Using DFU File Manager and DFuSE to Re-Flash NanoM

May 17, 2016

Overview

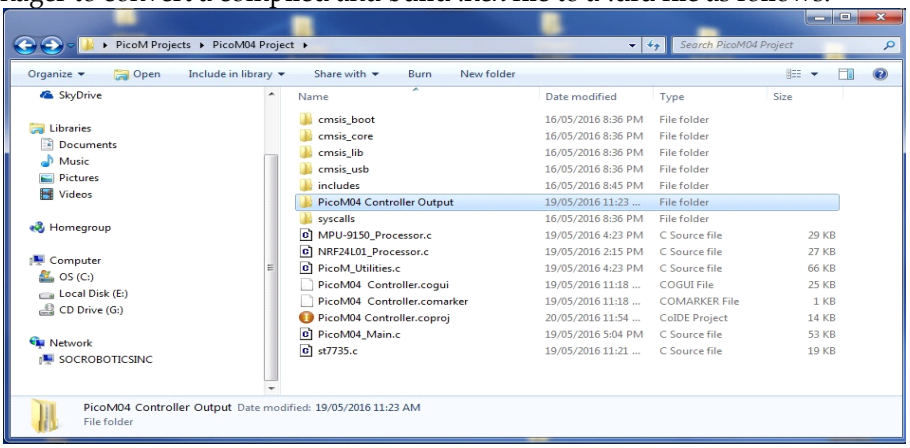
ST provides two applications that greatly simplify the process of re-flashing an ST processor such as the STM32F042, STM32F072, STM32F303 or STM32F405 using the chips built-in DFU bootloader and USB port. By using DFuSe no special hardware is required. Essentially re-flashing an ST STM32Fxxx processor is a two step process - first convert a .hex file to a .dfu file and then download the .dfu file to the target processor via the processors USB port.

All STM32Fxxx processors have a built-in DFU bootloader. All SOC Robotics boards that use STM32Fxxx parts have the ability to be re-programmed (re-flashed) using the DFU bootloader - the bootloader is activated either by installing a special jumper that pulls the BOOT0 pin (which is on all STM32Fxxx processors) high followed by repowering the board or by executing jmp to bootloader under software control. All SOC Robotics projects support a special ESCb command that forces the application to start the DFU bootloader (except for the STM32F042 which still requires the DFU jumper be installed). Once the new program is downloader the board must be power cycled with the DFU jumper removed to start the new program. Note that except for the STM32F042 the DFU jumper is not required when using the ESCb command.

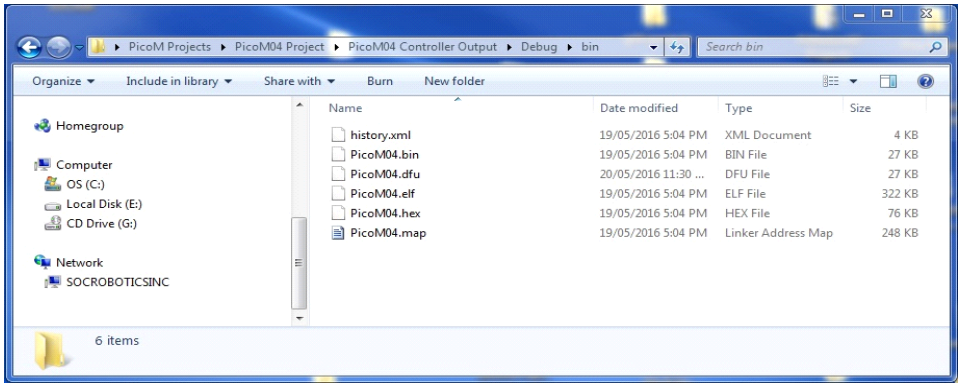
A step by step summary of how to use DFU File Manager and DFuSE is provided below. See the SOC STM32F Development Tool Installation Guide for instructions on how to install the ST software tools, ARM GNU Tool Chaing and CooCox IDE. This example uses the CooCox project for the PicoM04 Wireless Processing Node mounted on a IMU8420 10DOF Datalogger.

Converting .hex file to .dfu file

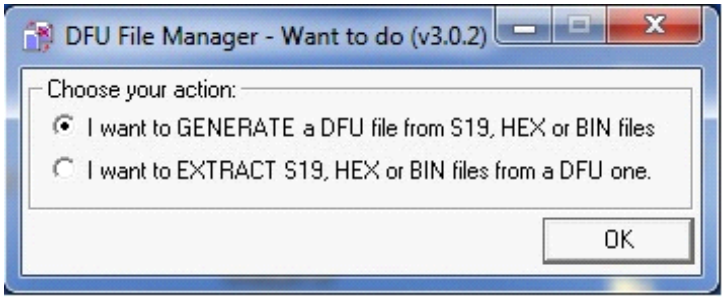
Use DFU File Manager to convert a compiled and build .hex file to a .dfu file as follows:



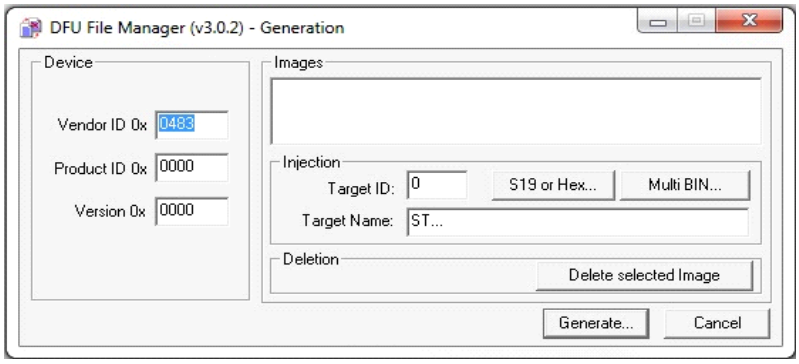
First navigate to the project file .hex file which is located several folders down in the PicoM04 Controller Output folder of the main PicoM04 Project folder.



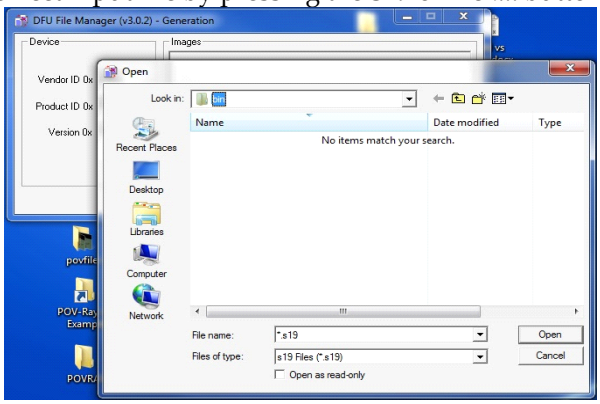
This folder contains .elf, .hex, .map and .bin files along with a previously created .dfu file.



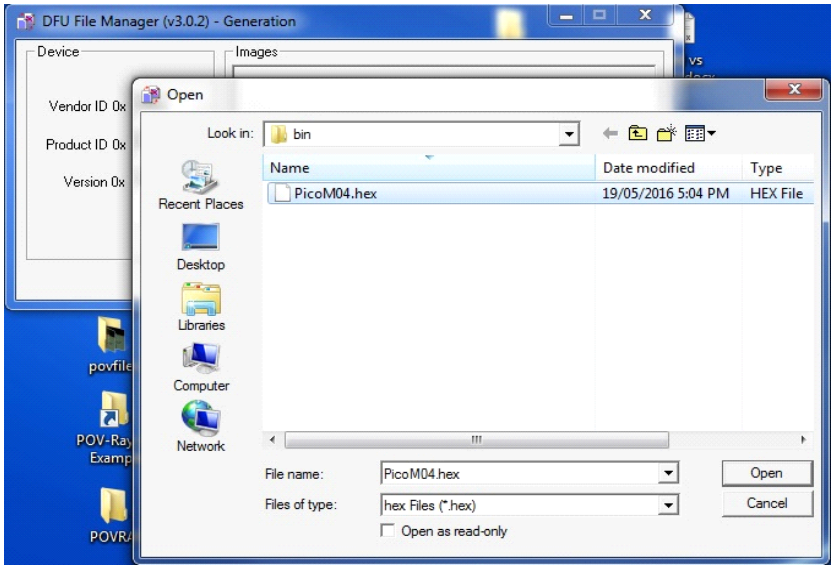
Launch the DFU File Manager application and press OK with GENERATE a DFU file selected.



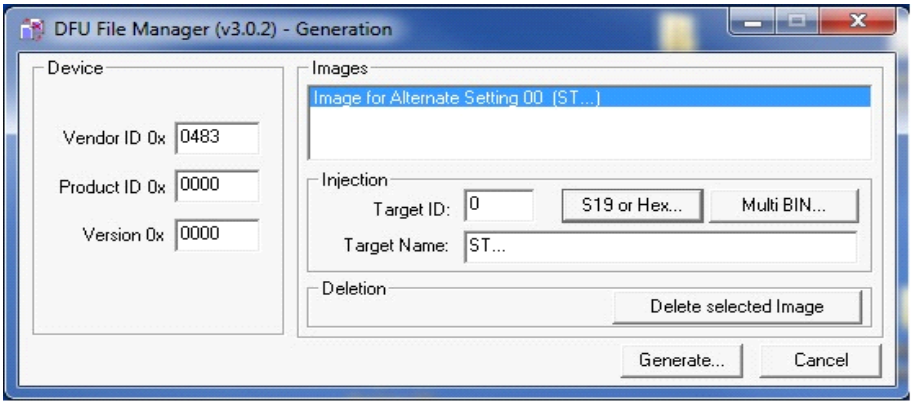
First step is to select in the correct input file by pressing the S19 or Hex... button.



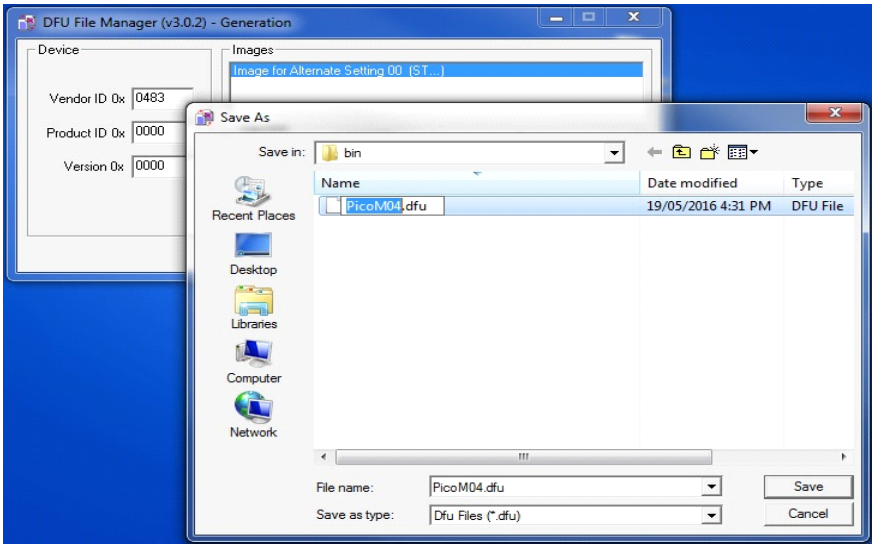
By default the file type s19 is displayed - switch this to .hex.



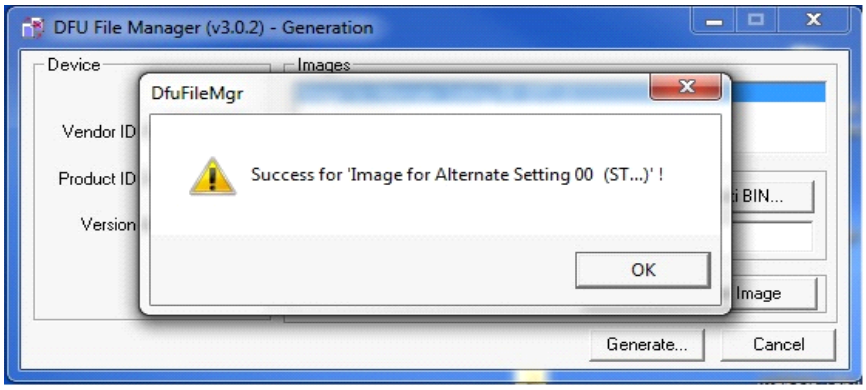
Select PicoM04.hex and select Open.



Now press the Generate button which opens the Save As dialog - enter the name of the target or if already created from a previous Generate select the .dfu file.



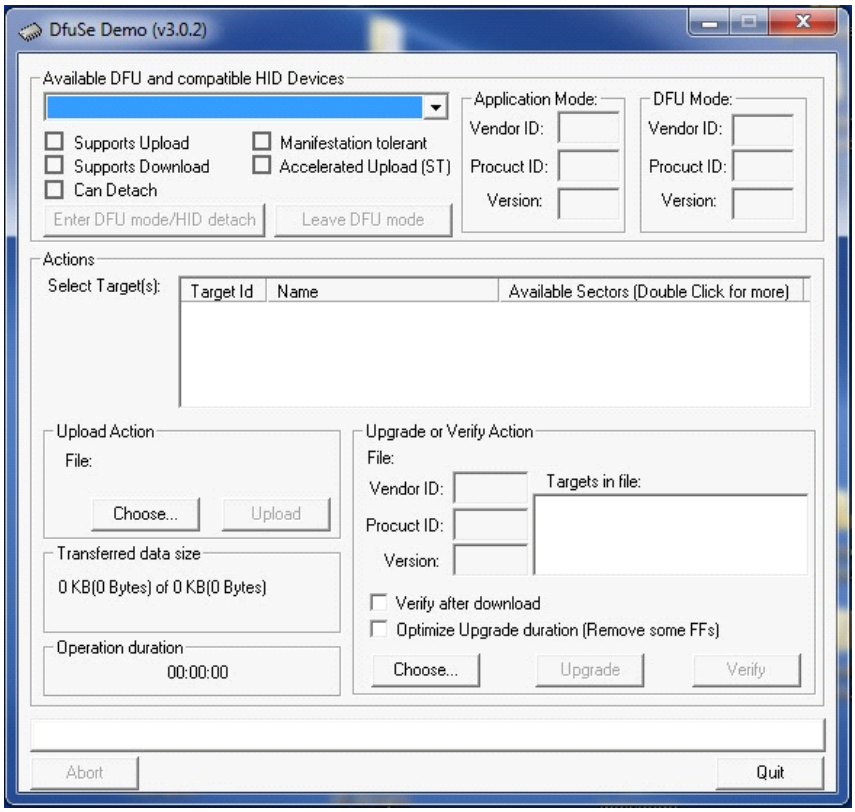
Press Save - this starts the conversion process.



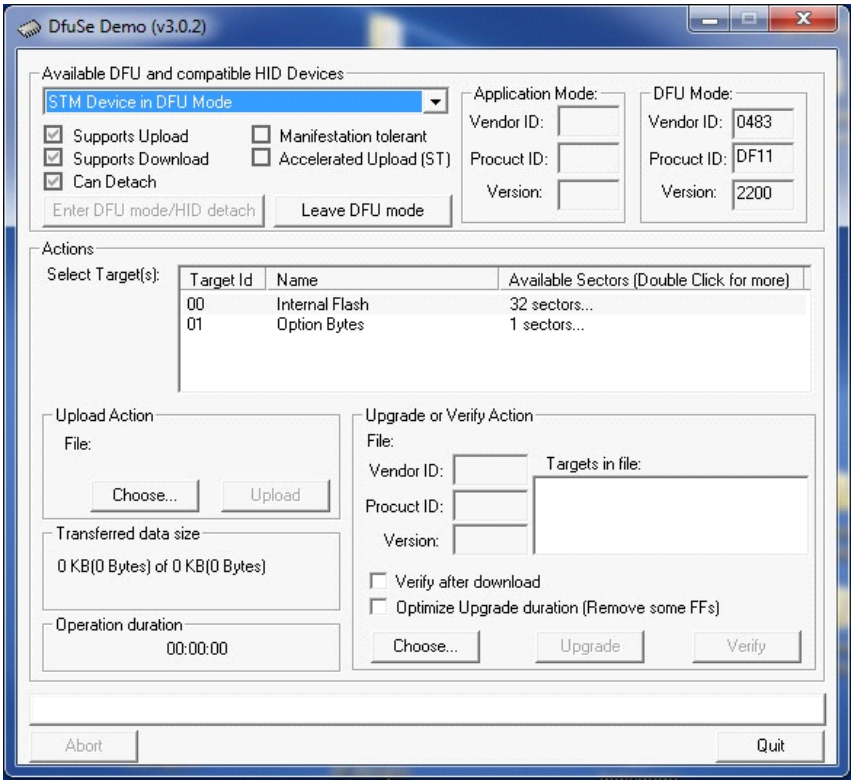
The DFU image file has been created - press OK to finish.

Programming the target board with DFuSe

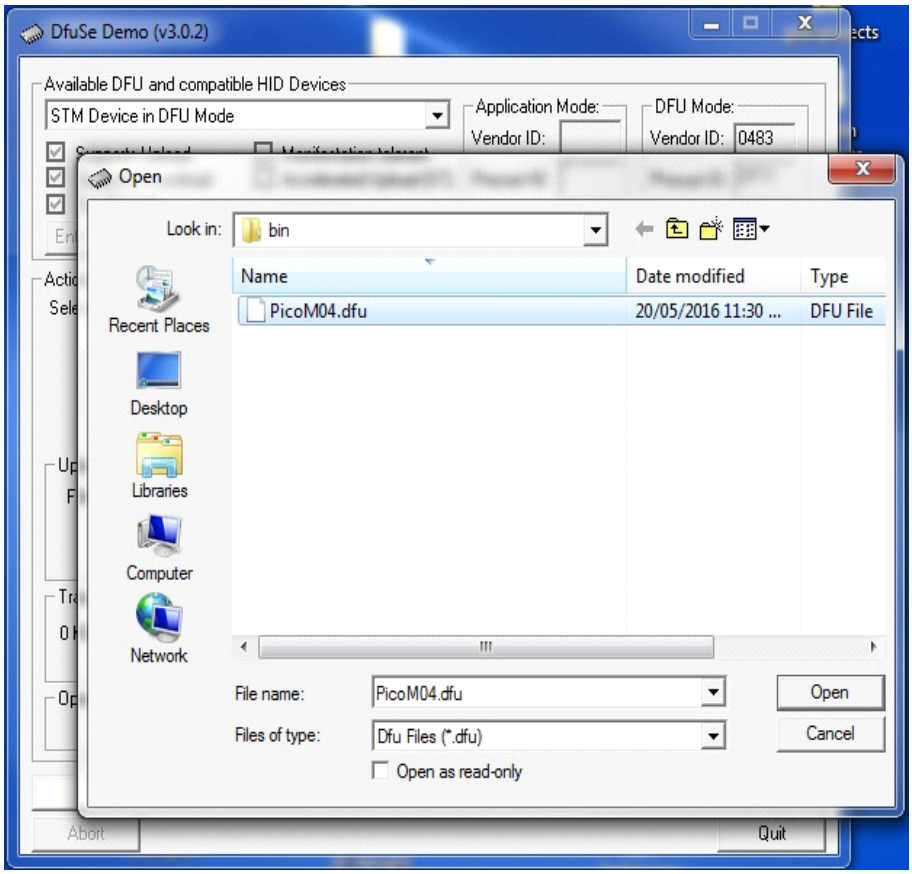
DFuSe is a DFU (Device Firmware Upgrade) programming utility (see SOC STM32F Development Tool Installation Guide for setup instructions. Start DFuSe.



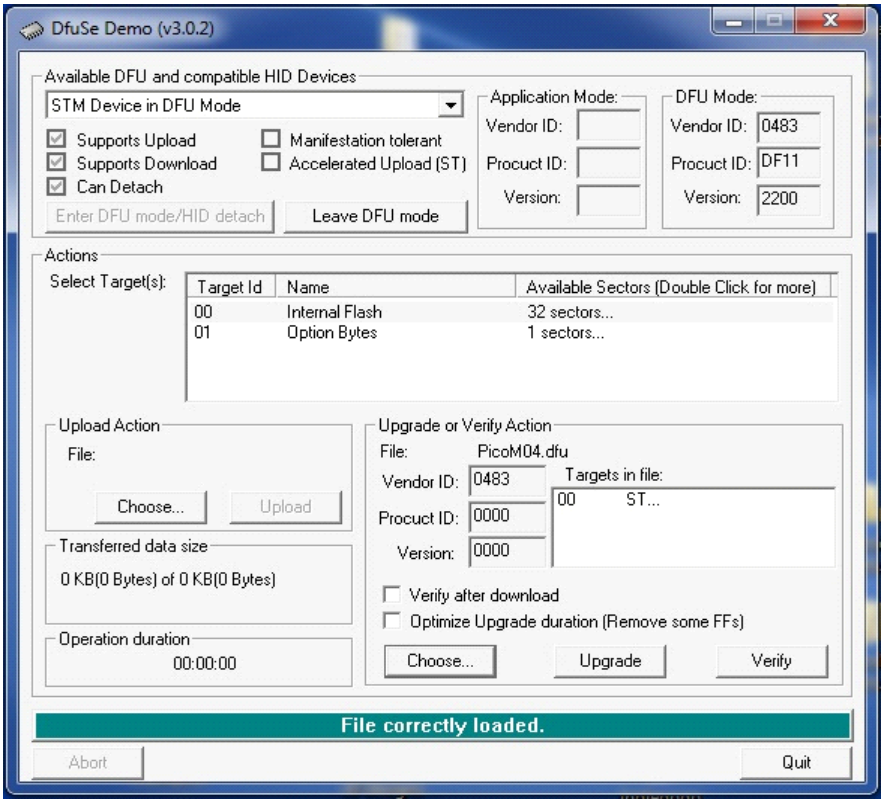
Install the DFU jumper on the PicoM04, connect a USB cable and re-power the board. If the DFU driver was installed correctly the following screen should show a STM Device found message as below.



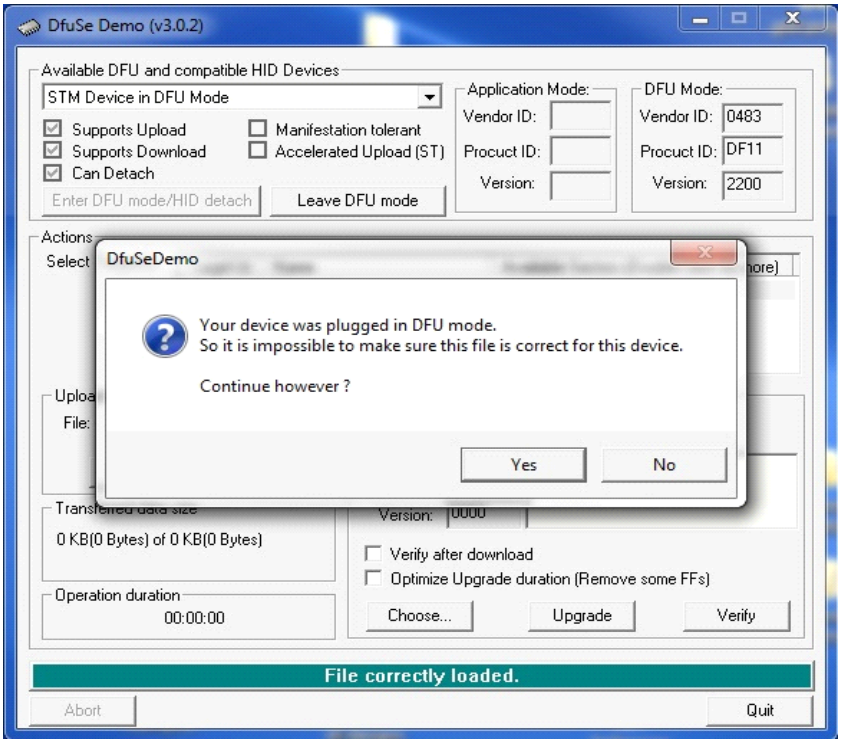
Press Choose and enter the name of the .dfu file which in this case is PicoM04.dfu. You may need to navigate to the bin project file where the dfu file created by DFU File Manager placed it.



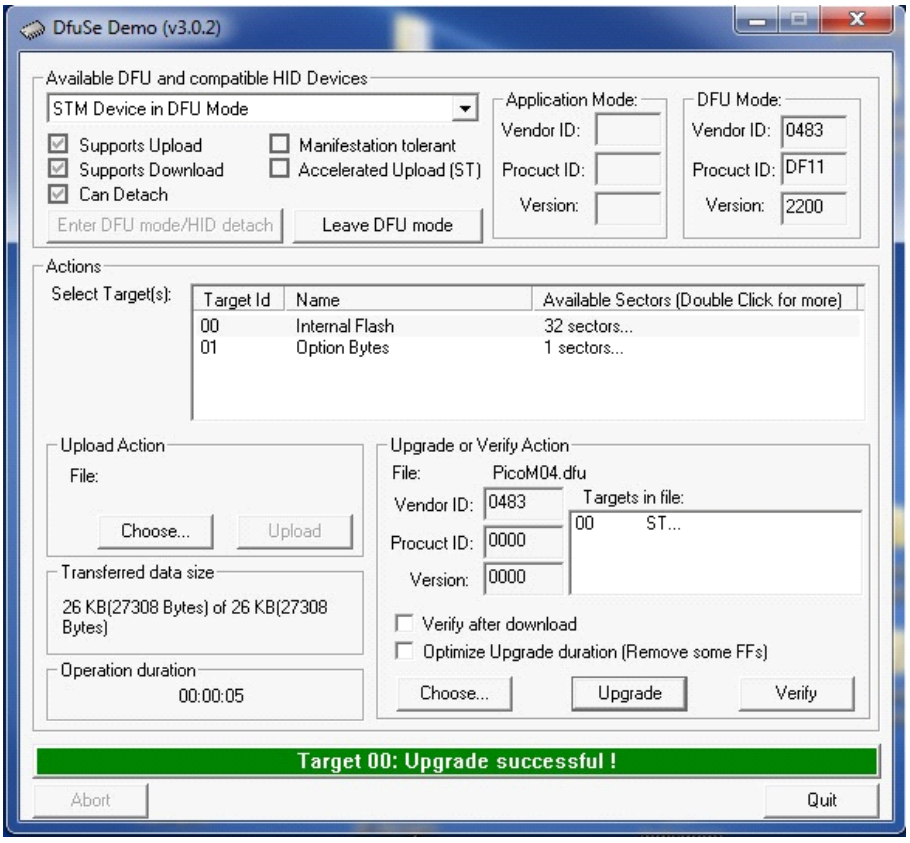
Hit Open to load the dfu file.



DFuSe now shows loaded file parameters - note the Vendor ID of the file matches the Vendor ID in the top right corner of the target.



Press upgrade and click the Yes button to continue.



The board is now loaded with the new program.

Starting the board

Once DFuSe finishes programming the board make sure the DFU jumper is not installed and cycle the power - the new application should now start.