

Documentation Projet STM32 NanoEdgeAI

Introduction

Ce document décrit le processus de développement et de portage d'un projet utilisant la STM32-L152RE avec NanoEdgeAI pour la détection des points cardinaux via un module magnétique. Le projet initial, conçu pour la STM32-L476RG, a été adapté pour fonctionner sur la nouvelle plateforme.

Objectifs

- Créer un datalogger pour collecter des données magnétiques.
- Développer un système de classification des points cardinaux en utilisant NanoEdgeAI.
- Porter le projet existant de la STM32-L476RG vers la STM32-L152RE.

Matériel et Logiciels Utilisés

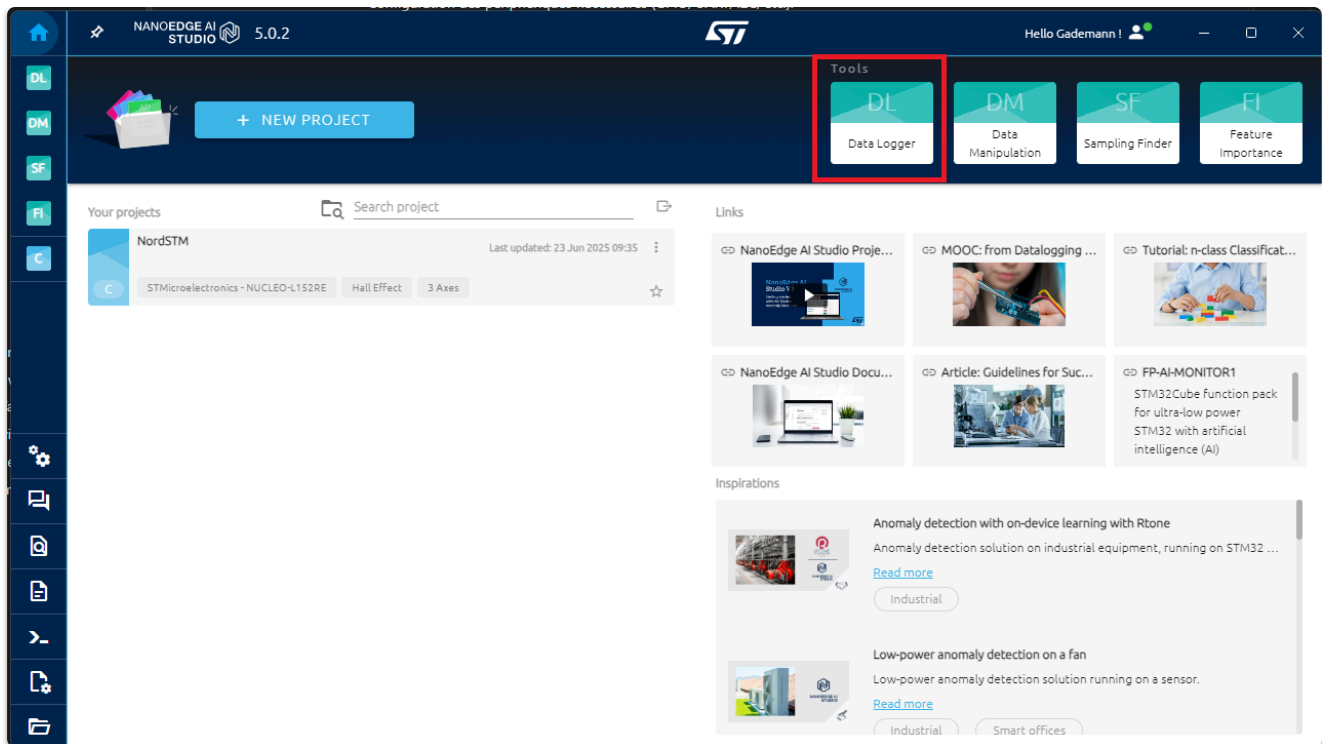
- **Matériel :**
 - STM32-L152RE
 - Module magnétique piloté par la librairie LIS2MDL
 - STM32-L476RG (pour référence)
- **Logiciels :**
 - STM32CubeIDE
 - NanoEdgeAI Studio
 - STM32CubeMX

Étapes du Développement

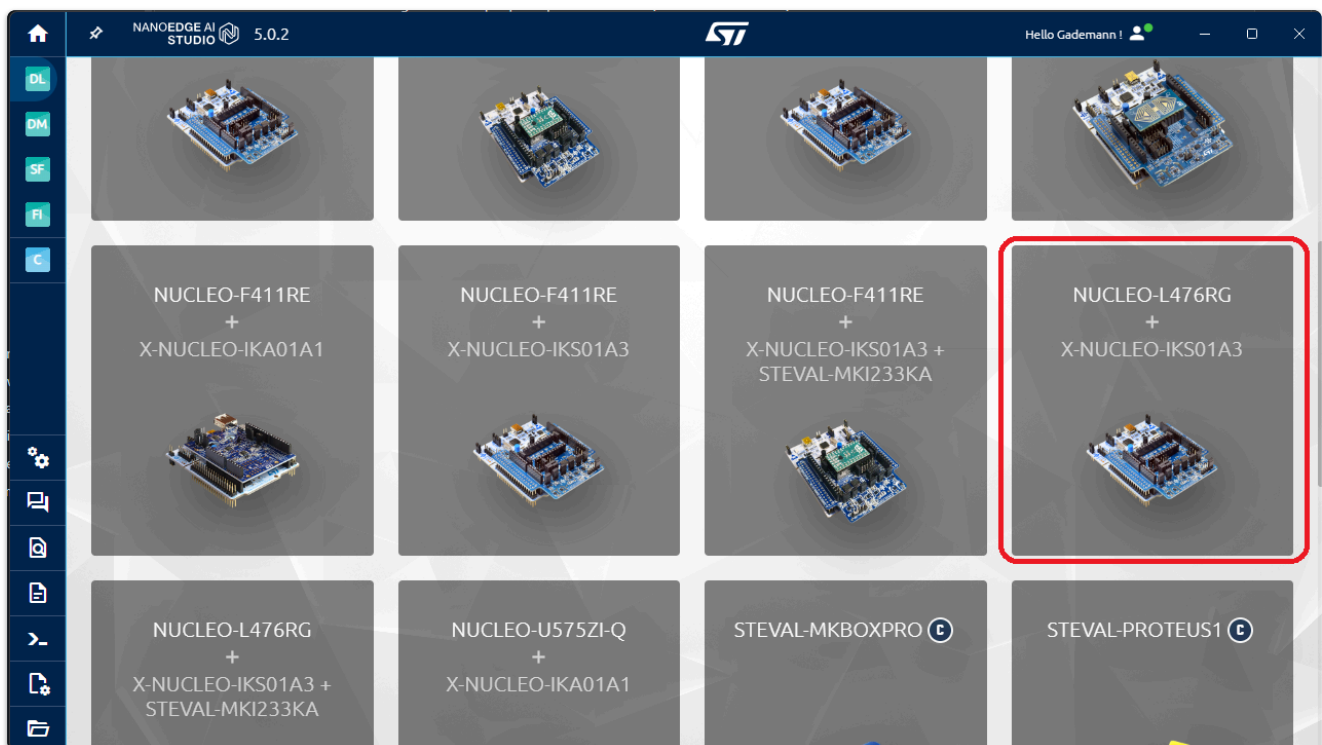
1.Création du dataset avec NanoEdgeAI :

1.1: Configuration du datalogger

Créer un outil datalogger en cliquant sur "Datalogger":



Choisir le modèle NUCLEO-L476RG + X-NUCLEO-IKS01A3 bien que vous ayez une NUCLEO-L152RE



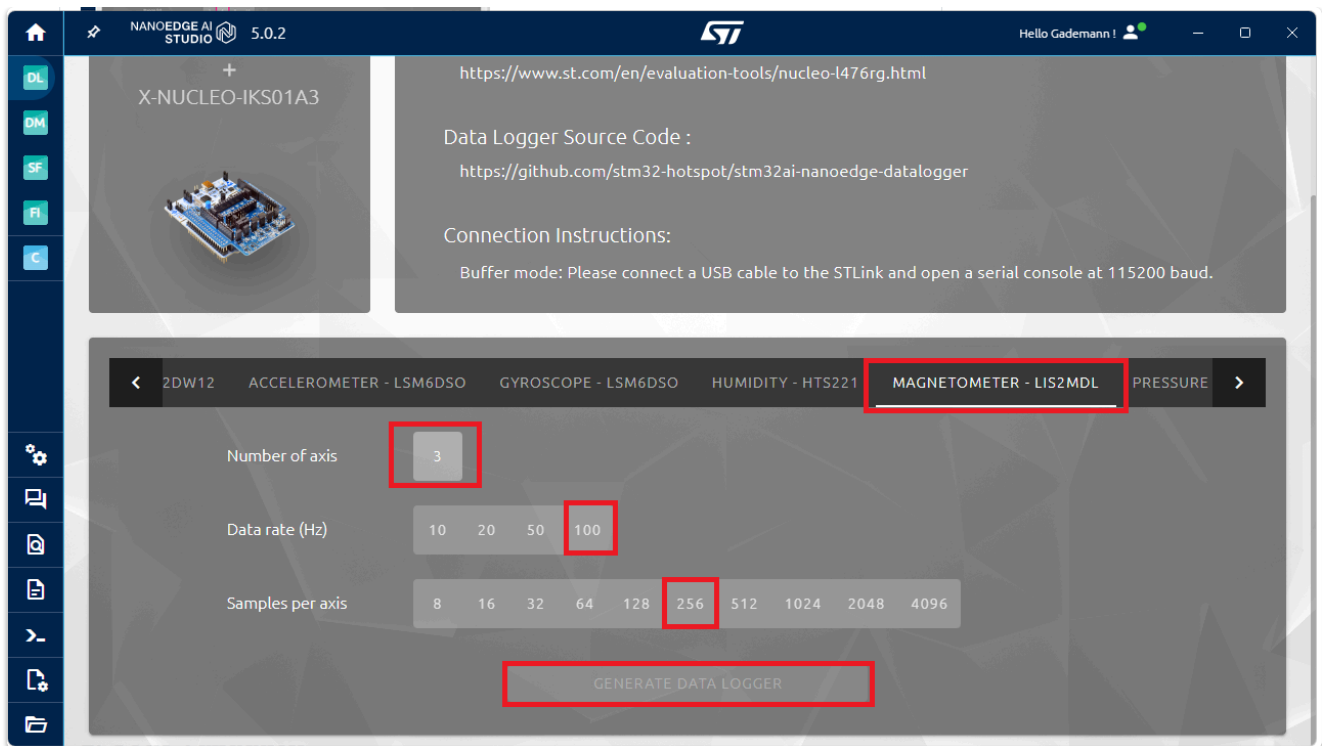
Sélectionner le magnétomètre en cliquant sur MAGNETOMETER-LIS2MDL avec la configuration suivante:

Nombre d'axes: 3 axes

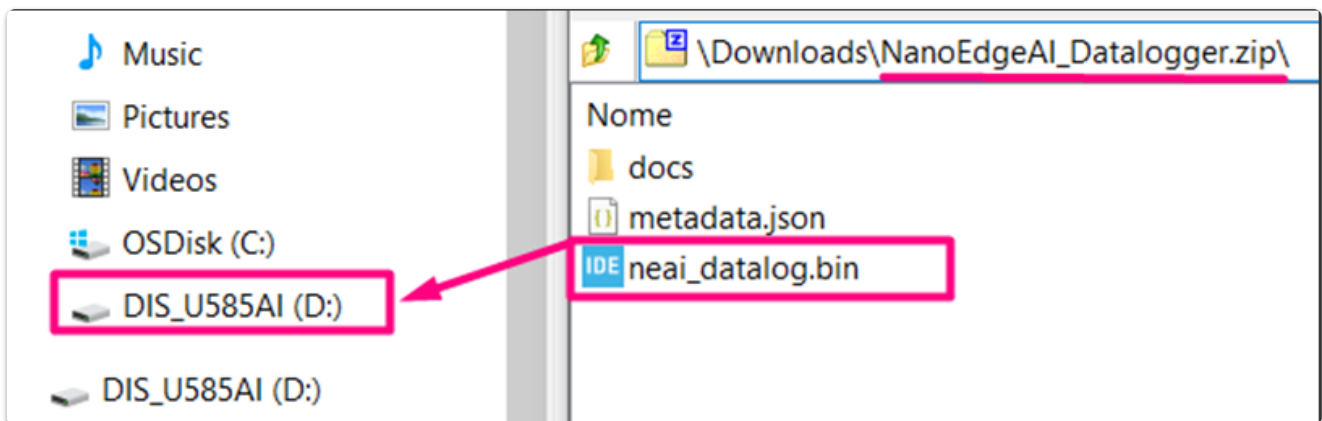
Fréquence: 100hz

Echantillons par axes: 256

Cliquer sur GENERATE DATA LOGGER



Téléverser le datalogger sur la carte (branchée au préalable)

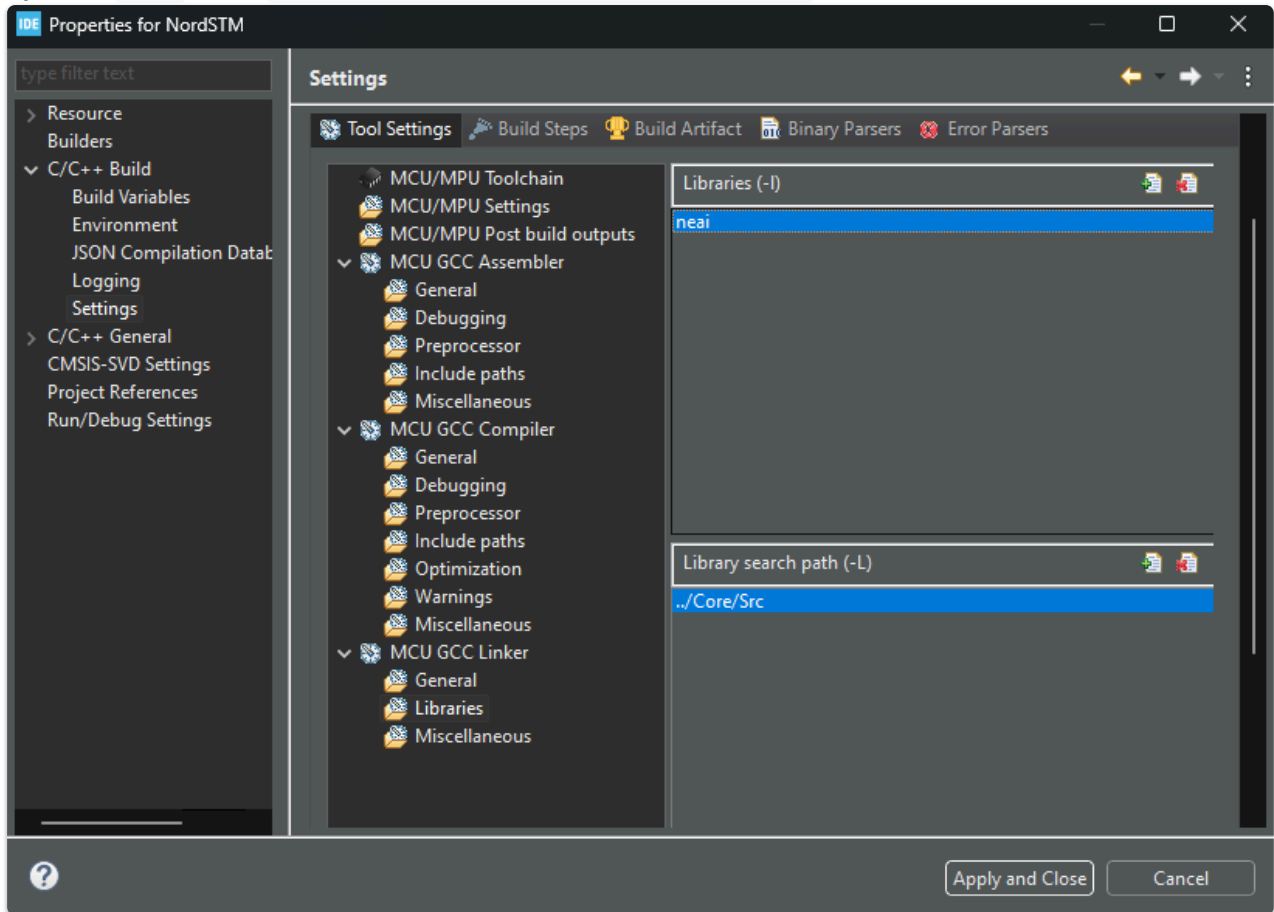


- Création des classes
TODO

2. Création du Projet Initial

- Configuration du projet :
 - Création d'un nouveau projet dans STM32CubeIDE pour la STM32-L152RE.
 - Configuration des périphériques nécessaires (GPIO, UART, I2C, etc.).
- Source du projet existant :
 - Le projet existant se trouve à l'emplacement suivant : `C:\Users\--votre--utilisateur--\STM32Cube\Repository\Packs\STMicroelectronics\X-CUBE-MEMS1\11.1.0\Drivers\BSP\Components\lis2mdl`.
- Création d'un projet parallèle :
 - Création d'un nouveau projet pour la STM32-L152RE.
 - Copie des fonctions du projet existant (STM32-L476RG) vers le nouveau projet (STM32-L152RE).

- Ajout de flag à la compilation
- Clique-droit sur le projet Properties
- Ajouter neai dans C/C++ Build/MCU GCC Linker/Libraries



- Vérification de la compilation :
 - Compilation du projet pour s'assurer qu'il n'y a pas d'erreurs.
 - Résolution des éventuels problèmes de compilation.

3. Intégration des Datasets NanoEdgeAI

- Ajout des datasets :
 - Intégration des datasets générés par NanoEdgeAI dans le projet STM32-L152RE.
 - 1 - ajouter le fichier libneia.a dans le dossier du projet STM32-L152RE: Core/Src
 - 2 - ajouter le fichier knowledge.h dans le dossier du projet STM32-L152RE: Core/Inc
 - 3 - ajouter le fichier NanoEdgeAI.h dans le dossier du projet STM32-L152RE : Core/Inc
 - Configuration des datasets pour la détection des classes

Voici la fonction principale:

```
#if NEAI_MODE

neai_state = neai_classification(neai_buffer, class_output_buffer, &id_class);

printf("Class: %s. NEAI classification return: %d.\r\n", id2class[id_class],
neai_state);

#else
```

```

for (uint16_t i = 0; i < AXIS * SAMPLES; i++) {

    printf("%.3f ", neai_buffer[i]);

}

printf("\n");

#endif

/* Reset drdy_counter in order to get a new buffer */

drdy_counter = 0;

/* Clean neai buffer */

memset(neai_buffer, 0x00, AXIS * SAMPLES * sizeof(float));

/* Set device in continuous mode */

lis2mdl_data_rate_set(&dev_ctx, MAGNETOMETER_ODR);

lis2mdl_operating_mode_set(&dev_ctx, LIS2MDL_CONTINUOUS_MODE);

HAL_Delay(20);

```

4. Développement de la Classification des Points Cardinaux

- **Inspiration du cours :**
 - Utilisation des concepts appris en cours pour la détection des classes avec le gyroscope: `neai_classification()`
 - Adaptation de ces concepts pour le capteur magnétique.
- **Utilisation de la librairie du capteur magnétique :**
 - Intégration de la librairie du capteur magnétique LIS2MDL.
 - Développement des fonctions de classification des points cardinaux.

Tests et Validation

- **Tests unitaires :**
 - Vérification du bon fonctionnement de chaque classe.
 - Vérification du bon fonctionnnement de l'afficheur 7 segments.
 - Vérification du bon fonctionnement du buzzer.
 - Vérification du bon fonctionnement du moteur pour la vibration.
- **Tests d'intégration :**

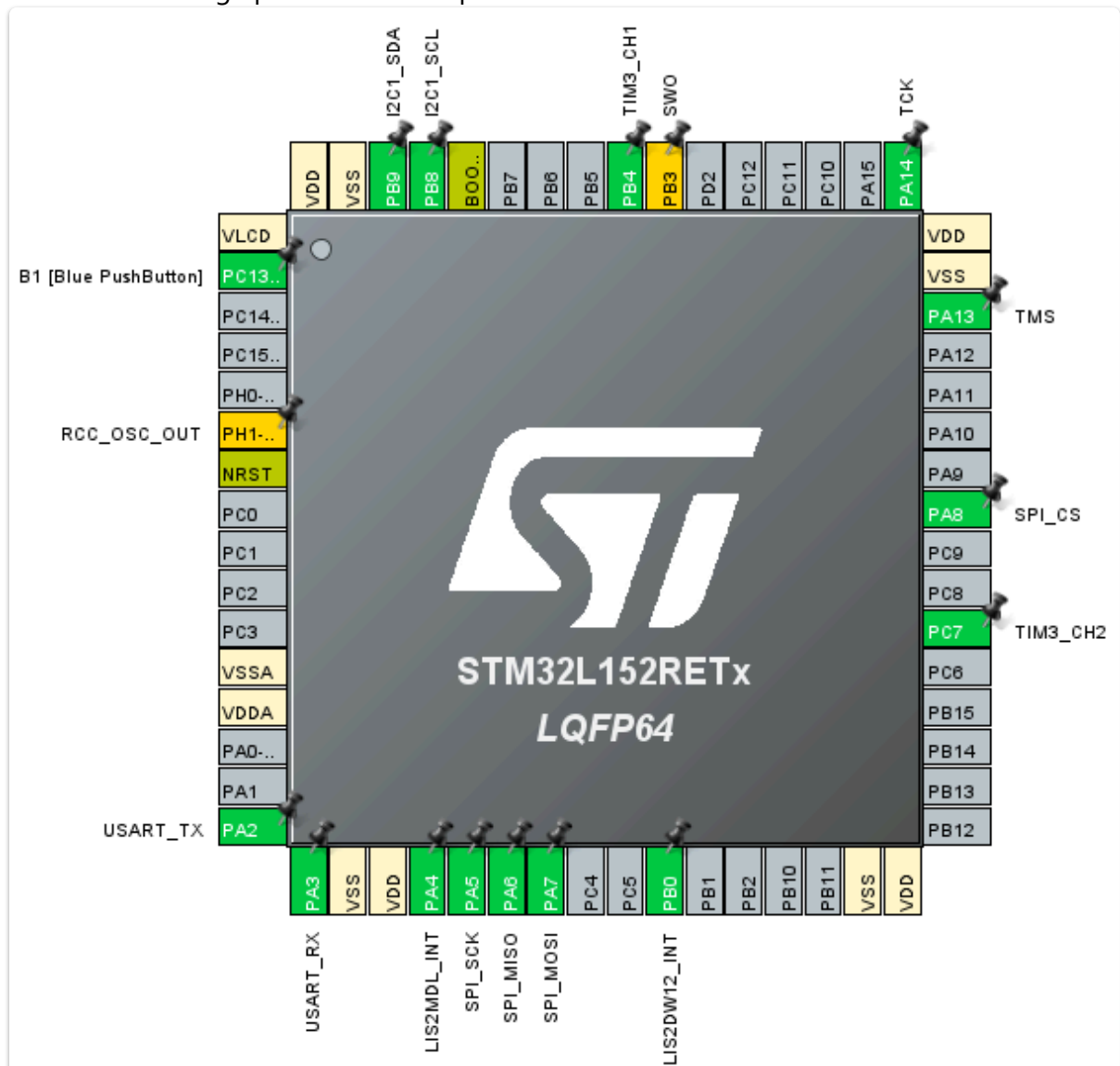
- Vérification du bon fonctionnement de l'ensemble du système.
- Validation des performances de l'IA embarquée pour la classification des points cardinaux.
- Validation de l'affichage du point cardinal en fonction de la classe détectée.
- Validation de la vibration si point cardinal nord.
- Validation du buzzer quand il y a trop de perturbation magnétique.

Résultats

- **Performances :**
 - Comparaison des performances du module magnétique sur les deux plateformes.
 - Évaluation de l'impact de l'IA embarquée sur les performances globales du système.

Annexes

- **Schéma de câblage :**
 - Schéma de câblage pour la nouvelle plateforme.



- **Code source :**
 - <https://github.com/psycotickyl/NordSTM>
- **Références :**
 - Liste des références et des ressources utilisées pour ce projet.

- Datalogger: <https://github.com/stm32-hotspot/stm32ai-nanoedge-datalogger>
- Projet avec capteur magnétique : <https://github.com/stm32-hotspot/stm32ai-nanoedge-datalogger/tree/main/Projects/NUCLEO-L476RG>