

F150.FW: Firmware upgarde via DFU

This document describes the procedure on how to perform a firmware upgrade using DFU device capabilities

Table of Contents

- [Overview](#)
 - [References](#)
 - [Configure the project environment](#)
 - [Prerequisites](#)
 - [Project setup](#)
 - [Create a workspace](#)
 - [Import the projects](#)
 - [Scripts environment setup](#)
 - [Generate a release build](#)
 - [Setup bootloader for the first time](#)
 - [Upload a new DFU image to the device](#)
 - [If your device is already running an application image](#)
 - [Once the device is already in DFU mode \(LED blinking red fast\)](#)
-

Overview

The intent of this document is to:

- Setup the project environment to build the project
- Define how to build a DFU-compatible image
- Define how to re-program the device

References

- [F150.FW: Bootloader design](#)

Configure the project environment

Prerequisites

- Python >3.7
- STM32CubeProg

Project setup

Clone the repository into your PC.

Create a workspace

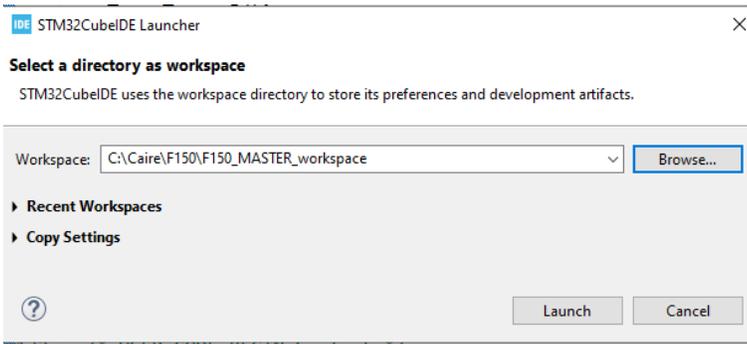
In a folder on the same level as the root of your working copy, name it "F150_MASTER_workspace".

IMPORTANT: the build script expects the Workspace folder to be at the same level as your repo folder.

It should look like:

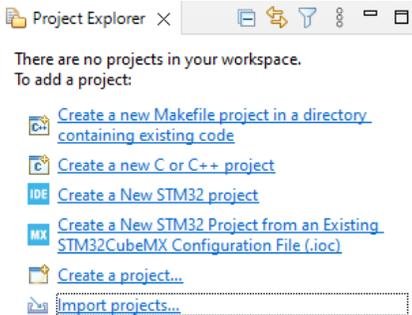
Name	^	Dat
F150_MASTER		11/
F150_MASTER_workspace		11/

Open the workspace from STM32CubeIDE. You can get a popup when you click to open the program, or you can change the workspace from the IDE: **File > Switch Workspace**

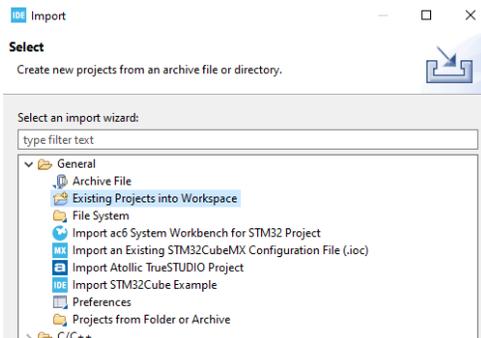


Import the projects

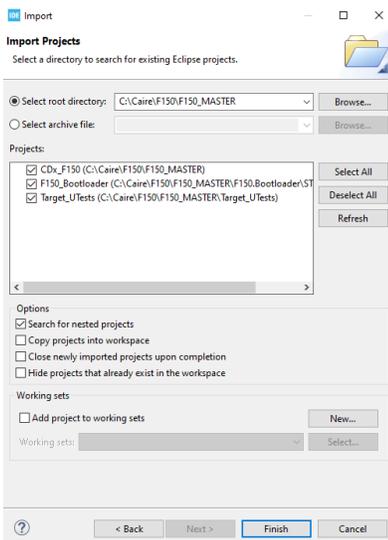
From the Project Explorer tab click on "Import Projects..."



General → Existing Projects into Workspace



Select your repo folder and click on finish:



Scripts environment setup

- In order to perform a project build and all scripts related to this process, we are using Python environments
- In order to set up this tool, you need to complete the following steps:
 1. Open a terminal on the repository root.
 2. Create and activate the virtual environment (requires python>=3.7).

```
python -m venv .venv
.venv/Scripts/activate.bat
```

3. Update pip:

```
python -m pip install --upgrade pip
```

4. Install requirements:

```
pip install -r requirements.txt
```

5. Close the terminal

Generate a release build

IMPORTANT: close STM32CubeIDE before running the build scripts.

- Each time you need to make a DFU-compatible image release, follow these steps:
 - Open a terminal on the repository root.
 - If you are using Power Shell

```
.\.venv\Scripts\Activate.ps1
```

- If you are using Terminal

```
.\.venv\Scripts\Activate.bat
```

- To build the code perform:

```
invoke build-stm32cubeide
```

- You will find the output files at <root>/Release_Package folder
 - All required binary files can be found at <root>/Release_Package folder/CaireDiagnostics-F150-Binary-v99.x/Binaries
 - The script to upgrade the device can be found at <root>/Release_Package folder/CaireDiagnostics-F150-Binary-v99.x/Tools

Setup bootloader for the first time

- In a blank device, you will need to upload the bootloader image available at <root>/Release_Package folder/CaireDiagnostics-F150-Binary-vMM.mm.bbbbBinaries
- Use the STM32CubeProg GUI and program the device with the bootloader image named **CaireDiagnostics-F150-Bootloader-vMM.mm.bbbb.hex**

Upload a new DFU image to the device

If your device is already running an application image

- This is the case when you have already upgraded your device previously via DFU upgrade
- You will need to send the START DFU command to the device via USB CDC connection
- Use your tools, or the provided demo application, to achieve this task.
- Once ready, the device will perform a warm restart and enter DFU mode

Once the device is already in DFU mode (LED blinking red fast)

- Once the device is in DFU mode, the device LED will start blinking red very fast.
- Open a terminal in the <root>/Release_Package folder/CaireDiagnostics-F150-Binary-v<desired version>/Tools folder and run:

```
python.exe .\UpgradeViaDFU.py
```

- After the upgrade is complete, you need to perform a power cycle of the device.