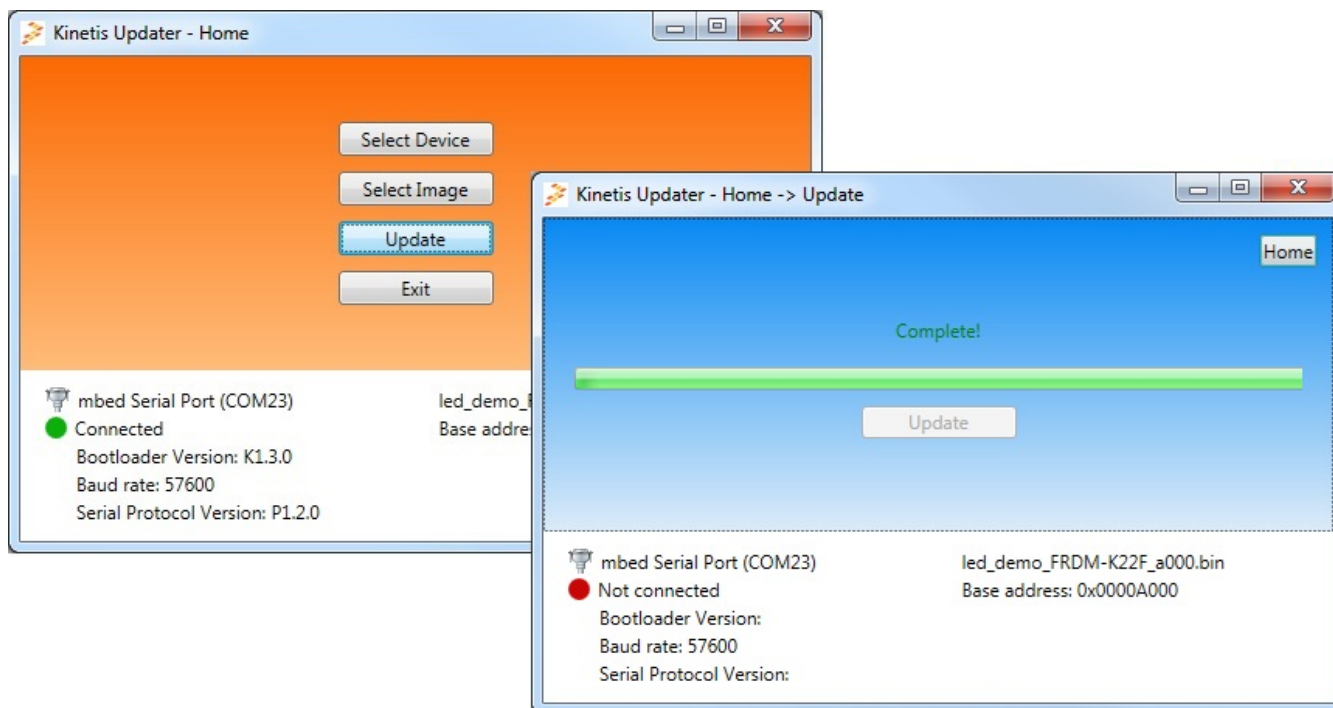


Figure 13. Perform the update

8. Click the "Exit" button.
9. At this point, the led(s) on the target board should be noticeably blinking indicating that the Kinetis Bootloader successfully installed the led_demo user application.

6.2 Returning to Flash-resident bootloader

To return to the Kinetis bootloader interface, simply hold SW3 and press and release the "Reset" button on the target board. When the device resets, the Kinetis bootloader will detect the press on SW3 and not jump to the user application. Verify you are in bootloader mode by again running the blhost.exe tool as done earlier.

```
C:\Freescale\Kinetis_Bootloader\bin\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 = 1258357504 (0x4b010300)
Current Version = K1.3.0
```

Figure 14. Back to the Kinetis bootloader interface

Pressing the "Reset" button alone will allow the Kinetis Bootloader to again launch the led_demo application.

7 Appendix A Kinetis flash-resident bootloader operation

This section describes the linkage between the Kinetis flash-resident bootloader and the user application. The demonstration described above illustrates a fairly simple collaboration between the Kinetis bootloader and the led_demo application. The considerations are:

- The flash-resident bootloader is located in flash at address 0.
- The user application is located in flash above the bootloader at BL_APP_VECTOR_TABLE_ADDRESS as defined in <install_dir>/apps/targets/<mcu>/src/bootloader_config.h
- The vector table for the User Application must be placed at the beginning of the application image.
- The Bootloader Configuration Area (BCA) must be placed at 0x3C0 from the beginning of the image.

7.1 Memory map overview

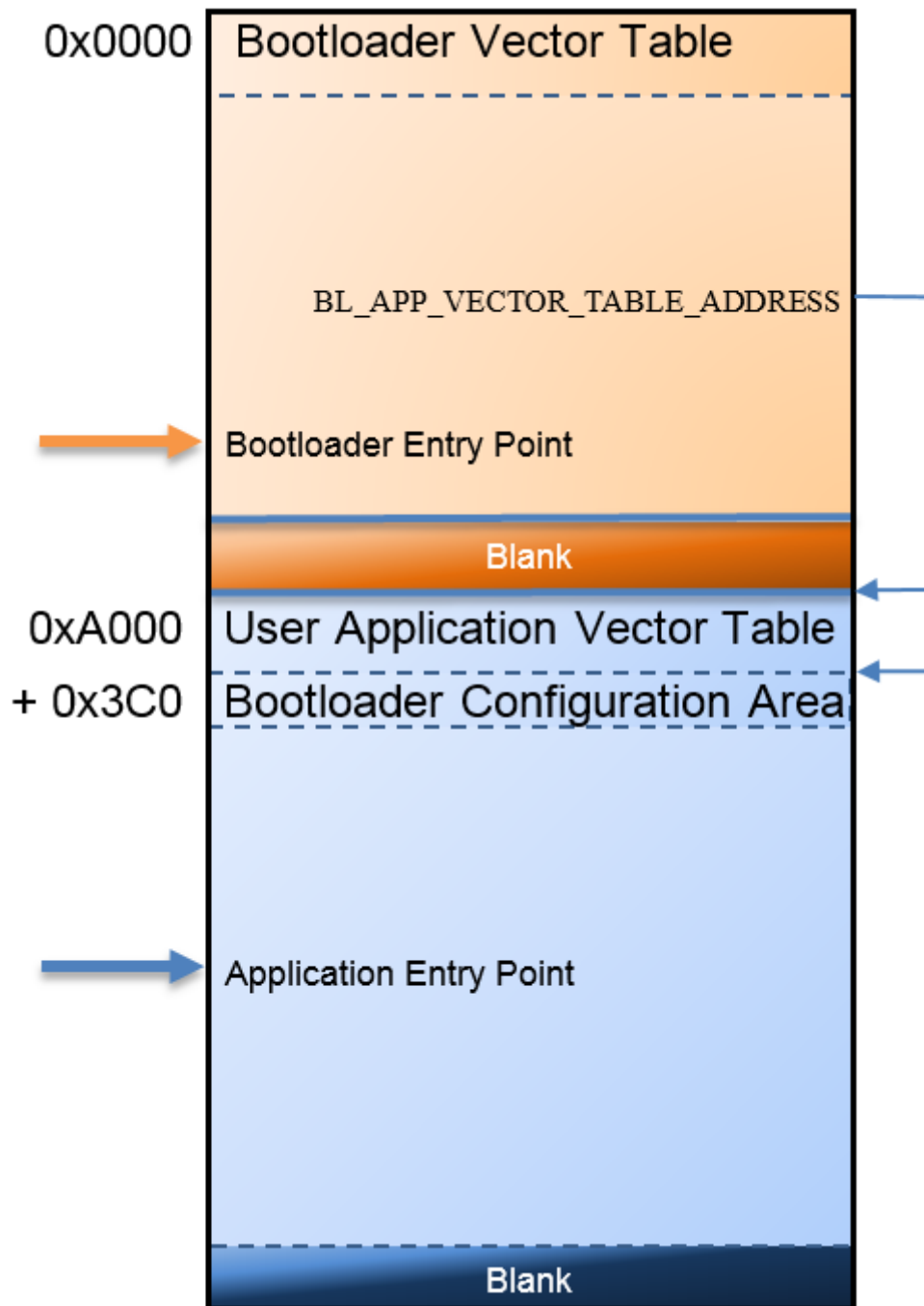


Figure 15. Device memory map

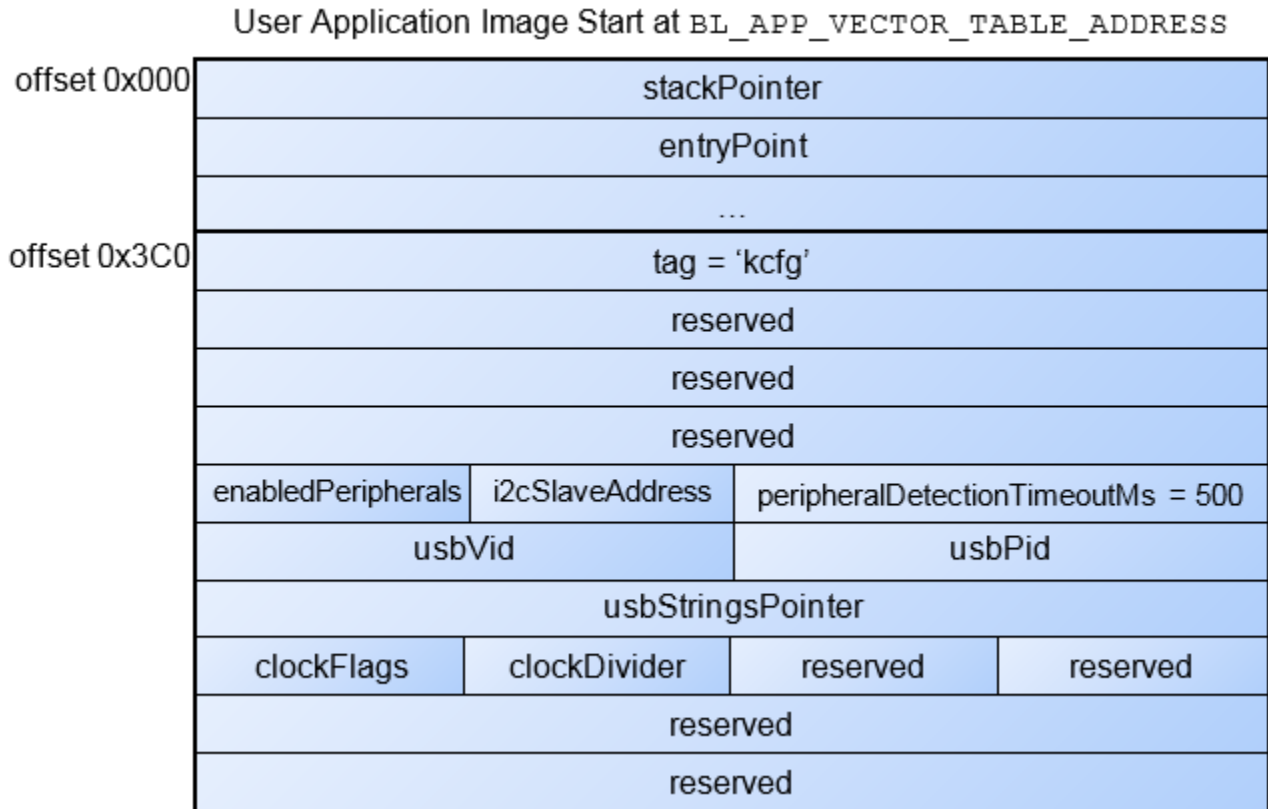


Figure 16. User application vector table and Bootloader Configuration Area (BCA)

7.2 User application vector table

The Kinetis bootloader checks `BL_APP_VECTOR_TABLE_ADDRESS+0` for the User Application stack pointer and `BL_APP_VECTOR_TABLE_ADDRESS+4` for the User Application entry point. Initially, this area is expected to be erased (0xFF) and the bootloader remains in its command interface.

Once a User Application is installed to `BL_APP_VECTOR_TABLE_ADDRESS`, the bootloader jumps to the application after a period specified by `peripheralDetectionTimeoutMs` in the Bootloader Configuration Area (BCA).

7.3 Bootloader Configuration Area (BCA)

The Bootloader Configuration Area is located at offset 0x3C0 from the beginning of the User Application image. This information is read by the Kinetis bootloader early during the bootloader initialization in order to set up clocks and gather other information relevant to detecting active peripherals. If the first four bytes of the BCA are not 'kcfg', the bootloader does not use any information from the BCA on flash.

For this tutorial, the `led_demo` application set the tag field to 'kcfg' and the `peripheralDetectionTimeoutMs` to 500 so that the bootloader would only wait 500 milliseconds before launching the `led_demo` application instead of the default 5 second wait period.

8 Revision History

This table summarizes revisions to this document.

Table 1. Revision History

Revision number	Date	Substantive changes
Rev. 0	12/2014	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
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 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.

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

© 2014 Freescale Semiconductor, Inc.

