AN3990
Application note

Upgrading STM32F4DISCOVERY board firmware using a USB key

Introduction

An important requirement for most Flash memory-based systems is the ability to update the firmware installed in the end product. This document provides general guidelines for creating a firmware upgrade application based on the STM32F4DISCOVERY board.

The STM32F4 series microcontroller can run user-specific applications to upgrade the firmware of the microcontroller-embedded Flash memory. This feature allows the use of any type of communication protocol for the reprogramming process (for example, CAN, USART and USB). USB Host mass storage is the example used in this application note.

The firmware upgrade using a USB Host is very advantageous because it is a standalone executed code in which the user does not need to use a host computer to perform the firmware upgrade. The user only needs a Flash disk to upgrade the target STM32 device.

Document contents

● *Section 1: Firmware upgrade overview* contains an overview of the firmware upgrade process and demonstrates how to run the firmware upgrade.

● *Section 2: How to use the firmware upgrade application* describes the user program and system requirements for the software and hardware.

Reference documents

● STM32F4DISCOVERY STM32F4 high-performance discovery board (UM1472)

● STM32F405xx, STM32F407xx, STM32F415xx and STM32F417xx advanced ARM-based 32-bit MCUs reference manual (RM0090)

● STM32F405xx STM32F407xx datasheet

● STM32F415xx STM32F417xx datasheet

The above documents are available at www.st.com/stm32f4-discovery.

1 Firmware upgrade overview

To program the firmware upgrade application to the Flash memory, use the STM32F4xx's embedded Bootloader or any in-system programming tool to easily reprogram this application.

The firmware upgrade application uses the USB Host to:

● Download a binary file (.bin) from a Flash disk (thumb drive) to the STM32F4xx's internal flash memory.

● Upload all the STM32F4xx's internal Flash memory content into a binary file.

● Execute the user program.

Note: *This application note is based on the STM32 USB On-The-Go (OTG) Host and device library. For more details about the USB Host stack and a mass storage demonstration, please refer to user manual (UM1021).*

1.1 Implementing the firmware upgrade application

The firmware upgrade application contains the source files in *Table 1*.

Table 1. Source files

File	Contents
`main.c`	Contains the USB initialization data. The USB Host state machine is then executed if the user wants to execute the firmware upgrade application or the program will execute the user code
`stm32f4xx_it.c`	Contains the interrupt handlers for the application
`command.c`	Contains the firmware upgrade commands (DOWNLOAD, UPLOAD and JUMP commands)
`flash_if.c`	Provides a medium layer access to the STM32 embedded Flash driver
`usb_bsp.c`	Implements the board support package for the USB Host library
`usbh_usr.c`	Includes the USB Host library user callbacks
`system_stm32f4xx.c`	Contains the system clock configuration for STM32 F4xx devices

After the board reset and depending on the user button state:

1. **User button pressed**: The firmware upgrade application is executed.
2. **User button not pressed**: A test on the user application start address will be performed and one of the below processes is executed.
   – User vector table available: User application is executed.
   – User vector table not available: firmware upgrade application is executed.

During the firmware upgrade application execution, there is a continuous check on the user button pressed state time. Depending on the state time of the user button, one of the following processes is executed.

Table 2.   User button state time control

User button state	Time	Process executed
Pressed	> 3 seconds	UPLOAD command will be executed immediately after completed execution of the DOWNLOAD command.
	< 3 seconds	Only the DOWNLOAD command is executed.

Note: *The UPLOAD command condition verification is signaled by the blinking state of the blue LED.*

Figure 1 illustrates the Firmware upgrade application flowchart.

Figure 1.   Upgrade application flowchart
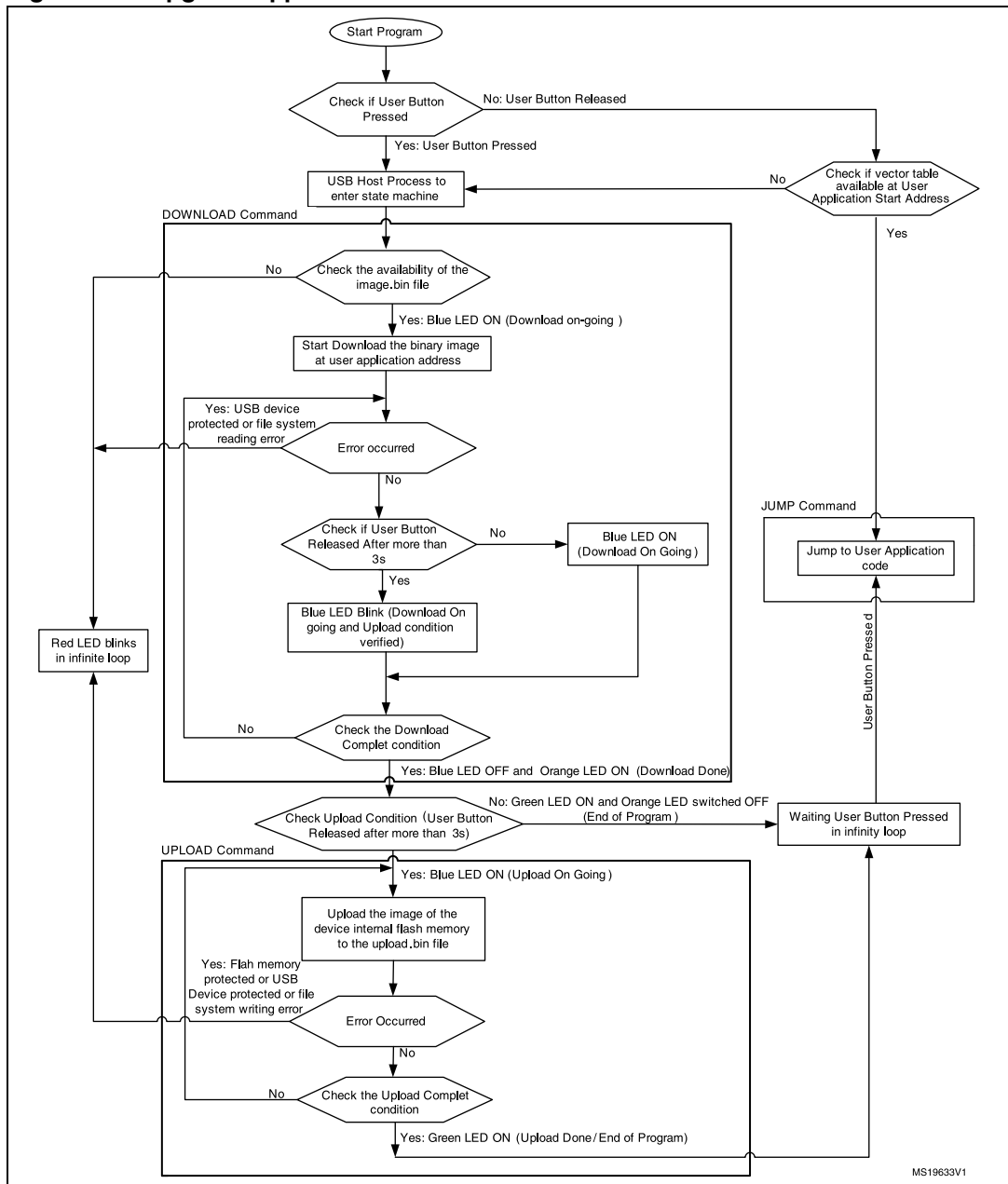
Note: *To execute the UPLOAD command the user button should be kept pressed 3s just after a board reset, at firmware startup.*

1.2 LEDs status

The following section describes the LEDs behaviors during the firmware upgrade application execution:

- Red LED blinks in infinite loop
  - Error (USB key disconnected, binary file not available or FATFS file system error).
- Red LED blinks in infinite loop and Blue LED ON
  - Error (No available Flash memory size to load the binary file).
- Red LED blinks in infinite loop, Blue LED ON and Orange LED ON
  - Flash erase error.
- Blue LED ON
  - DOWNLOAD ongoing.
- Blue LED ON and Orange LED ON
  - DOWNLOAD done; UPLOAD ongoing.
- Blue LED blinks
  - UPLOAD condition verified and the user should release the user button.
- Orange LED ON
  - DOWNLOAD done.
- Orange LED ON, Blue LED ON and Red LED blinks in infinite loop
  - USB key read out protection ON.
- Green LED ON and Orange LED OFF
  - DOWNLOAD done with success; and the MCU waiting until you press the user button to execute the JUMP command.
- Green LED ON and Orange LED ON
  - DOWNLOAD and UPLOAD done with success; and the MCU waiting until you press the user button before execute the JUMP command.

1.3 Commands description

The firmware upgrade application commands are listed in *Table 3*.

Table 3. Supported commands

Command	Description
DOWNLOAD	Reads the defined image file "DOWNLOAD_FILENAME" from the thumb drive and writes it to the embedded Flash memory.
UPLOAD	Reads the entire embedded Flash memory and saves the contents to the defined file name "UPLOAD_FILENAME" in the thumb drive.
JUMP	Executes the user code at the defined user application start address "APPLICATION_ADDRESS".

Note: *The maximum length for the file names (DOWNLOAD_FILENAME, UPLOAD_FILENAME) should be 11 characters as the LFN feature on the FAT file system is a patent of Microsoft® Corporation and when enabling it on commercial products, a license from Microsoft may be required depending on the final destination.*

1.3.1 DOWNLOAD command

To download a binary file from the flash thumb drive to the internal STM32F4xx's embedded flash memory the flowchart in *Figure 2: DOWNLOAD command* is applied.
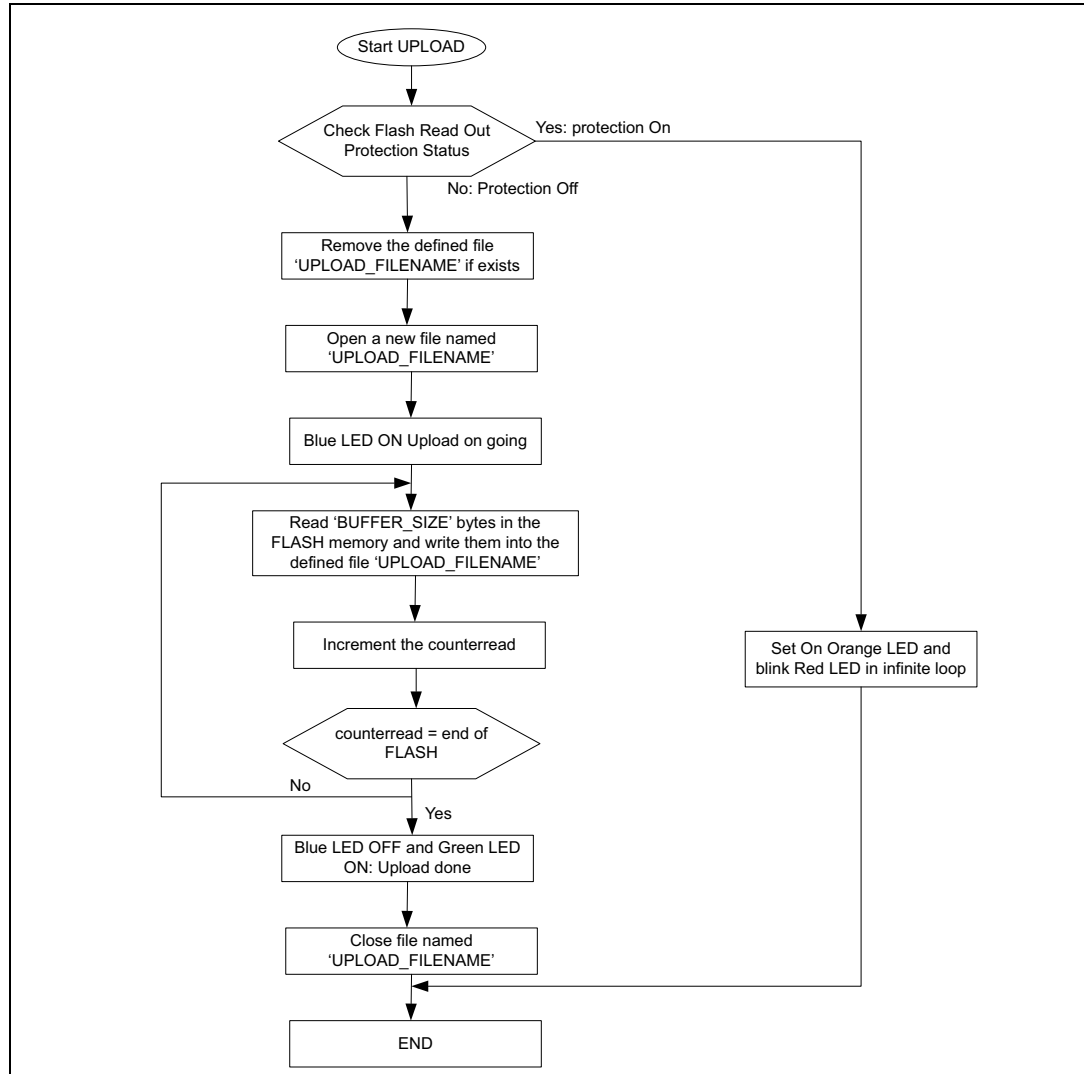
Figure 2. DOWNLOAD command
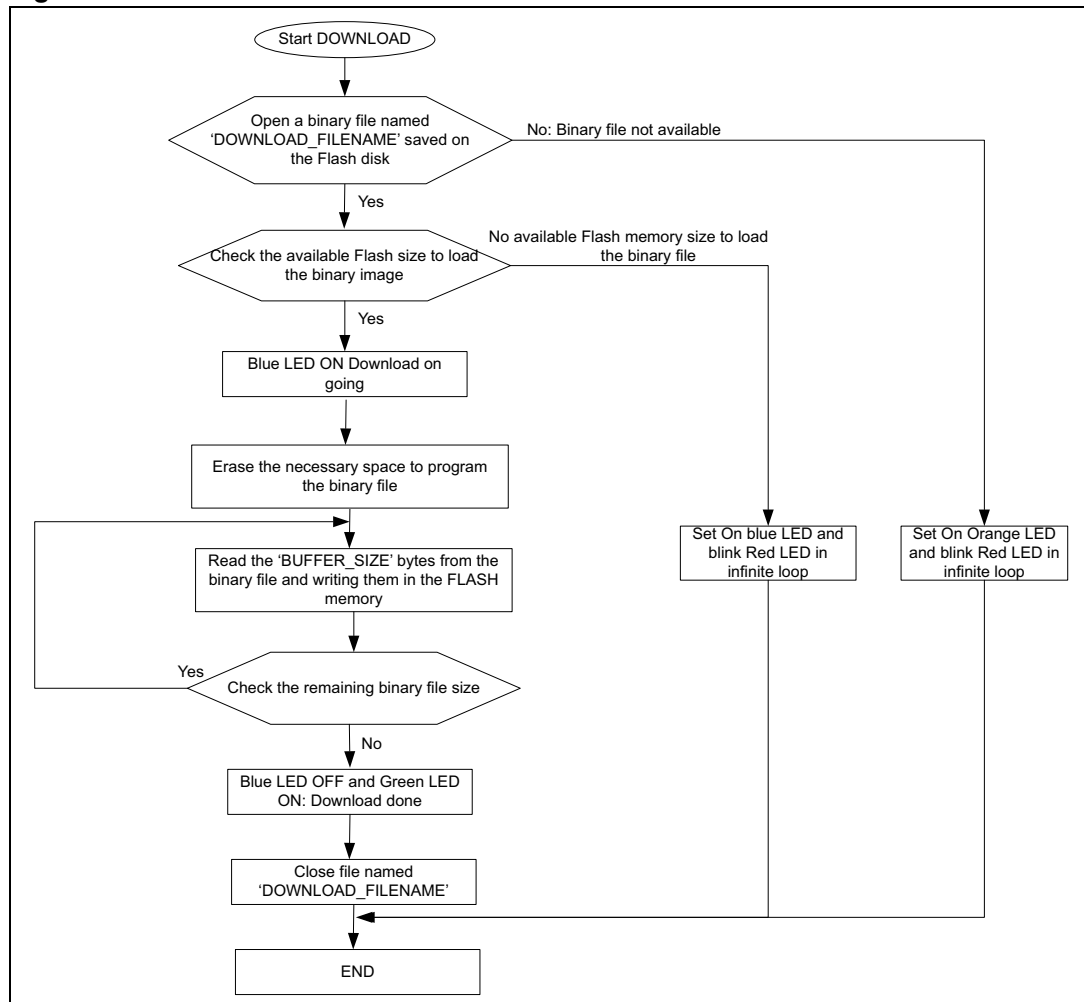
Note: 1 *BUFFER_SIZE is a user defined variable in the* `usbh_usr.h` *file that can be modified at compilation. BUFFER_SIZE = 512 * x; where x = [1,128] limited by firmware.*

2 *The DOWNLOAD command perform the erase of the required flash memory space then perform the program of the defined binary file using the flash word programming mode.*

3 *The Erase operation is performed for the FLASH memory space starting from the user application start address until the end of the flash memory.*

1.3.2 UPLOAD command

Figure 3 shows how to upload a copy of the internal Flash memory.

Figure 3. UPLOAD command

Note: 1   *BUFFERSIZE is a user-defined variable in the* `usbh_usr.h` *file that can be modified at compilation. BUFFERSIZE = 512 * x; where x = [1,128] limited by firmware.*

2   *When the user selects the UPLOAD command, the old* `UPLOAD.BIN` *file will be deleted and replaced by a new one that contains the new Flash memory data.*

3   *To execute the UPLOAD command the user button should be kept pressed 3s just after a board reset, at firmware startup.*

1.3.3 JUMP command

Once the new program has been loaded, you can use the JUMP command to execute this image which must be defined from this flash address: 0x08008000. Otherwise, the user must adapt the firmware to JUMP to another address. *Figure 4* illustrates the JUMP command flowchart.

Figure 4.   JUMP Command
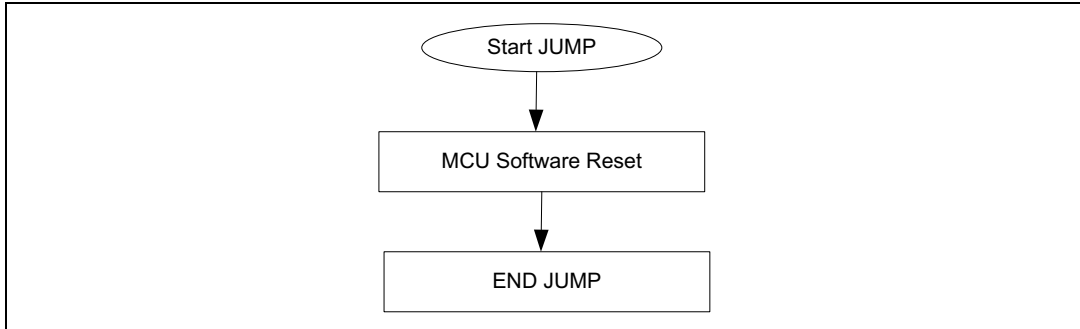
Note: Once the previous command(s) are performed with success the user green LED is ON and the MCU waits until the user button is pressed before executing the JUMP command.
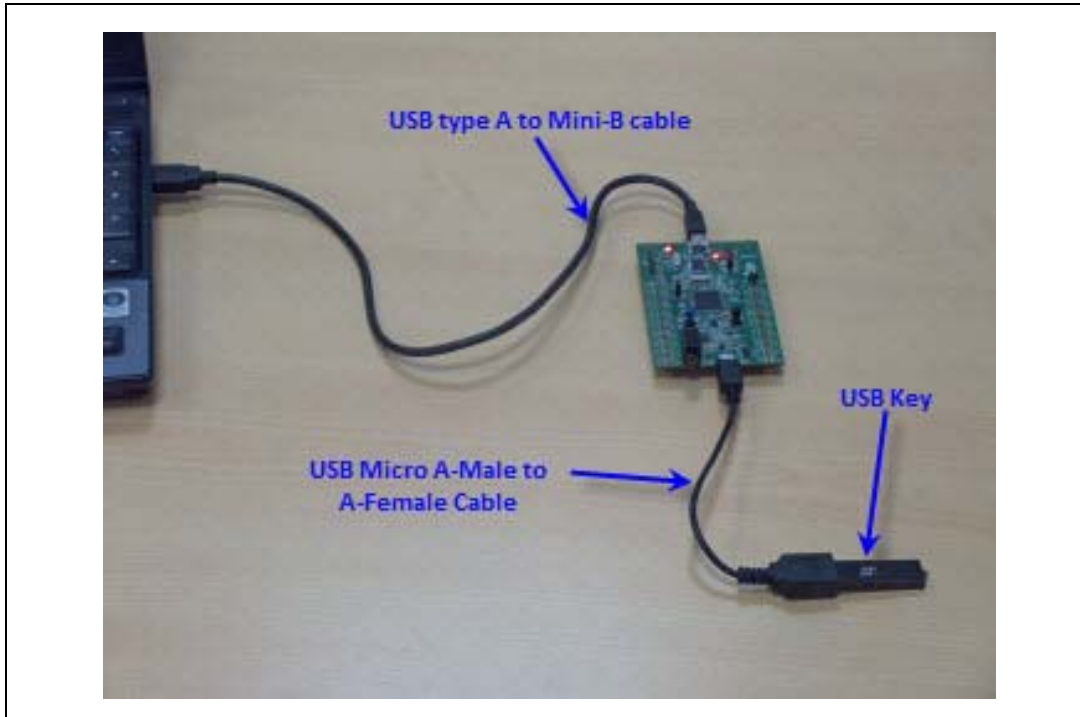
2   How to use the firmware upgrade application

2.1   System requirements

Before running your application, you should establish the connection with the STM32F4DISCOVERY board as following in *Figure 5*.

Figure 5.   Hardware environment

To run the firmware upgrade application on your STM32F4DISCOVERY board, the minimum requirements are as follows:

● Microsoft® Windows PC (2000, XP, Vista, 7)
● USB type A to Mini-B' cable, used to power the board (through USB connector CN1) from host PC and connect to the embedded ST-LINK/V2 for debugging and programming.
● USB micro A-Male to A-Female' cable, used to connect the USB key (through USB connector CN5) as USB Device to host STM32F4xx.

2.2 Running the firmware upgrade application

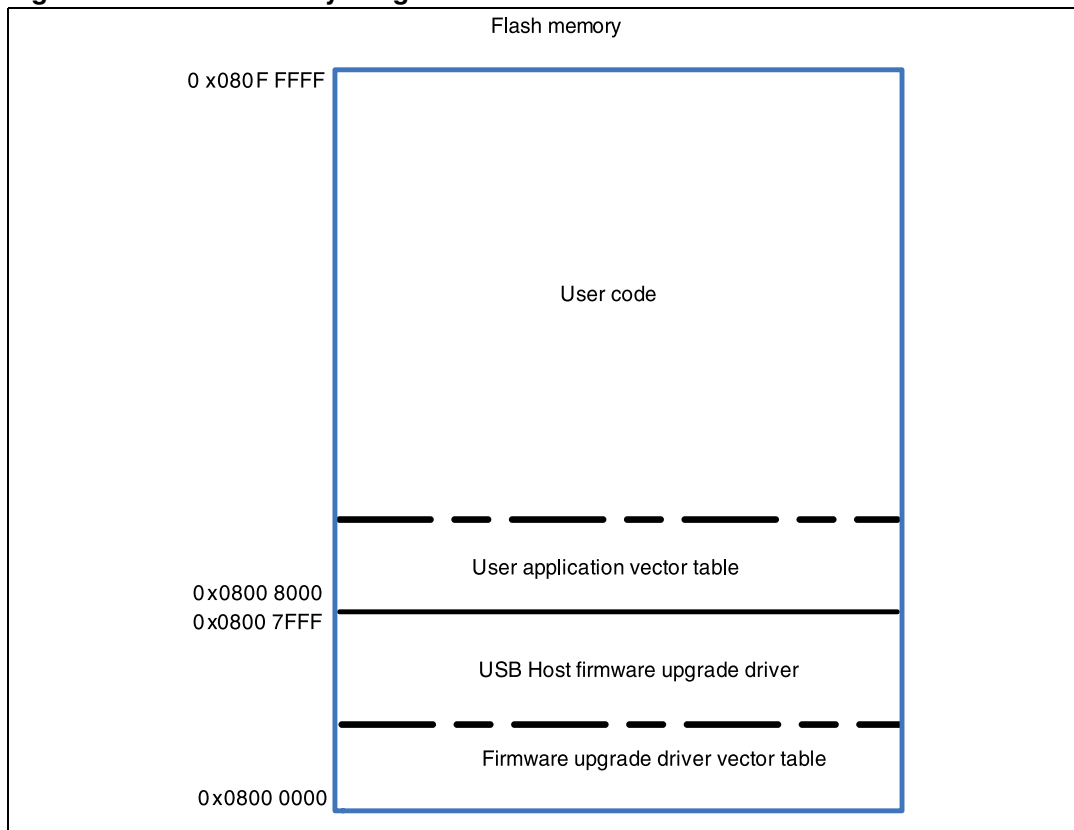To run the firmware upgrade application, proceed as follows:

1. Load the binary image of the user program to the root directory of a USB key. You can use the provided binary images (STM32F4-Discovery_xxxx_0x08008000.bin) under the Project\FW_upgrade\Binary folder. The binary should be renamed to "image.bin".

2. Program the firmware upgrade application into the internal Flash memory.

   a) Open the project (under Project\FW_upgrade) with your preferred toolchain.

   b) Compile and load the project into the target memory and run the project.

   c) Another option is to use the embedded Bootloader or any in-system programming tool to easily reprogram this application.

   – Use STM32F4-Discovery_FW_Upgrade_V1.0.0.hex with "in-system programming tool" such as STM32 ST-LINK Utility.

   – Use STM32F4-Discovery_FW_Upgrade_V1.0.0.dfu with "DFUse\DFUse Demonstration" tool.

   – For more details, please refer to the "Binary images for reprogramming firmware applications" section of UM1467.

3. Plug the USB key into the STM32F4DISCOVERY board through 'USB micro A-Male to A-Female' cable.

4. Follow the description provided in section *Section 1.1: Implementing the firmware upgrade application on page 2*.

2.3   User program condition

The user application (binary file) to be loaded into the Flash memory using the firmware upgrade application is built by the following configuration settings:

1.   Set the program load address to APPLICATION_ADDRESS in the toolchain linker file.
2.   Relocate the vector table to address APPLICATION_ADDRESS using the NVIC_SetVectorTable function or the VECT_TAB_OFFSET definition inside the `system_stm32f4xx.c` file.

Figure 6.   Flash memory usage

Note: 1   Be sure that APPLICATION_ADDRESS do not overlap with firmware upgrade application. For example, with all options enabled the total read-only memory size using EWARM compiler v6.21, with high optimization for size, is 17 936 bytes (17 744 bytes of read-only code memory and 192 bytes of read-only data memory). With these results, two sectors of 16 Kbytes each will be used to store the firmware upgrade application, so the user application flash start address will start from Sector2.

3 Frequently asked questions (FAQs)

How to change the name of the image to be loaded

The name of the binary file to be loaded in the USB Key can be changed by personalizing the "UPLOAD_FILENAME" definition in the `command.c` file.

How to change the name of the image to be downloaded

The name of the binary file to be downloaded in the internal flash memory at user application start address can be changed by personalizing the "DOWNLOAD_FILENAME" definition in the `command.c` file.

How to change the user application start address

The user application start address can be changed by personalizing the "APPLICATION_ADDRESS" definition in the `flash_if.h` file.

Note: *By editing the user application start address, make sure that the user program conditions defined in Section 2: How to use the firmware upgrade application on page 9 are respected.*

How to modify the size of the Flash memory to be uploaded

The size of the Flash memory to be uploaded can be changed by personalizing the "FLASH_SIZE" and the "FLASH_STARTADDRESS" definitions in the `flash_if.h` file.

When using the CPU frequency = 168MHz, how much time does the DOWNLOAD command take?

When the user application start address = 0x0800 8000 (start from sector 2) the erase operation takes 7.76s.

With BUFFER_SIZE = 512 * 64 = 32 Kbytes and image size = 25 Kbytes, the DOWNLOAD operation takes about 7.94 seconds (7.76s erase time + 0.18s program time).

With BUFFER_SIZE = 512 * 64 = 32 Kbytes and image size = 990 Kbytes, the DOWNLOAD operation takes about 12.7 seconds (7.76s erase time + 4.94s program time).

When using the CPU frequency = 168MHz, how much time does the UPLOAD command take?

With BUFFERSIZE = 512 * 64 = 32 Kbytes, uploading of all Flash memory (1Mbytes) takes about 1.5 seconds.

4 Revision history

Table 4. **Document revision history**

Date	Revision	Changes
24-Oct-2011	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.