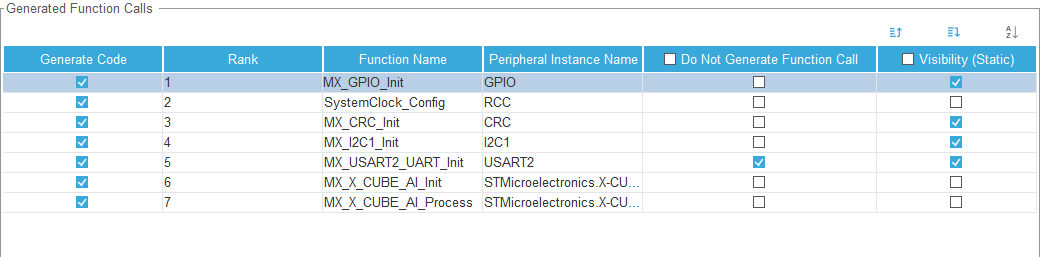
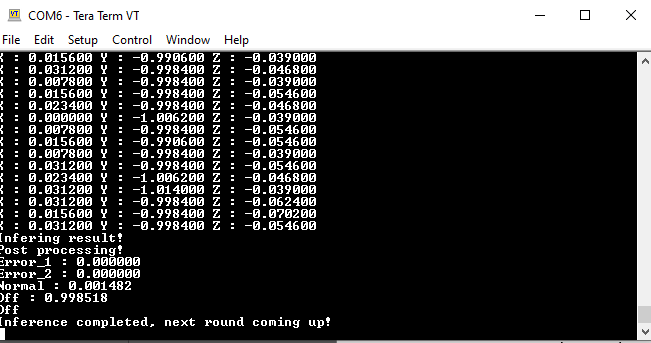
# Faire fonctionner notre modèle sur le Nucleo64

1. Ouvrez STM32 Cube MX pour que l’on commence à créer le projet qui servira à utiliser le modèle que nous venons de créer en temps réel pour déterminer le mode d’opération du système à l’aide des données du gyroscope.
2. Commencer par créer un nouveau projet en faisant File->New Project… et en sélectionnant ensuite le board Nucleo-F446RE dans l’onglet Board Selector puis en cliquant sur Start Project en haut à droite (faite Yes pour initialiser les périphériques à leurs modes par défaut).
3. Commencer par aller dans Connectivity, puis cliquez sur I2C1. Activer le en choisissant I2C->I2C puis, sélectionner l’option I2C Speed Mode->Fast Mode. Finalement entrer l’adresse 0x1D dans Primary Slave Address.
4. Assurez-vous que le Baud Rate du USART2 est à 115200. Cette fois-ci nous n’avons pas besoin de mettre d’autre option comme dans le projet précédent puisque le board ne recevra pas de données en provenance de l’ordinateur.
5. Cliquez maintenant sur Software Packs->Manage Software Packs. Allerz ensuite dans l’onglet STMicroelectronics et ouvrez X-CUBE AI. Installer le module Artificial Intelligence. Une fois que c’est fait fermer la fenêtre.
6. Cliquez maintenant sur Software Packs->Select Components. Ouvrez l’onglet STMicroelectronics.X-CUBE-AI ainsi que les deux sous-onglets. Cochez Core et sélectionner Application->ApplicationTemplate. Faite OK.
7. Vous avez désormais un nouvel onglet Software Packs qui se situe dans Catégories (le même endroit où on retrouve Connectivity et Timers. Ouvrez-le et sélectionner le module STMicroelectronics.X-CUBE-AI. Cliquez sur Add Network. Entrez le nom suivant anomalydetector (il est important d’utiliser ce nom puisque l’on va copier des fichiers qui se base sur ce nom plus tard). Cliquez sur Browse… qui se situe à coté de Model. Aller chercher votre fichier MODEL\_NAME.h5 et sélectionner le. Cliquez sur Analyse (une fois que l’analyse est terminée vous pouvez regarder quelques informations dans cette fenêtre, mais je vous conseille d’aller voir la structure de notre modèle en fermant la fenêtre et en cliquant sur Show Graph). Cliquez maintenant sur Validate on Desktop (ceci va faire rouler notre modèle sur l’ordinateur pour s’assurer que la librairie X-CUBE AI a été en mesure de bien convertir notre modèle TensorFlow vers un code en C qui fonctionne).
8. Aller maintenant dans Clock Configuration et entrer 180 dans la case HCLK (le même cas que pour le projet précédant). Ce projet va roulez avec une fréquence 2 fois plus vite simplement pour permettre au modèle d’AI de rouler à une vitesse convenable (ce n’est pas réellement nécessaire dans le cas de cet exemple, mais vous devez le faire puisque nous allons copier des fichiers plus tard qui se base sur ce chiffre).
9. Dirigez-vous maintenant dans l’onglet Project Manager. Le nom du projet importe peu mais je vous conseille AnomalyDetector. Sélectionner Toolchain->STM32CubeIDE. Augmenter le Minimum Heap Size à 0x2000 (c’est simplement pour augmenter la quantité de mémoire dynamique disponible au programme).
10. Allez dans Code Generator et cochez la case Generate peripheral initialization…
11. Allez dans Advanced Settings et cochez les cases pour obtenir le résultat suivant :



1. Cliquez sur GENERATE CODE. Une fois terminer, fermer STM32 Cube MX.
2. Ouvrez STM32 Cube IDE. Faite File->New->STM32 Project From an Existing STM32CubeMx…Aller cherchez le fichier PROJECT\_NAME.ioc que vous venez de générer. Cliquez ensuite sur Finish.
3. Comme nous l’avons fait plus tôt, nous allons maintenant copier les fichier ce situant dans STM32-AI/AnomalyDetector/Core/Src et STM32-AI/AnomalyDetector/Core/Inc vers les dossier /Src et /Inc de notre projet (faite Yes pour écraser nos fichiers). Ensuite, Copier les fichiers situés dans STM32-AI/AnomalyDetector/X-CUBE-AI/App dans le dossier X-CUBE-AI/App de notre projet.
4. Par défaut, noter projet ne peut pas écrire de nombre à décimal (float) alors nous devons aller activer ce paramètre dans le « linker ». Faite un clic droit sur notre projet puis cliquer sur Properties. Allerz ensuite dans C/C++ Build->Settings. Sous l’onglet Tool Settings cliquer sur MCU Settings et cochez la case « Use float with printf… ». Faite Apply and Close.
5. Si vous avez encore l’ancien projet d’ouvert (deux projets se situant dans l’explorateur de STM32CubeIDE), je vous conseille de fermer le premier en faisant un clic droit sur le nom tout en haut de sa hiérarchie en en cliquant sur Close Project (il va rester là mais il ne sera plus actif).
6. Branchez votre board dans l’ordinateur si ce n’est pas déjà fait.
7. Ouvrez main.c situé dans Core et faite Yes pour changer de perspective. Explorer un peu si vous le désirez (le code dans le mode n’est pas très impressionnant puisqu’il ne fait qu’appeler des fonctions plus complexes situé un peu partout dans les autre fichier). Si vous vous sentez explorateur, utiliser le raccourcis Ctrl+Clic Gauche sur une fonction pour être amené directement à sa définition).
8. Faite Project->Build Project.
9. Cliquez maintenant sur Run->Run As->STM32 Cortex-M…Faite OK sur la fenêtre qui va apparaître.
10. Une fois terminez l’application devrait être en marche sur le board.
11. Pour voir le résultat il nous faut désormais ouvrir Tera Term. Une fois celui-ci ouvert, cliquer sur Serial et sélectionner le port du board. Faite OK. Cliquer maintenant sur Setup->Terminal…->New Line->Receive->AUTO puis faite OK. Cliquez ensuite sur Setup->Serial port… puis changer le baudrate à 115200 puis faite New Setting.
12. Vous devriez désormais vour quelque chose de similaire à ceci :



1. Félicitation vous venez de faire fonctionner votre modèle d’AI sur un microcontrôleur 😊 Les données qui sont afficher corresponde aux 25 dernières mesures du gyroscope (elle sont toute utilisé pour former notre image de 25x3x1), suivi de la probabilité de chacune des catégories selon l’AI et enfin sa prédiction (dans le cas-ci haut c’est « Off » avec 99.85% de probabilité).
2. Essayer de reproduire les mêmes conditions que lorsque vous avez accumuler des données et que vous les avez catégorisés en Off, Normal, Error\_1 et Error\_2. Le modèle devrait être capable de prédire de façon assez précise ce qui se produit dans la réalité.