1. Installer Everything (<https://www.voidtools.com/>)
2. Installer STM32 Cube MX (<https://www.st.com/en/development-tools/stm32cubemx.html>)
3. Installer STM32 Cube IDE (<https://www.st.com/en/development-tools/stm32cubeide.html>)
4. Installer Visual Studio Code (<https://code.visualstudio.com/>)
5. Installer Python 3.8.9 (<https://www.python.org/downloads/>)
   1. Pendant l’installation, assurez-vous de cocher cette case :

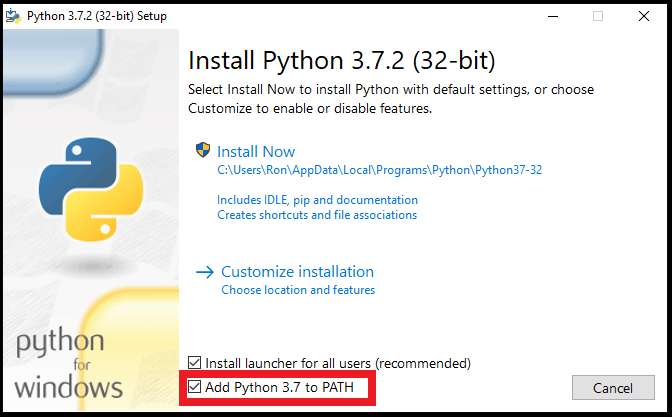
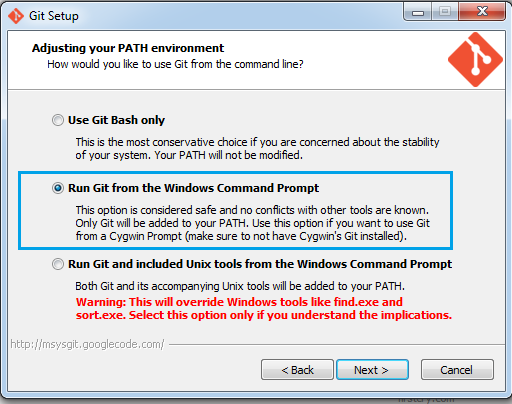
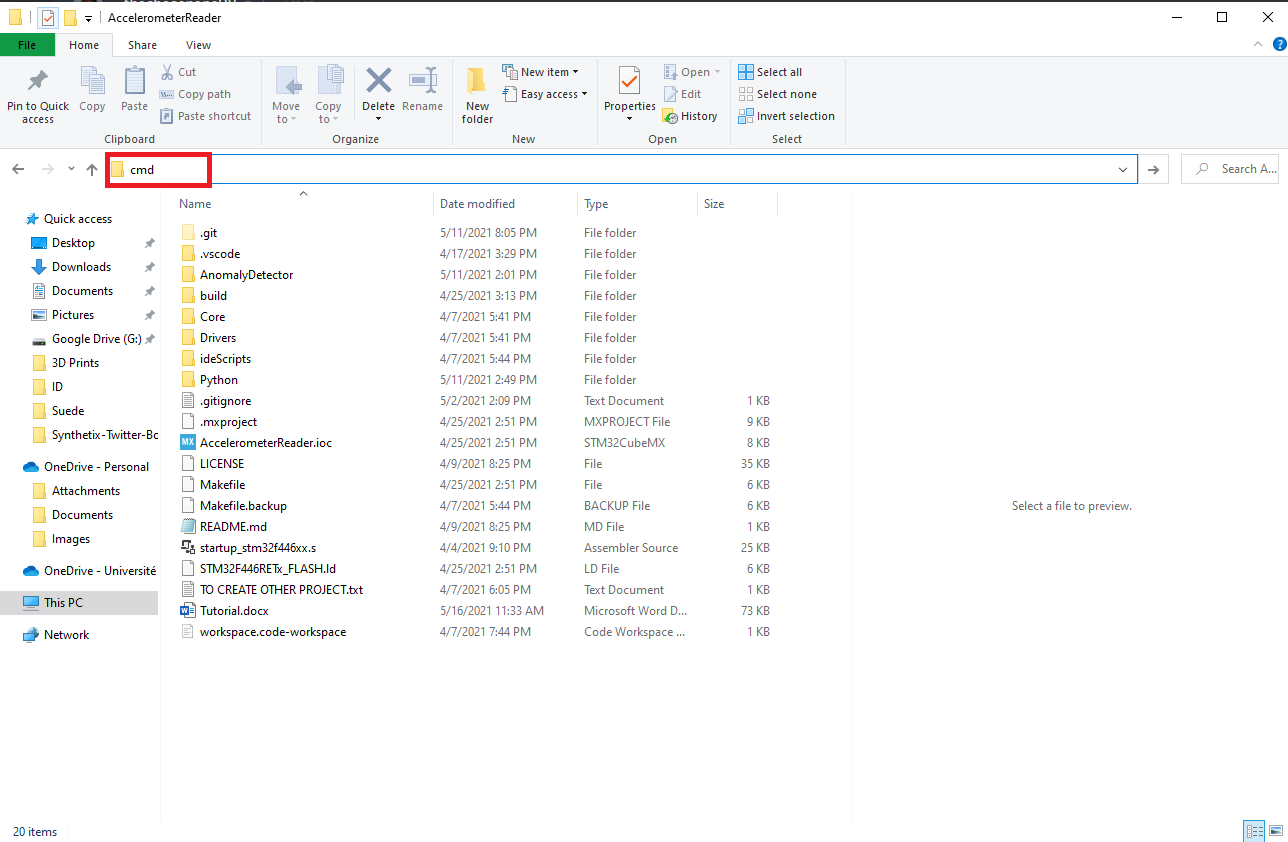
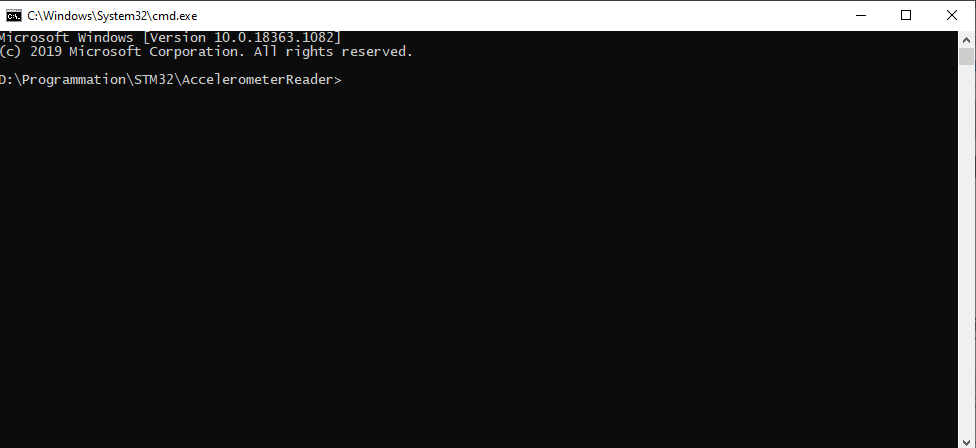


Figure 1: Ne porter pas attention à la version dans cette image

1. Installer Git (<https://git-scm.com/downloads>)
   1. Pendant l’installation, assurez-vous de cocher cette case :



1. Créer vous un dossier qui servira à contenir le projet. Si vous êtes sur Windows, assurez-vous qu’il n’y ai pas d’espace dans tout le chemin menant au dossier. Par exemple :
   1. D:\Programmation\STM32\AccelerometerReader
2. Ouvrez un CMD dans le répertoire que vous venez de créer
   1. Pour vous faciliter la vie, tapez cmd dans la barre en haut de Explorer comme suit :
   2. Ensuite, appuyer sur Enter. Cela vous ouvrira automatiquement le CMD avec comme référence le dossier dans lequel vous vous situez.



1. Clonez maintenant mon GitHub avec la commande suivante :
   1. git clone <https://github.com/thechosenone98/STM32-AI.git>
   2. Déplacer vous dans le répertoire nouvellement créer : cd STM32-AI
2. Créer un environnement virtuel Python et installer les librairies requises avec les commandes suivante :
   1. Commençons par installer virtualenv pour nous permettre de créer un environnement virtuel : pip install virtualenv
   2. Ensuite on crée l’environnement virtuel :

virtualenv –python=PATH\_TO\_PYTHON.EXE PATH\_TO\_VENV\_LOCATION

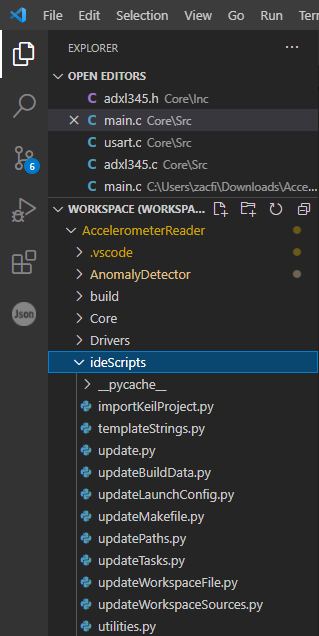
Exemple : virtualenv --python=C:\Python38\python.exe D:\Programmation\STM32\TutoTest\STM32-AI\venv

* 1. Activer l’environnement avec : PATH\_TO\_VENV\_LOCATION\Scripts\Activate

Exemple : D:\Programmation\STM32\TutoTest\STM32-AI\venv\Scripts\activate

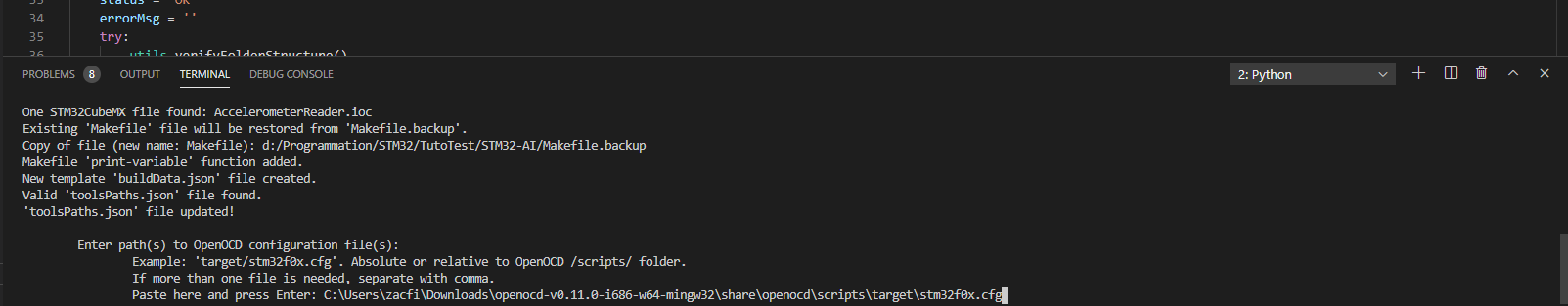
* 1. Installer toutes les librairies requises avec : pip install -r requirement.txt

1. Ouvrez Visual Studio Code.
   1. Faite File->Open Folder…->Choisissez le fichier STM32-AI
   2. Ensuite, faite File->Save Workspace As… et sauvegarder ce fichier directement dans le dossier SMT32-AI (le nom n’est pas important, tant que l’extension reste .workspace)
   3. Une fois que c’est fait, ouvrez dans l’éditeur le fichier ideScripts :



* 1. Ouvrez update.py
  2. Faite Ctrl+Shift+P puis taper python interpréter (sélectionner Python : Select Interpreter)
  3. Sélectionner STM32-AI puis Enter Interpreter Path et enfin Find…
  4. Naviguer dans venv puis dans Script et sélectionner python.exe
  5. Faite maintenant un clic droit sur update.py et choisissez Run Python File in Terminal

1. Maintenant il va falloir que nous pointions certain fichier important (cette étape est CRUCIALE) :
   1. À l’aide d’Everything chercher le fichier suivant stm32f0x.cfg et copier son chemin en faisant un clic droit dessus puis Copy Full Name to Clipboard. Coller ensuite ce chemin dans l’invite de commande de Visual Studio Code :



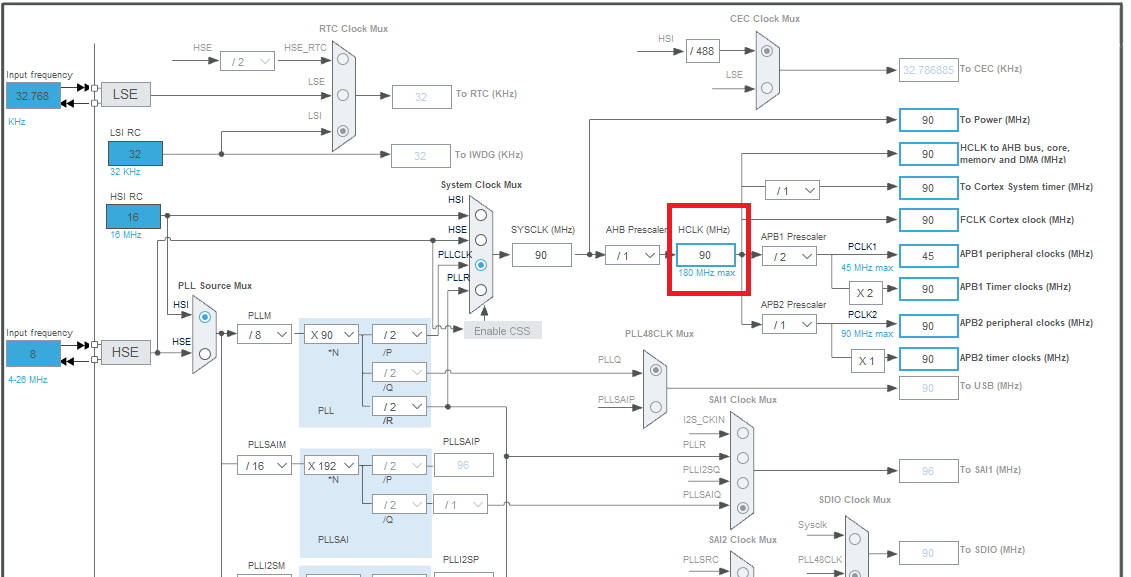
Appuyer sur Entrée

Note : Il est important d’encadrer tout bout du chemin qui contient des espace (par exemple :

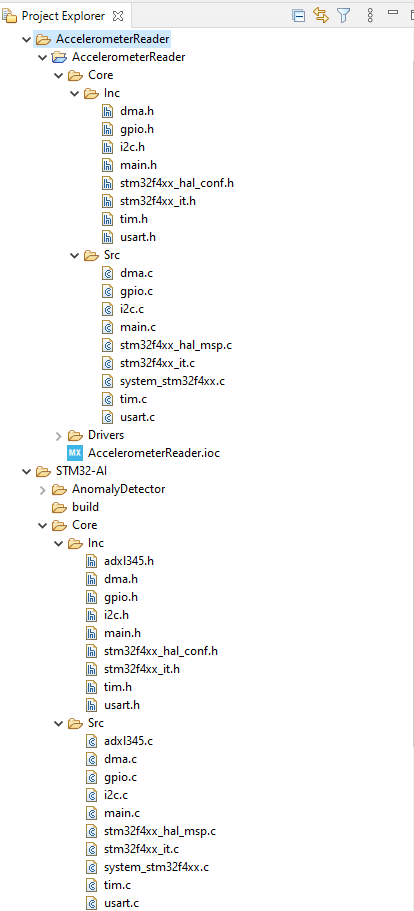
D:\Software\"STM32 Cube IDE"\STM32CubeIDE\_1.6.0\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.productdb.debug\_1.6.0.202102151443\resources\cmsis\STMicroelectronics\_CMSIS\_SVD\STM32F446.svd)

* 1. Refaite la même chose pour la liste de mot suivant :
     1. stm32f446.svd

1. Ouvrir STM32 Cube MX.
2. Cliquer sur Access MCU Selector.
3. Aller dans l’onglet Board Selector.
4. Rechercher NUCLEO-F446RE, sélectionnez-le et cliquez sur Start Project.
5. Cliquer sur Yes pour initialiser toutes les pins avec leur mode par défaut pour ce board de développement.
6. Ouvrez l’onglet Connectivity puis cliquer sur I2C1
7. Dans Mode :
   1. I2C -> I2C
8. Dans Configuration :
   1. Speed Mode -> Fast Mode
   2. Primary Slave Address -> 0x1D
9. Toujours dans Connectivity, cliquez sur USART2
10. Dans Configuration :
    1. Aller dans l’onglet DMA Settings puis cliquez sur Add
    2. Dans DMA Request, sélectionnez USART2\_RX
    3. Dans DMA Request Settings :
       1. Mode -> Circular
11. Cliquer maintenant sur la catégorie Timers et cliquez sur TIM10
12. Sous TIM10 Mode and Configuration, cochez la case Activated
13. Dans Configuration :
    1. Prescaler -> 90-1
    2. Counter Period -> 1000-1
14. Aller maintenant dans l’onglet Clock Configuration (situer en haut complètement)
15. Écrivez 90 dans cette case puis faite Entrée :



1. Aller maintenant dans l’onglet Project Manager
   1. Project Name -> AccelerometerReader
   2. Project Location ->Sélectionner le dossier ou vous avez créer votre venv de Python.
   3. Toolchain/IDE -> STM32CubeIDE
2. Aller dans l’onglet Code Generator située sur la gauche
3. Sous Generated files, cocher la case Generate Peripheral Initialization…
4. Vous pouvez aussi cocher Set all free pins as analog (sous HAL Settings)
5. Cliquer maintenant sur GENERATE CODE en haut à droite
6. Ouvrez STM32 Cube IDE manuellement
7. Faite File->New->STM32 Project From an Existing STM32Cube MX Configuration File
8. Aller sélectionner le fichier PROJECT\_NAME.ioc situé dans le dossier du projet que vous venez de créer.
9. Cliquez sur Finish.
10. À l’aide de l’explorateur de ficher de Windows aller dans le dossier STM32-AI/Core/Src
11. Copier tous les fichiers directement dans l’ide dans le dossier PROJECT\_NAME/Core/Src
12. Faite oui pour remplacer les fichiers déjà existants.
13. Refaite la même chose avec les fichiers située dans Core/Inc.
14. Faite Project->Build Project



1. Copier-coller les fichier suivant (de STM32-AI vers notre projet) tout en respectant les dossiers dans lesquels ils vont :
   1. adxl345.h
   2. adxl345.c
   3. main.c