

Notice d'utilisation

Prototype d'analyse de la marche sans fil



ESIÉE

PARIS

GOURDON Mathéo
MARIANO Arthur
TRIGUI Ianis

E5FE
2020 - 2021

Table des matières

Introduction	3
Matériel nécessaire	3
Comment utiliser le dispositif	4
Téléversement des programmes Arduino	5
Utilisation de l'interface Python	6

Introduction

Ce document a pour objectif d'être une notice d'utilisation du projet Analyse de la marche réalisé dans le cadre du projet fil rouge à l'ESIEE Paris. Tous les fichiers nécessaires sont disponibles sur notre github (<https://github.com/arthurmio/Walking-Analysis-Project>).

Notre document est séparé en deux grandes parties :

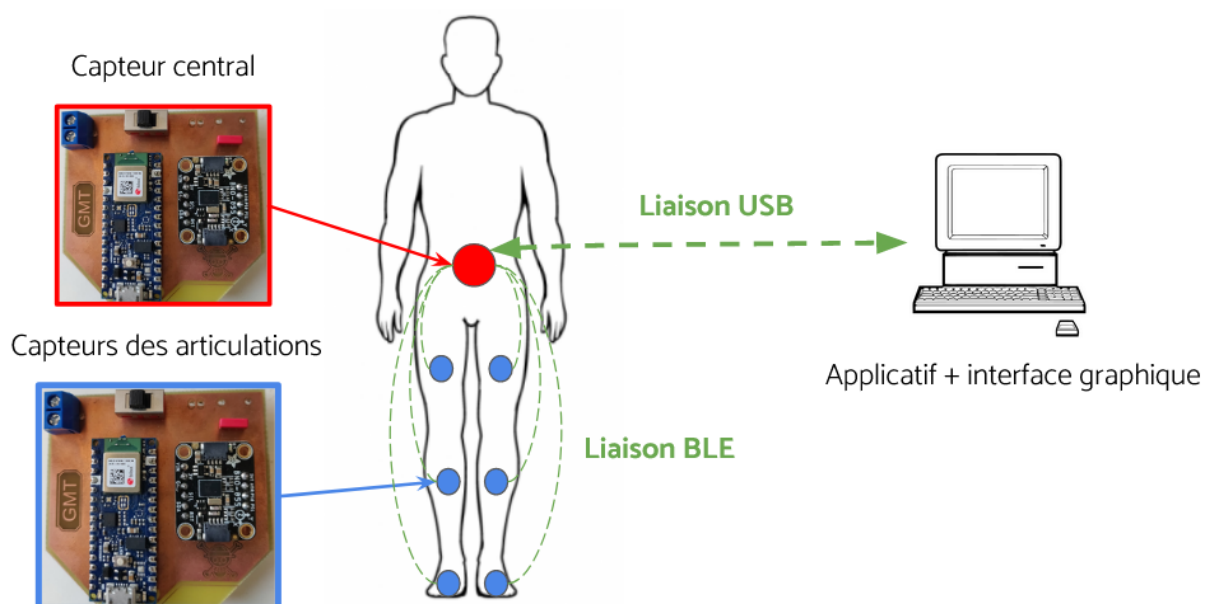
- Hardware : consistant en le matériel nécessaire et comment l'utiliser
- Software : consistant à téléverser les programmes Arduino et à utiliser l'interface Python

Pour toutes questions liées à l'utilisation ou à la compréhension de certains fichiers, n'hésitez pas à contacter un des contributeurs du github.

Matériel nécessaire

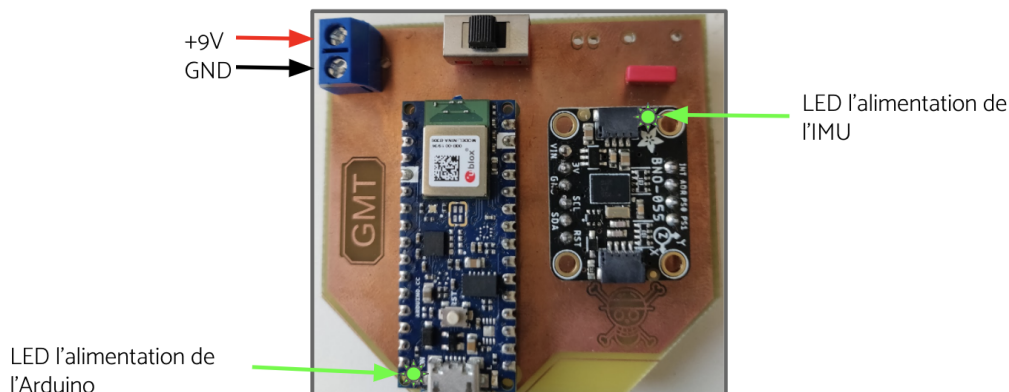
Pour se servir de notre dispositif, il est essentiel de se munir de tout le matériel nécessaire. Pour ce qui est du dispositif en lui-même il nous faudra donc les éléments suivant:

- 7 boîtiers;
- 7 batteries chargées 9V;
- 7 circuits imprimés;
- Un ordinateur;
- Un câble micro-usb -> USB.



Comment utiliser le dispositif

Pour installer le matériel il faut tout d'abord passer les languettes de scratch dans les boîtiers grâce à la fente prévue à cet effet. Puis brancher les batteries aux cartes imprimées de cette manière :



Si le système est correctement alimenté, alors les LEDs d'alimentation vertes seront allumées sur l'Arduino ainsi que sur le BNO055

Une fois chaque carte alimentée, il faut insérer les batteries dans le boîtier à l'intérieur des encoches afin de les maintenir. Une fois les batteries placées, nous pouvons ajouter le circuit imprimé dans le boîtier et le fermer. La carte reliée au dos doit être reliée grâce au câble micro-usb à un ordinateur portable.



Pour mettre le circuit en route il suffit d'activer les interrupteurs de chaque carte.

Téléversement des programmes Arduino

Le logiciel de développement utilisé pour les programmes des cartes Arduino est l'IDE Arduino. Il faudra donc tout d'abord installer l'environnement, pour installer les drivers et les librairies qui seront utilisés pour communiquer avec l'Arduino Nano 33 BLE ainsi qu'avec l'IMU BNO055.

Puis il faudra téléverser les programmes sur l'ensemble des cartes arduino.

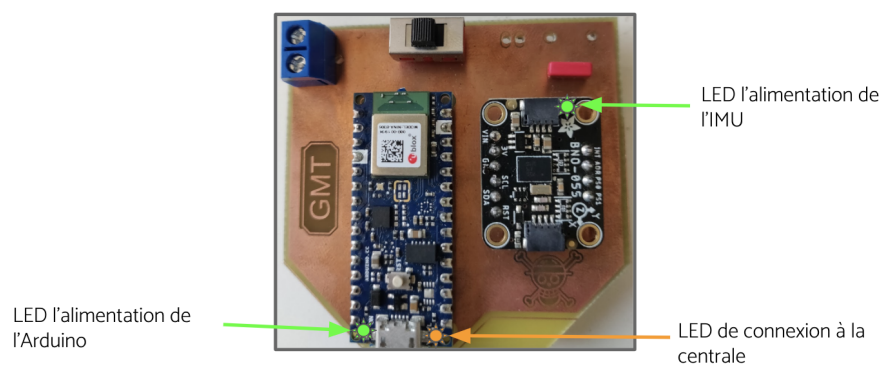
Commençons par les 6 capteurs périphériques, les programmes sont situés dans l'arborescence suivante dans le Github : ***Walking-Analysis-Project/Arduino/Peripherals***

Chacunes des 6 Arduinos périphériques devront être téléversées avec un programme différent peu importe l'ordre. Les programmes sont notés **Peripheral_X.ino** avec X allant de 1 à 6. Une fois cette étape réalisée, les périphériques sont désormais initialisés.

Passons maintenant à la centrale, le programme se situe dans l'arborescence suivante : ***Walking-Analysis-Project/Arduino/Central*** et le programme se nomme **Central.ino**, il suffit de le téléverser sur la carte centrale.

Lorsque toutes les cartes auront reçu leur programme maintenant il faut réaliser une calibration rapide des IMUs, pour cela il suffit simplement de tenir dans sa main la capteur et de l'orienter dans toutes les directions.

Une fois toutes les étapes précédentes réalisées, le système est prêt à l'emploi. Afin de notifier si un périphérique se connecte à la centrale, il suffit de regarder si la LED orange est allumée ou non. Si le système fonctionne bien, la centrale devrait se connecter puis se déconnecter à l'ensemble des périphériques. A noter qu'une connexion dure moins d'une seconde.



Si jamais le système cesse de fonctionner, il suffit de re-téléverser le code de la centrale ou bien de "reset" les périphériques bloqués en connexion en appuyant 2 fois sur le bouton reset de l'Arduino concernée.

Utilisation de l'interface Python

Notre interface a été réalisée sur Python (version 3.9.2). L'IDE de développement a été Visual Studio Code mais ce n'est absolument pas une nécessité de l'utiliser pour faire fonctionner ces programmes.

Cette interface utilise différentes librairies qui devront être téléchargées pour pouvoir exécuter le programme (voir en-tête de chaque programme) :

- numpy;
- matplotlib;
- time;
- serial;
- tkinter.

Les fichiers d'interface sont situés dans le répertoire "**python**" du github.

Il existe 4 fichiers différents avec chacune une particularité (décrite dans l'en-tête en commentaire) :

- **interface.py** : est un fichier d'interface (de l'arduino) réalisé avec tkinter permettant de lisser les courbes (peut poser problème);
- **interface_bis.py** : est le même fichier que interface.py, cependant sans le lissage des courbes (à utiliser principalement);
- **serie.py** : qui est notre interface avec un affichage des courbes en temps réel (fonctionne moyennement car beaucoup de données sont compliquées à être affichées);
- **data_save.py** : affiche les différentes courbes de l'arduino, cependant sans l'interface créée avec tkinter.

Avant l'exécution d'un de ces 4 programmes, il faut préciser sur quel port COM est connectée notre arduino centrale.

Pour paramétrer le port COM que nous utilisons dans le programme, il faut modifier la ligne suivante : `arduinoData=serial.Serial('COM4', 115200)`.

Une fois les librairies et les fichiers téléchargés, le port COM paramétré, vous pouvez exécuter un des 4 fichiers en utilisant python3.