

Rapport de Séance n°5 : 04/02/2020

Ordre du jour :

1. Fablab
2. Finition du code permettant de détecter les coups
3. Tests sensibilité et marge d'erreur
4. Code vérification de la combinaison
5. Code pour stocker le tableau dans la carte arduino

1. Fablab

Avant les vacances, nous sommes allés au fablab pour fabriquer notre porte que j'avais préalablement modélisé sur inkscape.



Pour finaliser notre maquette, il ne nous reste plus qu'à attacher notre porte à la « boîte » à l'aide d'une charnière, ajouter une poignée de porte et visser un socle au tout pour que ce soit bien stable lorsqu'on tapera sur la porte.

2. Finition du code permettant de détecter les coups

Le code qu'on avait écrit la séance dernière comptabilisait mal les coups, car il n'intégrait pas le coup 0. De ce fait, on a sorti le bout de code qui permet d'afficher le compteur sur la console du « if ». De plus, on a vérifié que notre code était bien fonctionnel et ne comportait aucune erreur.

Ensuite, on a adapté notre programme au capteur piezo car depuis nous l'avons codé avec un bouton poussoir. Et, nous avons testé notre code. Les coups n'étaient pas bien comptabilisés car on avait mis un délais trop important. On a donc du effectuer plusieurs tests pour ajuster notre délai.

3. Tests sensibilité du capteur et marge d'erreur des coups

Dans un premier temps, nous avons effectué des tests sur la sensibilité du capteur piézo. En effet, jusque-là on tapait sur le capteur, alors qu'on doit taper de l'autre côté du capteur, sur la porte. De ce fait, il fallait augmenter la sensibilité. En effet, il fallait que le capteur capte bien tous les coups sans qu'on ait à taper trop fort sur la porte mais qu'il ne capte pas les mouvements de portes ou autre.

Ensuite, on a effectué des tests pour se rendre compte de la marge d'erreur sur le temps des coups à appliquer car on ne peut pas taper exactement le même rythme à chaque fois. Après 10 tests chacun sur un rythme donné, nous avons remarqué (voir tableur) que la différence entre le temps du coup qu'on effectué et celui de base était maximum de 60ms donc nous avons décidé de mettre une marge d'erreur de plus ou moins 60.

4. Code vérification de la combinaison

Ensuite, nous avons écrit le code de vérification de la combinaison. Il s'agissait de vérifier que chaque coup de la combinaison effectuée, qui sont stockés dans un tableau soient bien égaux plus ou moins la marge d'erreur avec le tableau de coups de la combinaison de référence.

Il faudra donc qu'on écrive le code permettant à l'utilisateur d'enregistrer sa combinaison de référence.

5. Code pour stocker le tableau dans la carte arduino

On avait vu que la carte arduino pouvait stocker sur 1 octet des données grâce à la librairie EEPROM. Sauf que nos valeurs ne faisaient pas forcément 1 octet mais parfois plus, comme par exemple quand on a 356, qui est stockée sur 2 octets. De ce fait, nous avons cherché chacun de notre côté comment remédier à ce problème et j'ai trouvé sur ce site :

(<https://www.carnetdumaker.net/articles/stocker-des-donnees-en-memoire-eeeprom-avec-une-carte-arduino-genuino/>), on pouvait stocker des données typées sur plusieurs octets grâce a «EEPROM.get » ou « EEPROM.put ». Et, aussi, que les « int » étaient stockés sur 2 octets. De ce fait, nous avons pu écrire notre programme permettant de stocker chaque valeur du tableau de la combinaison sur la mémoire de la carte arduino.