ELMO Board Programming Guide

Version history			
Version	Date	Author	Changes
0.1	17.08.2015	CCZI	First draft
0.2	18.08.2015	CCZI	Added ToC, Clarified section 2.2: SB12-SB14 solder bridge removal, Changed section 2.4: 12 KB and file name FAIL.TXT
1.0	18.09.2015	CCZI	Release
1.1	25.09.2015	CCZI	Drag&drop programming only works with the F411 Nucleo, Added section 2.5 using the STM ST-LINK Utility.

Author: C. Cziezerski

Table of Contents

Programming the ELMO

This document describes how to program the ELMO with a bin file from the mbed.org compiler.

There are two ways to program the ELMO. Each of these methods is described in the following two sections of this document:

1. STM DfuSe Demonstrator via USB
2. mbed enabled with any ST-LINK device, e.g. STM Nucleo boards

.

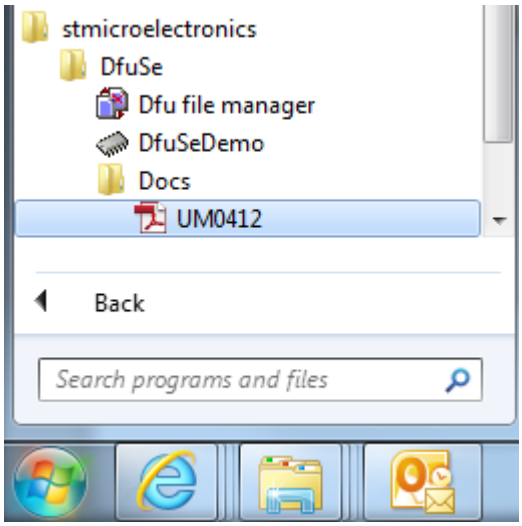
1. DFU mode with DfuSe Demonstrator

1.1. Prerequisites

- You have downloaded the .bin file from the mbed.org compiler.
- You have the DfuSe USB device firmware upgrade STMicroelectronics extension installed

Get the DfuSe Demonstrator (STSW-STM32080) version 3.0.4 software from ST's website: (12.1 MB)
http://www.st.com/web/en/catalog/tools/FM147/CL1794/SC961/SS1533/PF257916

If needed, familiarize yourself with the tool by reading UM0412.PDF.



1.2. A note on the DFU file format

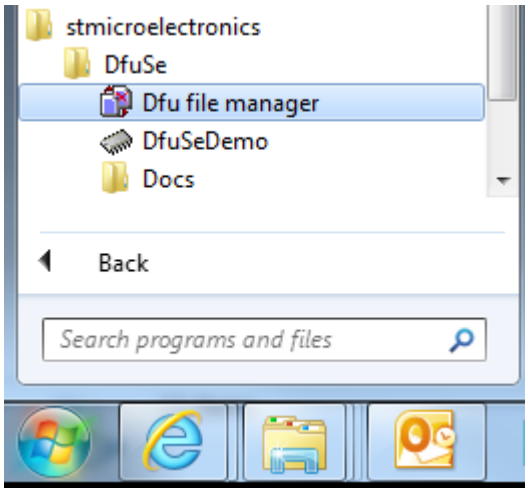
(From the installed document UM0412)

"Users that have purchased DFU devices require the ability to upgrade the firmware of these devices. Traditionally, firmware is stored in Hex, S19 or Binary files, but these formats do not contain the necessary information to perform the upgrade operation, they contain only the actual data of the program to be downloaded. However, the DFU operation requires more information, such as the product identifier, vendor identifier, Firmware version and the Alternate setting number (Target ID) of the target to be used, this information makes the upgrade targeted and more secure. To add this information, a new file format should be used, to be called DFU file format. For more details refer to the "DfuSe File Format Specification" document (UM0391)."

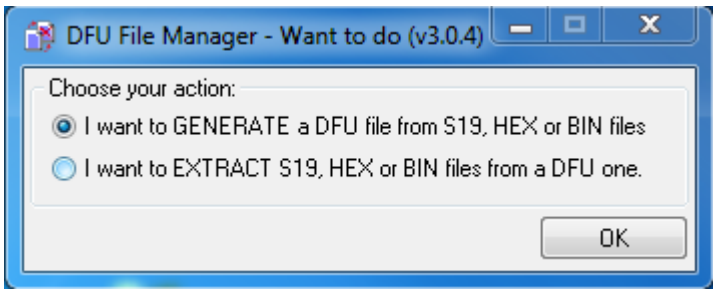
1.3. Generate a DFU file from a binary

The DFU tool requires a .dfu file in order to work. Here is how it is generated:

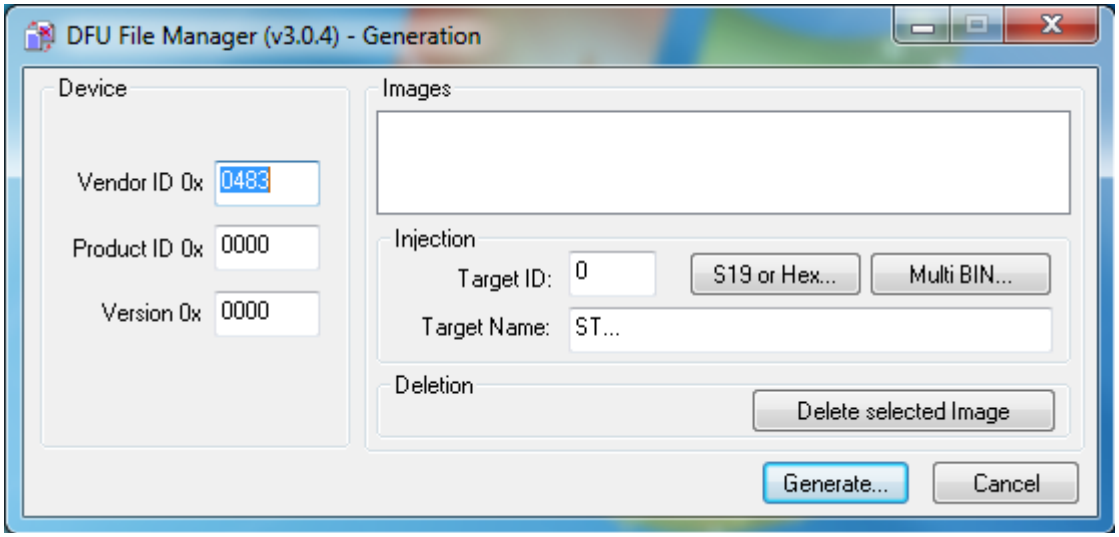
1. Start the Dfu file manager application



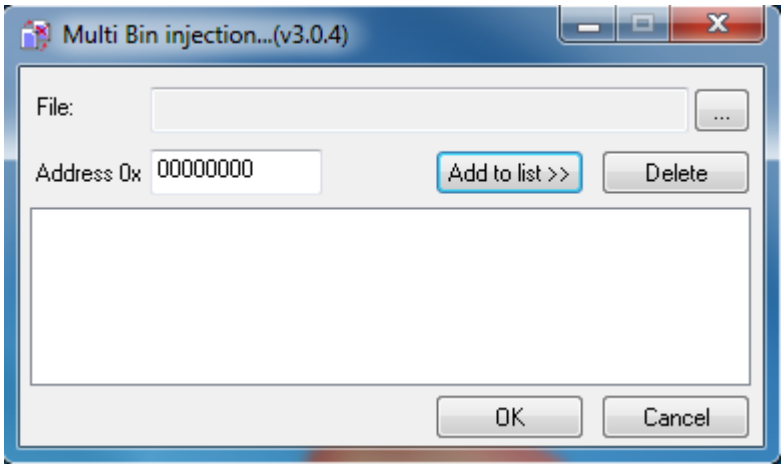
2. Choose the first option (generate a DFU file)



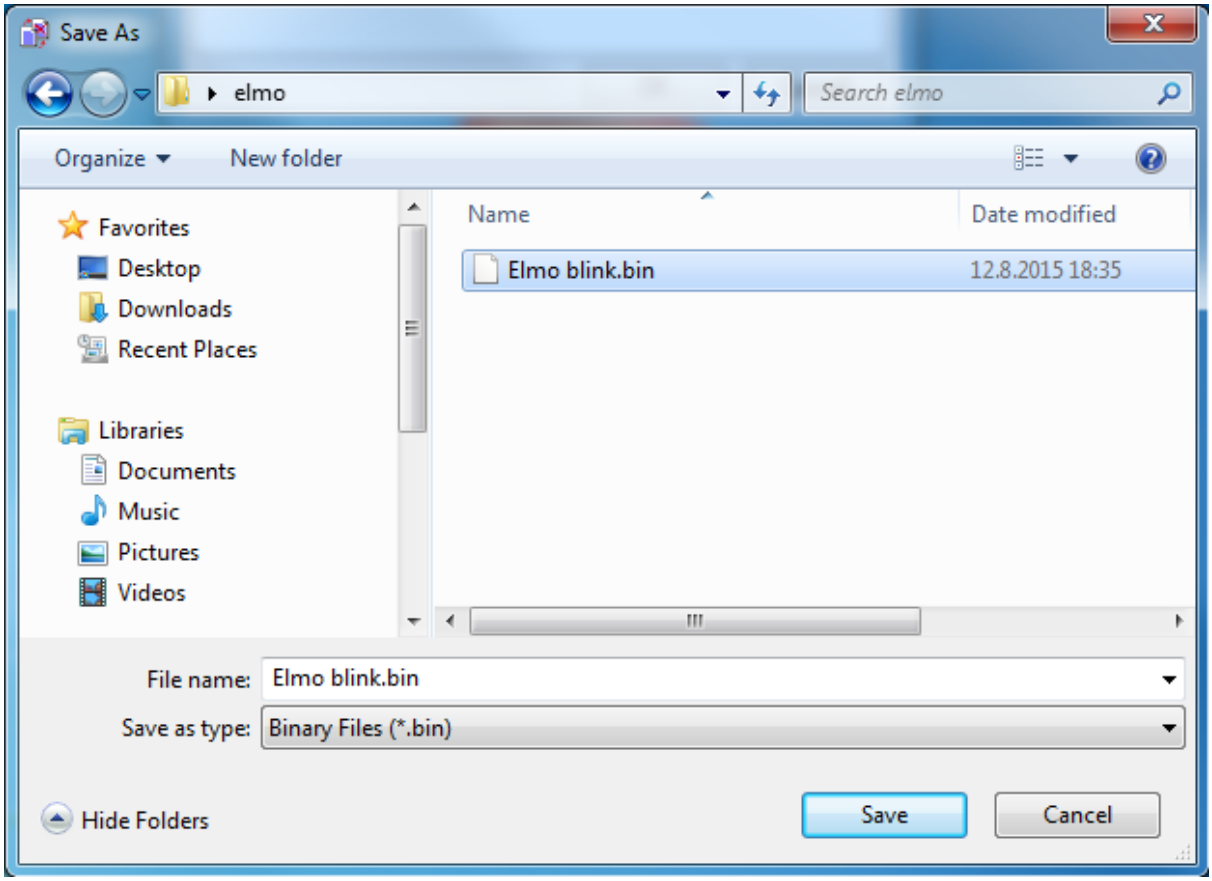
3. On the following GUI, select “Multi BIN…”



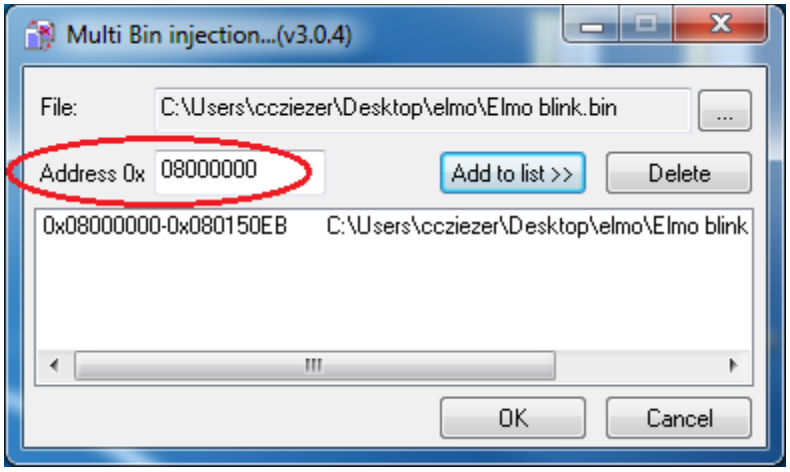
4. Select the "…" to browse for the source file



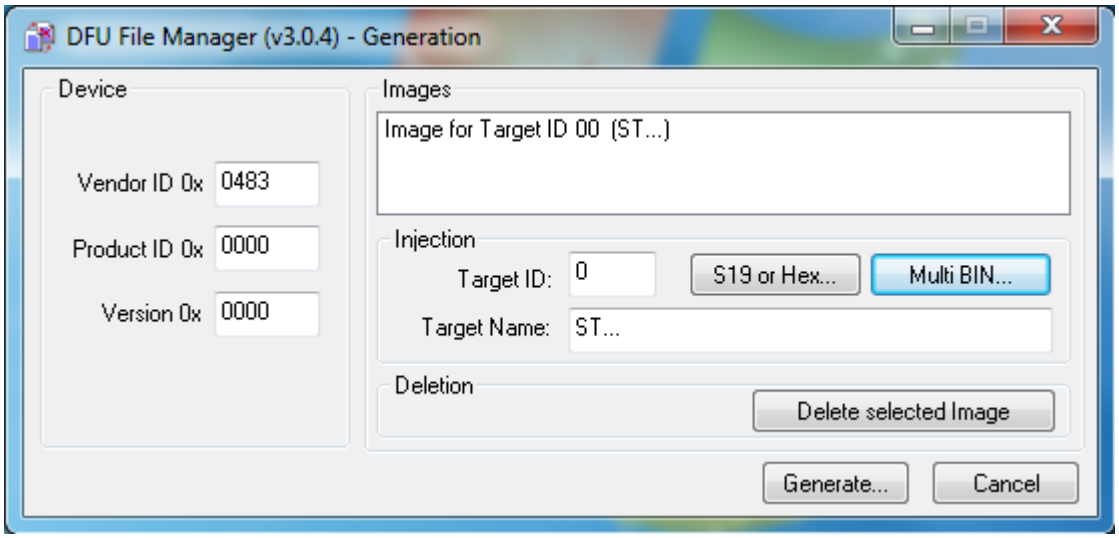
5. Select the .bin file and select "save". (the file will not be modified)



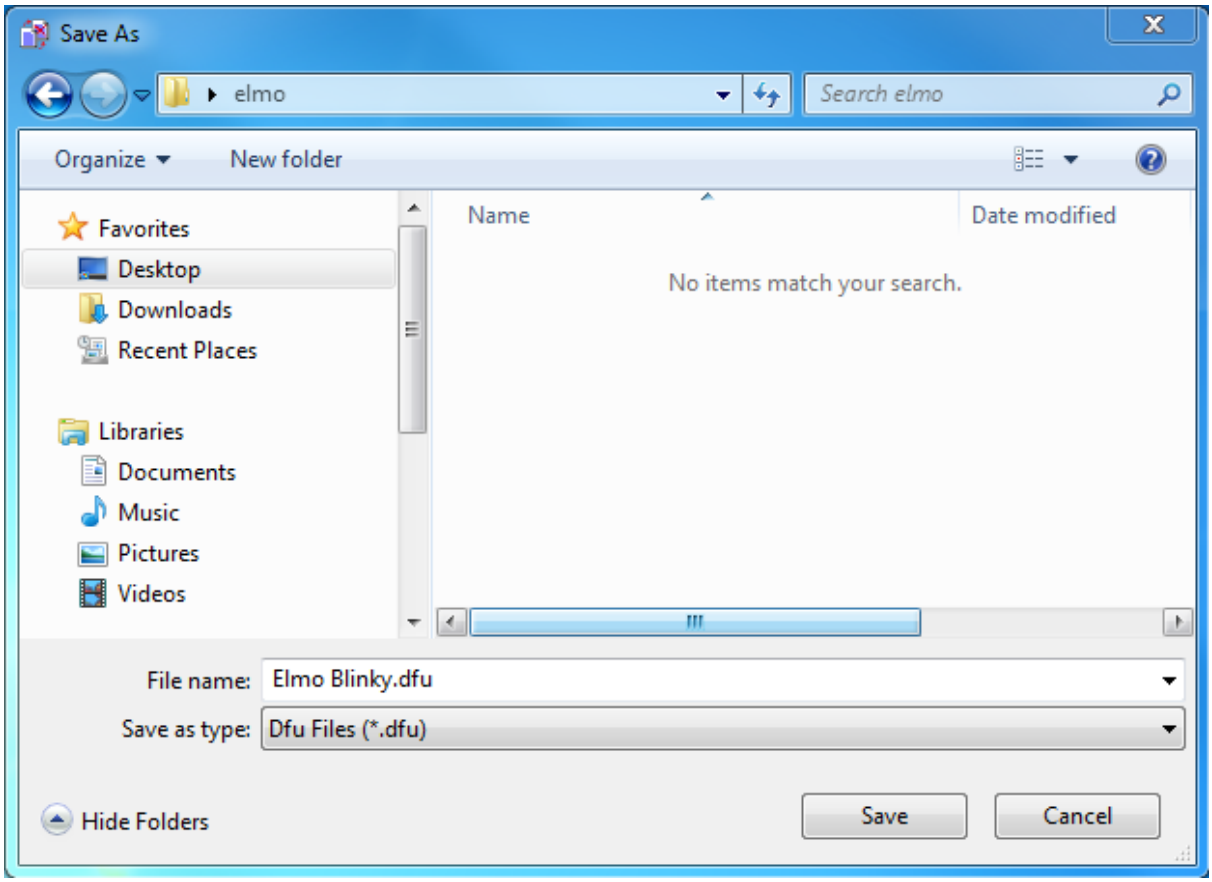
6. On the following GUI, change the Address to 0x08000000, i.e. the STM32F411's start address of the first sector of the internal flash, and add the file via "Add to list >>", so that the file appears in the centre list and choose "OK".



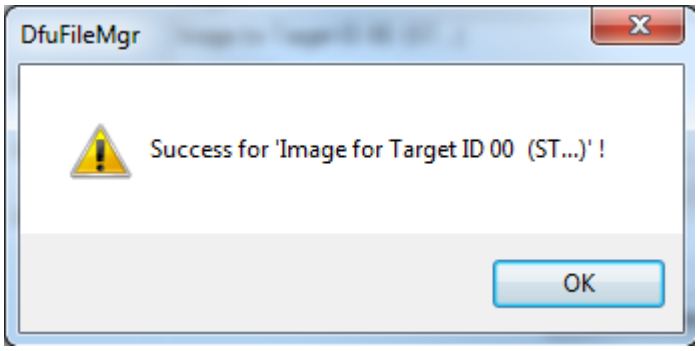
7. Finally, change the PID and version if required and select "Generate…" to generate the .dfu file.



8. Choose the destination folder and file name



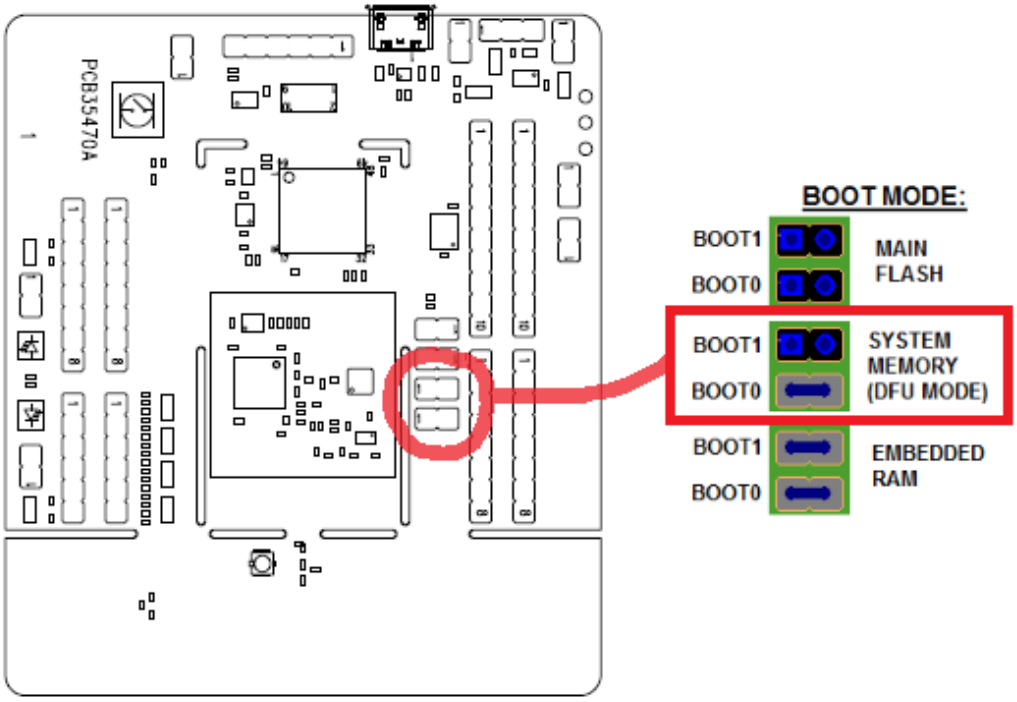
9. Success! (or maybe not…*)



* Note: if you can't find the file, look in the following folder:
C:\Users\(user name)\AppData\Local\VirtualStore\Windows\SysWOW64

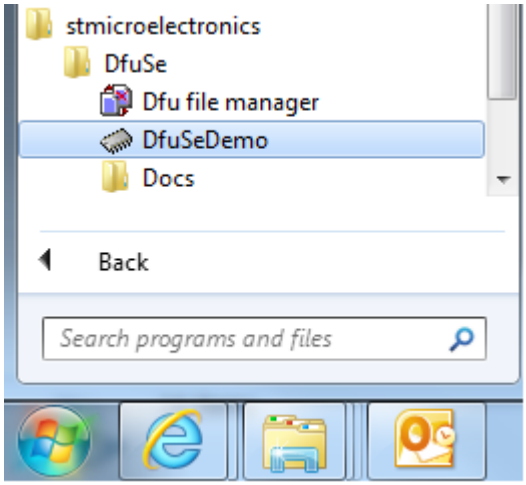
1.4. Configure the ELMO Board to load the STM32 DFU bootloader

Set the ELMO Board to start from the internal bootloader/DFU mode by setting the BOOT0 jumper.

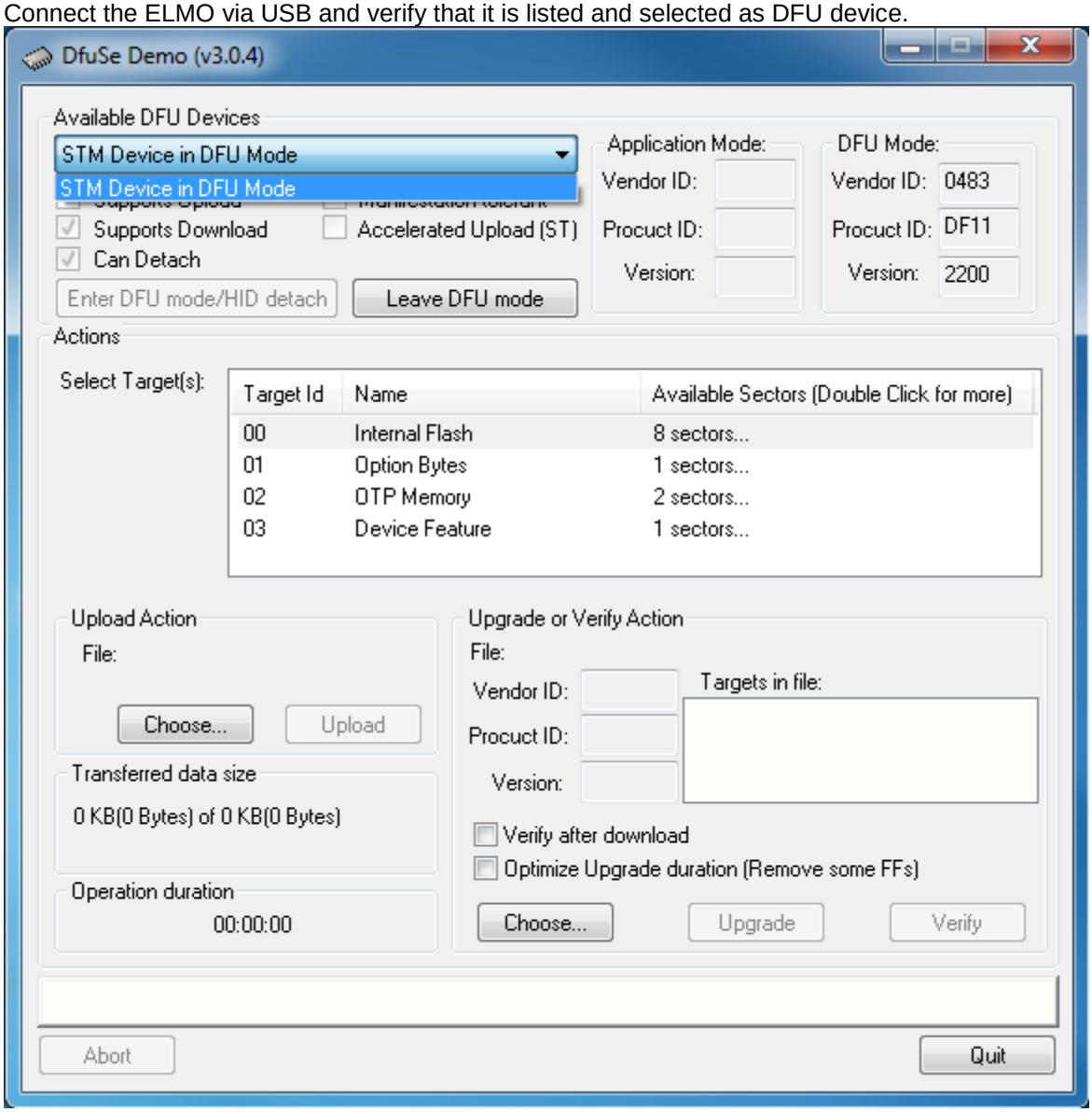


1.5. Program the ELMO via the on board USB connector J202

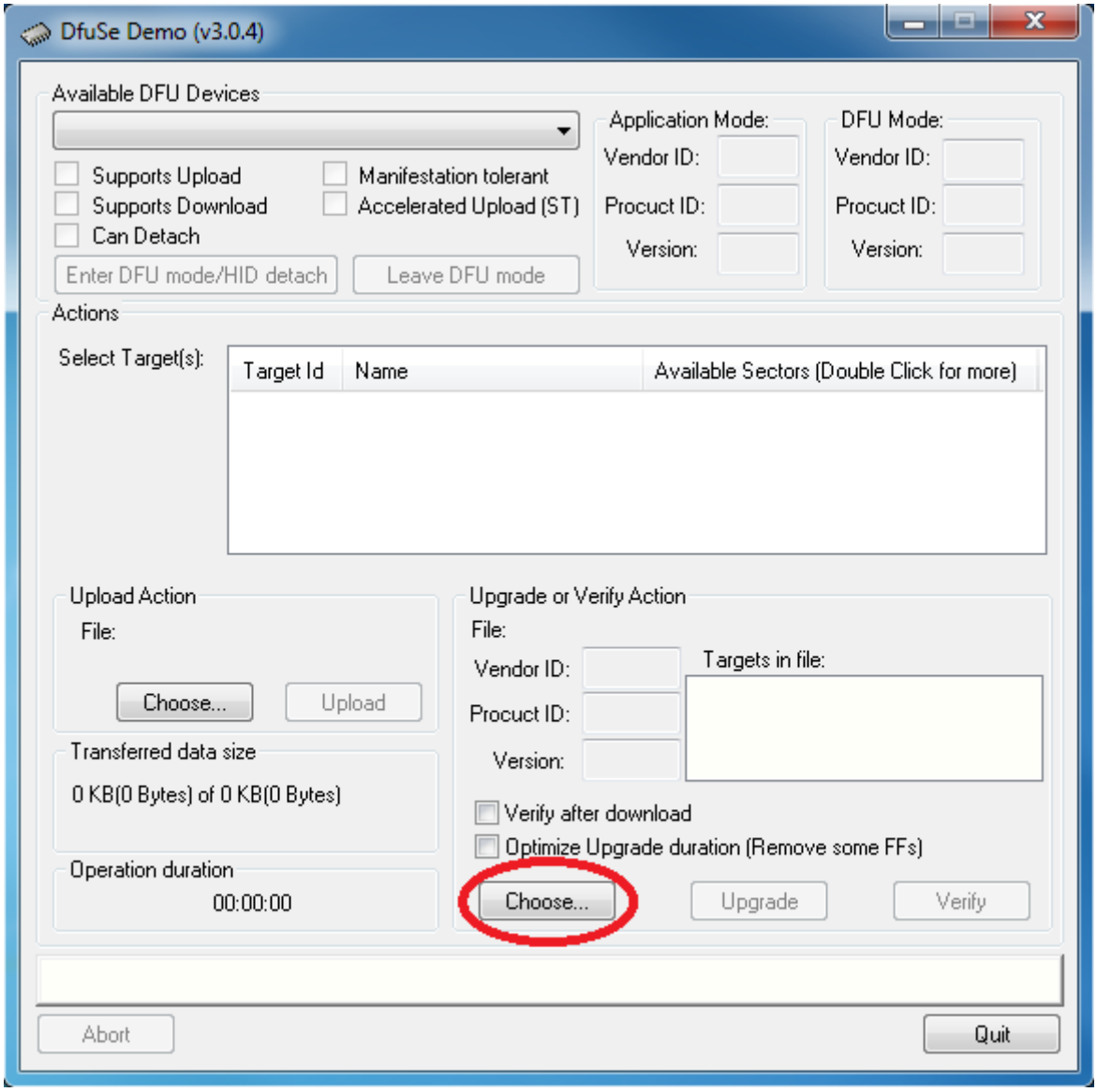
1. Start the firmware DfuSeDemo upgrade utility.



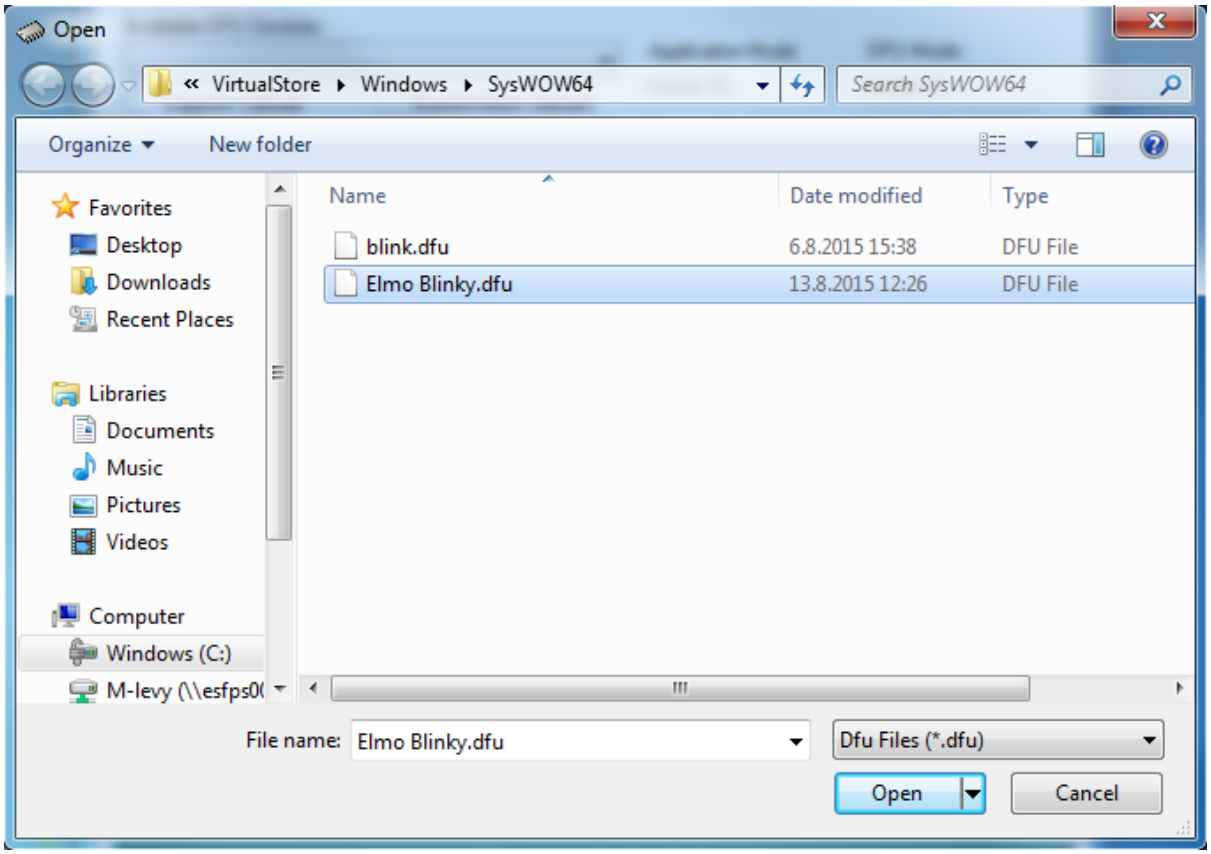
2. Connect the ELMO via USB and verify that it is listed and selected as DFU device.



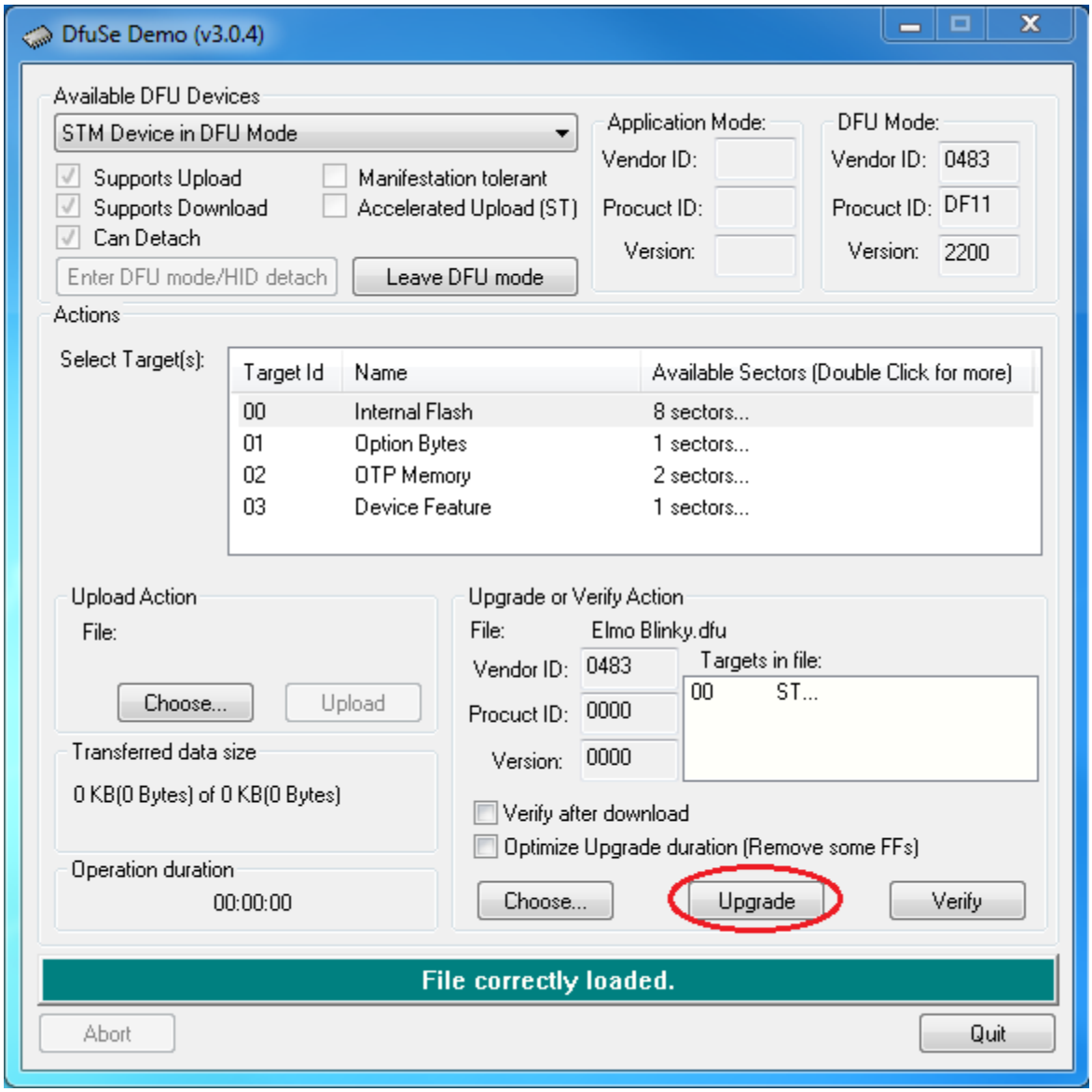
3. Select “Choose” in the “Upgrade or Verify Action” subsection to select the .dfu source file.



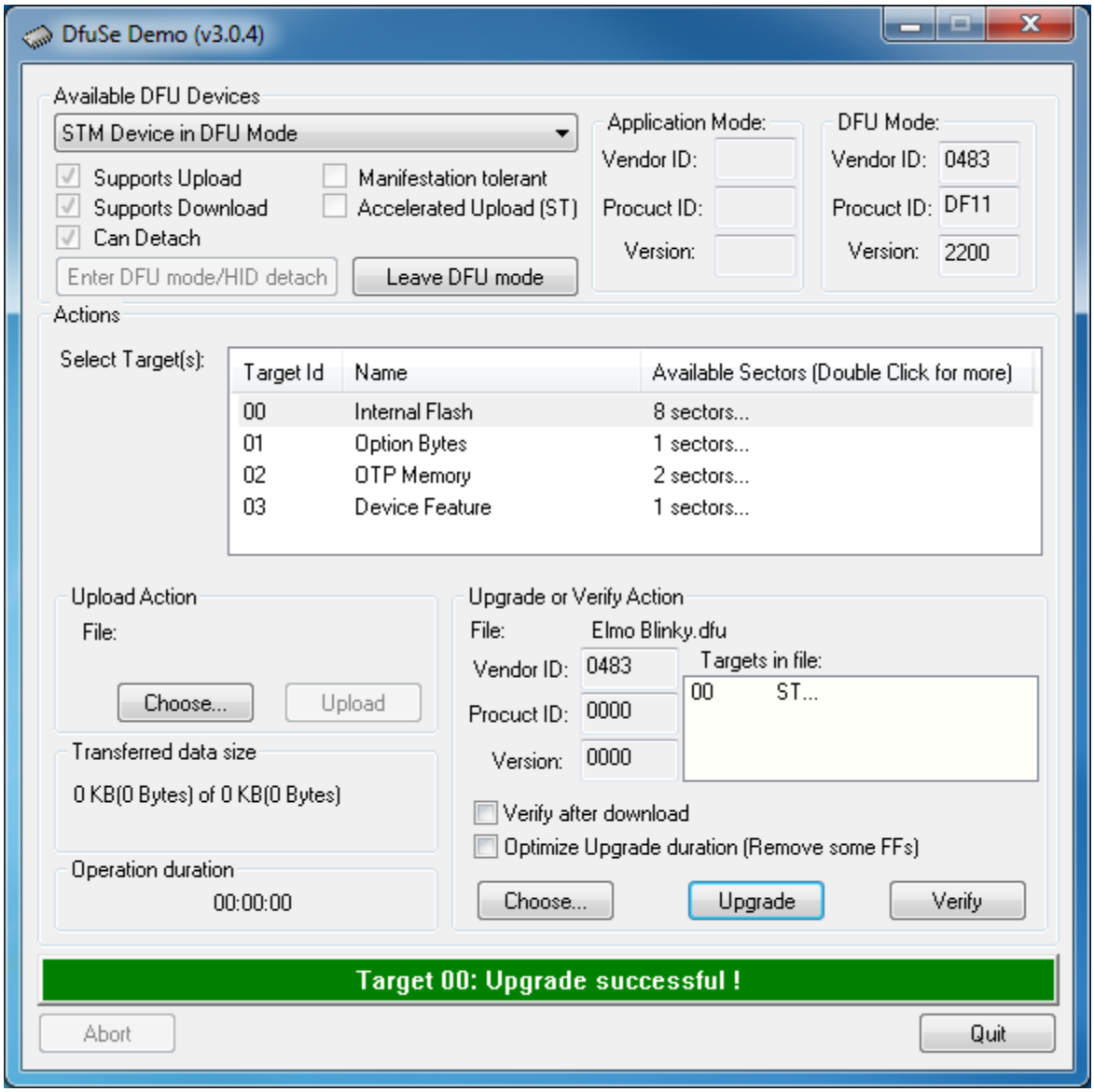
4. Select the file dfu source file and choose “Open”



5. Select “Upgrade” to program the ELMO.



6. Success!


DfuSe Demo (v3.0.4)
Available DFU Devices
STM Device in DFU Mode
Supports Upload
Supports Download
Can Detach
Manifestation tolerant
Accelerated Upload (ST)
Enter DFU mode/HID detach
Leave DFU mode
Application Mode:
Vendor ID:
Procuct ID:
Version:
DFU Mode:
Vendor ID: 0483
Procuct ID: DF11
Version: 2200
Actions
Select Target(s):
Target Id
Name
Available Sectors (Double Click for more)
00 Internal Flash 8 sectors...
01 Option Bytes 1 sectors...
02 OTP Memory 2 sectors...
03 Device Feature 1 sectors...
Upload Action
File:
Choose...
Upload
Transferred data size
0 KB(0 Bytes) of 0 KB(0 Bytes)
Operation duration
00:00:00
Upgrade or Verify Action
File: Elmo Blinky.dfu
Vendor ID: 0483
Procuct ID: 0000
Version: 0000
Targets in file:
00 ST...
Verify after download
Optimize Upgrade duration (Remove some FFs)
Choose...
Upgrade
Verify
Target 00: Upgrade successful !
Abort
Quit

2. mbed Enabled via ST-LINK/V2-1

It is possible to use the ST-LINK part of e.g. a Nucleo F411 board to program the ELMO in the familiar mbed-enabled manner. The ELMO Board will show up as mass storage device and can be programmed by drag & dropping the binary onto the drive.

For this, the nucleo board needs to be modified as described in the following section.

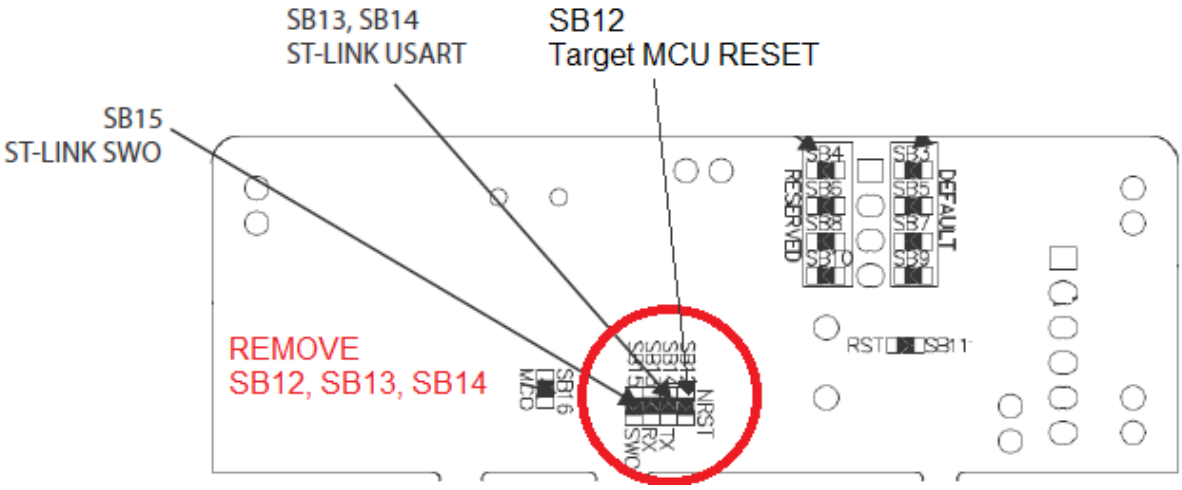
2.1. Prerequisites

- The Nucleo needs to be modified to connect to an external target.
- The ELMO Board and the Nucleo need to be connected via SWD

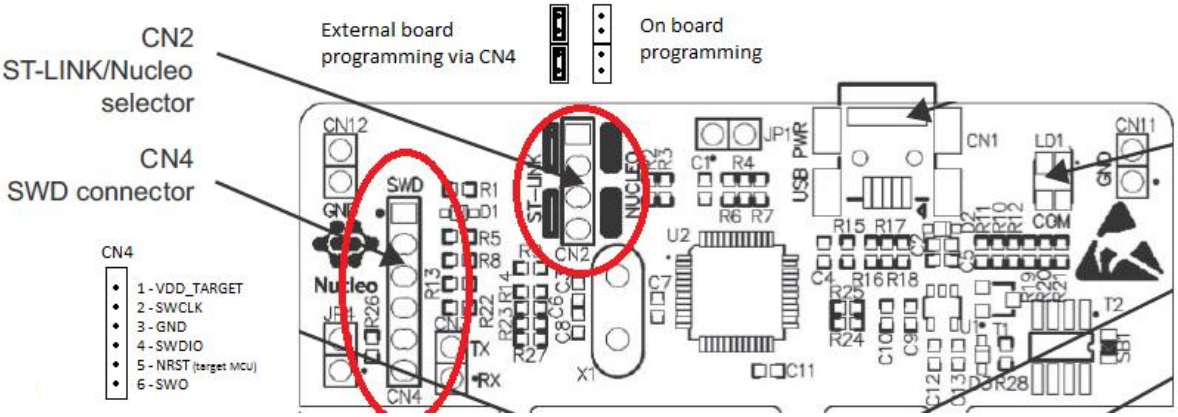
2.2. Modifying the Nucleo for External Board Programming

It is necessary to remove the solder bridges SB12, SB13 and SB14 on the bottom side of the Nucleo, so that the target micro controller on the Nucleo does not interfere with the external board.

SB13 and SB14 are not strictly needed for programming, but removing them means that the ELMO's serial debug header J613 can be connected to CN3 of the Nucleo's to support the Virtual Com Port for mbed.



Both jumpers on the CN2 selector on the top side of the Nucleo board need to be set.

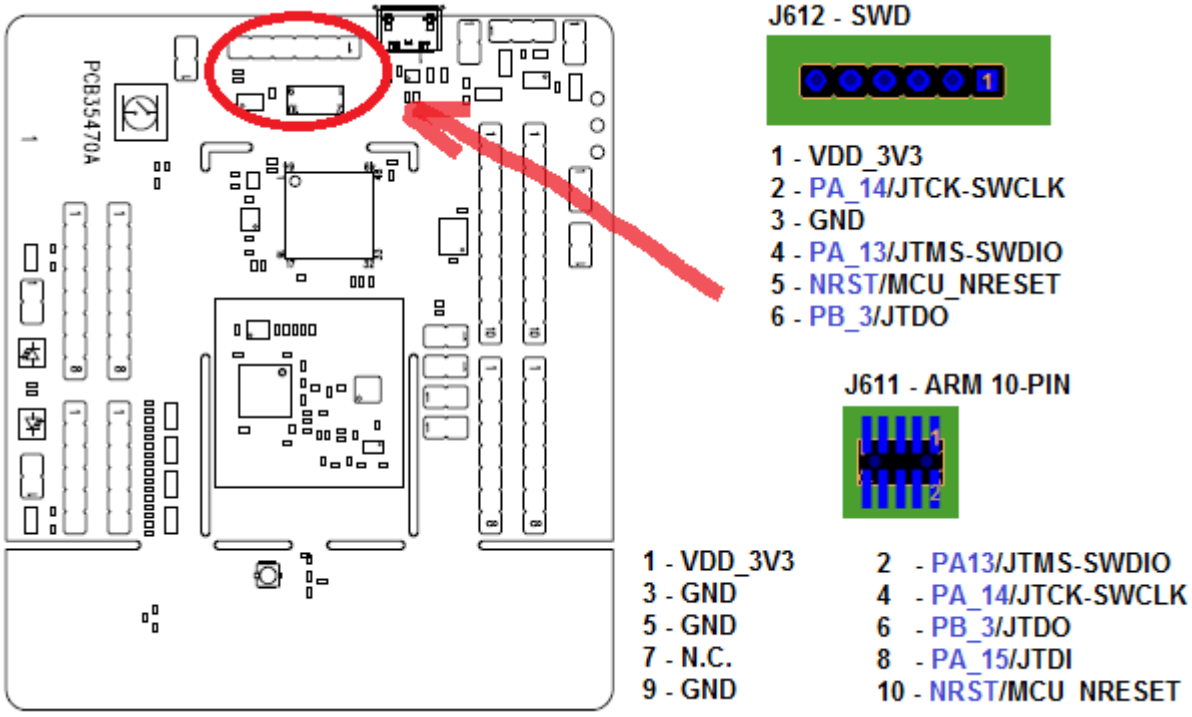


Now, connect the Nucleo's CN4 SWD header to the ELMO Board's J612 SWD header with a 6-pin cable with MKL 0.64/25-style connectors on both ends. Alternatively, a 6-pin to 10-pin SWD adapter cable can be used to connect to the ARM 10-pin header J611 instead.

The ELMO Board's SWD connector is explained in the following section.

2.3. ELMO SWD Connectors

Below are the SWD and JTAG/ARM 10-PIN headers that can be used to connect to a Nucleo board in order to support mbed-enabled programming of the ELMO Board.

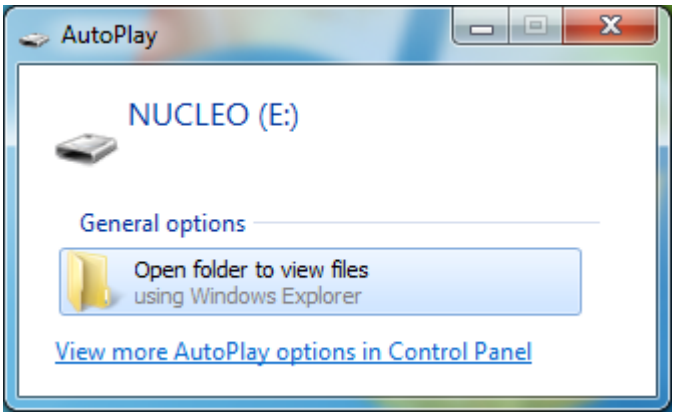


It is important to notice that the ELMO board needs its own power during the programming, since the SWD voltage VDD_TARGET has to be provided by the target itself.

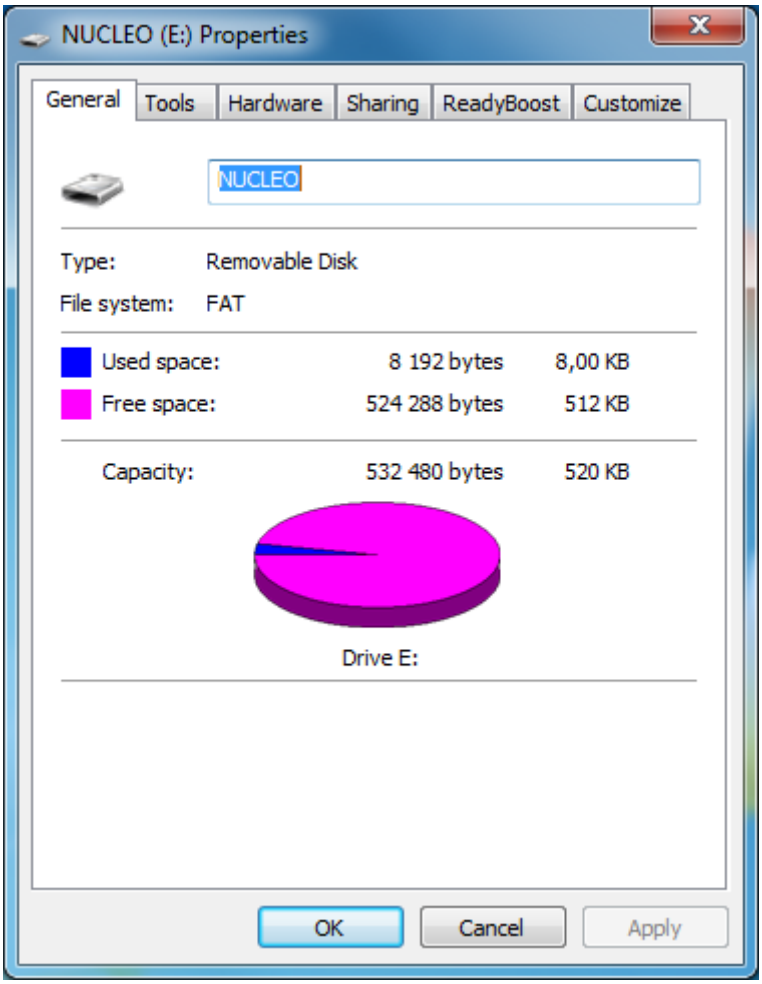
2.4. Program the ELMO by drag&drop (Nucleo F411)

Skip to section 2.5 if you are not using an STM Nucleo F411.

Once the ELMO board is powered and connected to the Nucleo, connect the Nucleo to the computer. The ELMO Board's FLASH will enumerate as mass-storage device on the computer. To program the ELMO simply drag and drop the .bin file into this storage. The ELMO Board will automatically be reset by the Nucleo once the programming is successful.



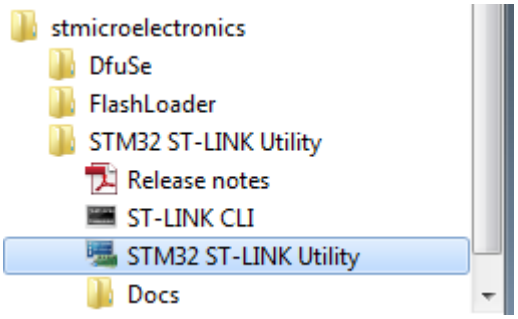
A quick check to see if the flash memory is mounted correctly can be done by opening the drive properties. If anything went wrong, the drive will be 12 KB big (for the Nucleo F411) and contains a FAIL.TXT file with more information.



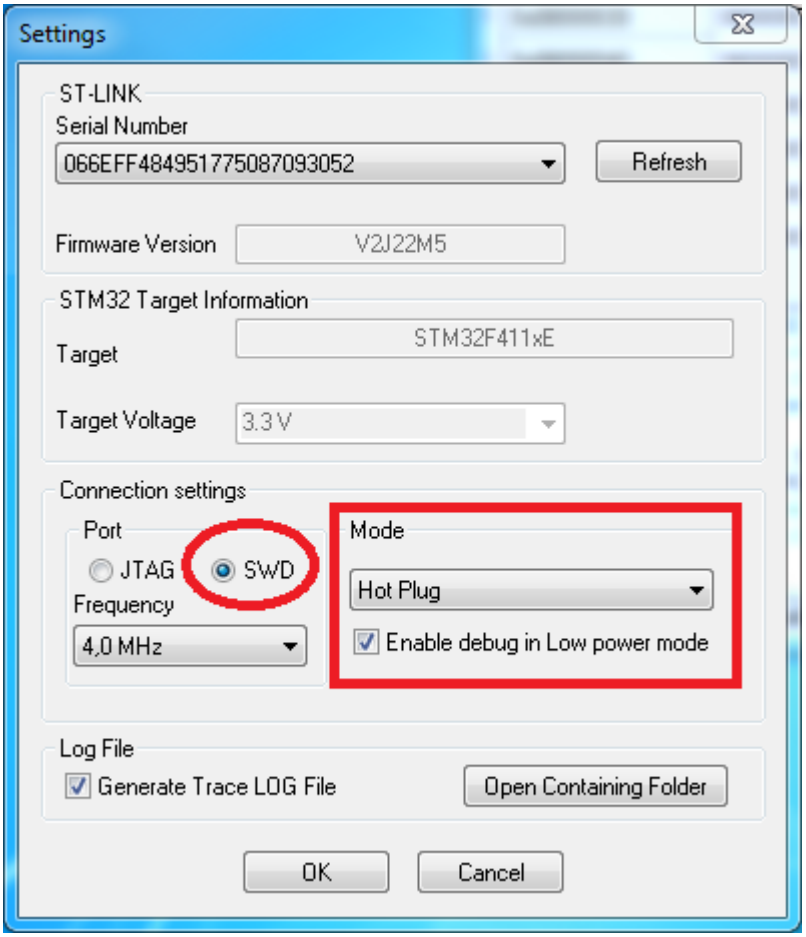
If you continue to only see a FAIL.TXT, check that your Nucleo really is the F411. If it isn't a Nucleo F411 you will have to use it as generic ST-LINK programmer with the STM ST-LINK utility as described in the next section.

2.5. Program the ELMO with the STM ST-LINK Utility

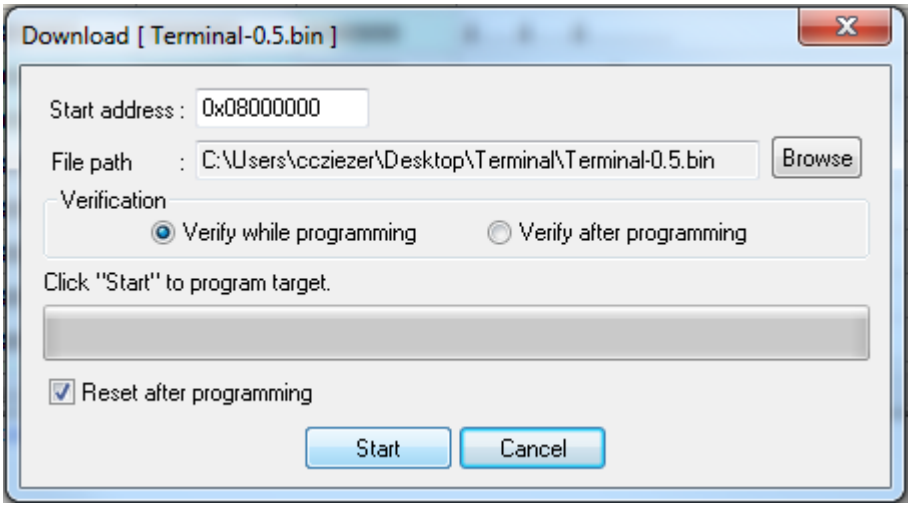
Once the ELMO board is powered and connected to the ST-LINK, connect the ST-LINK to the computer and execute the STM ST-LINK Utility software.



Go into Target->Settings and make sure that your ST-LINK is showing a serial number. Set the Port to SWD and the Mode to Hot Plug and press OK.



Select the binary to program via File->Open File and select from Target->Program & Verify… (or press CTRL-P). Set the Start Address to 0x08000000 and click Start to start programming the ELMO.



The STM ST-LINK Utility should after a short while reply with a Verification…OK message.

The ELMO is now programmed and ready for use.