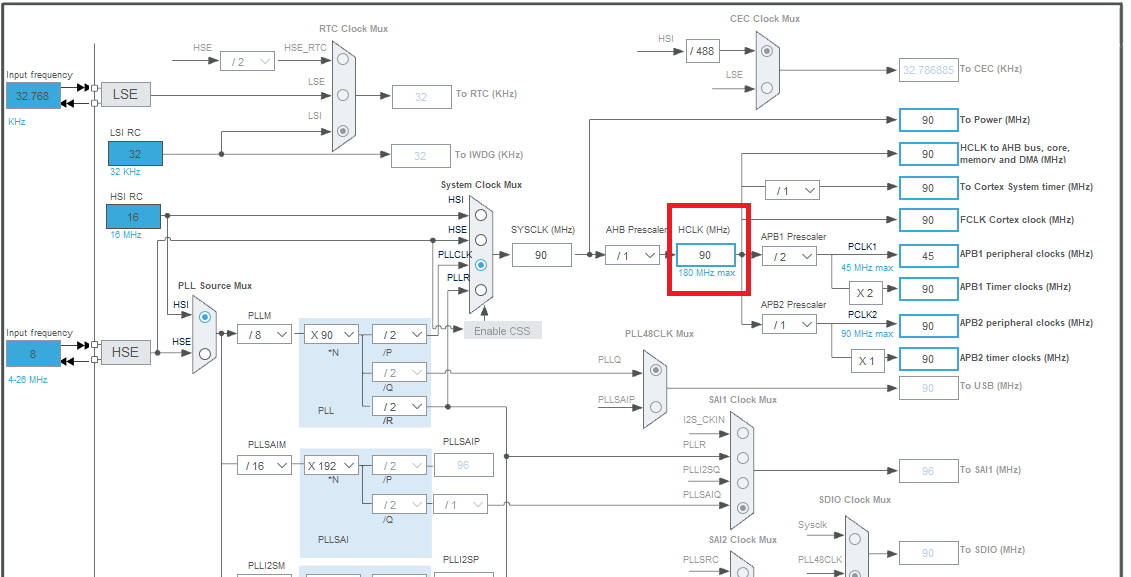
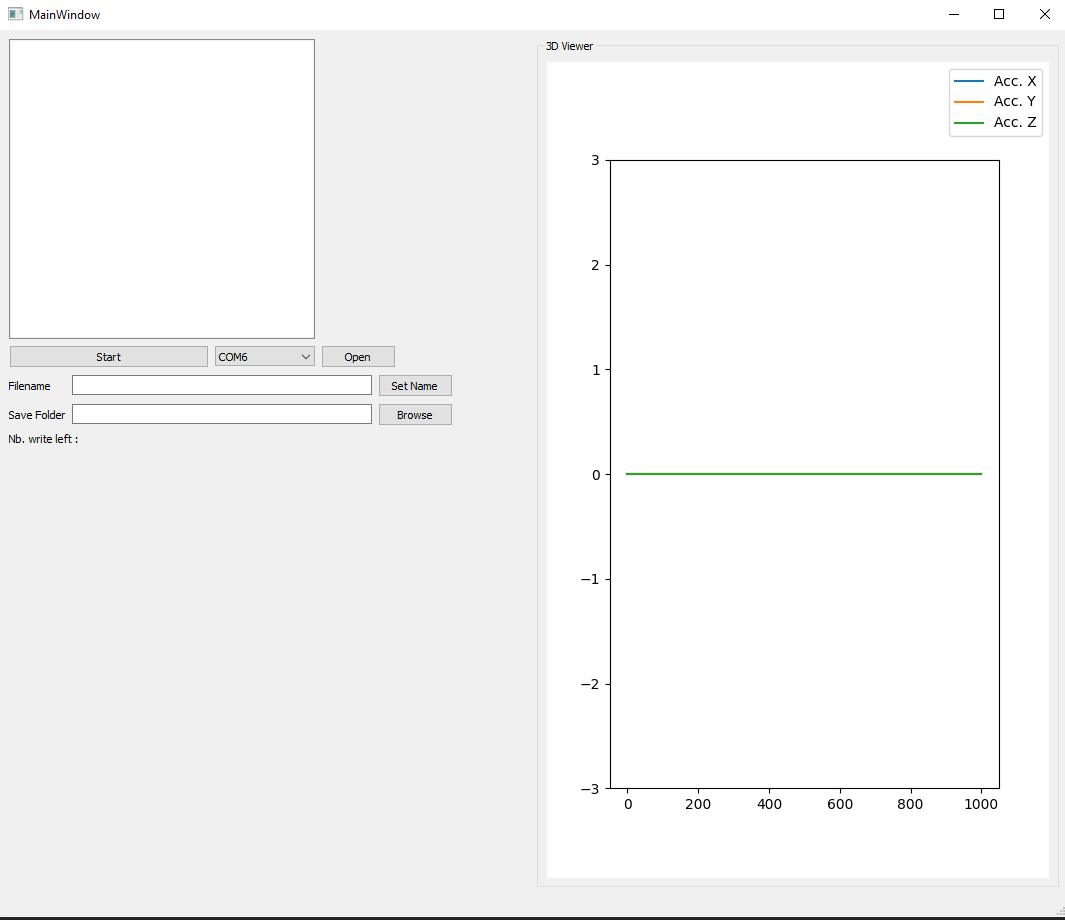
# Créer le projet pour accumuler des données de l’accéléromètre

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 créé 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.
15. Brancher le STM32.
16. Faite Run->Run As->STM32 Application…
17. Ceci va programmer le board et quitter à la fin (avec une erreur disant que la connexion à été coupé mais ce n’est pas grave).
18. Retourner dans l’explorateur de Windows et rendez-vous dans le dossier STM32-AI/Python et ouvrez une invite de commande à cet endroit.
19. Activer votre environnement comme mentionné plus haut.
20. Faite la commande python main.py et une fenêtre va s’ouvrir.



1. Cette fenêtre nous permet d’enregistrer les données provenant de l’accéléromètre. Voici comment l’utiliser pour enregistrer nos différents modes d’opération (assurez-vous de tout lire avant de commencer) :
   1. Entrée le nom du mode d’opération que vous voulez enregistrer dans « Filename »
   2. Choisissez le dossier où vous voulez enregistrer vos différentes données. Pour ne pas à avoir à faire de modification plus tard, je vous recommande de sauvegarder vos données dans le dossier /STM32-AI/Python/AccelerometerData et d’écraser les fichiers qui se situe à l’intérieur de celui-ci.
   3. Assurez-vous d’avoir le bon port de sélectionner pour communiquer avec le STM32
   4. Cliquer sur Open puis sur Start
   5. La collecte de données commence automatiquement et la quantité de données accumulé s’affiche à coté de File Size
   6. Lorsque vous avez accumulé suffisamment de données pour ce mode d’opération (je recommande 10Mb) cliquer sur Close pour arrêter la capture de données et libéré le fichier.
   7. Répéter toutes les étapes pour les autres modes d’opération. Vous devez au final sauvegarder 4 modes d’opération avec les noms de fichier suivant Off.bin, Normal.bin, Error\_1.bin et Error\_2.bin).