Nathan
DJIAN-MARTIN
G2

Rapport de séance n°1 03/02/2020

Nous avons travaillé ensembles toute la séance sur la partie programmation car nous n'avons pas encore toutes les pièces pour assembler la maquette et c'est la partie la plus demandeuse de travail de notre projet.

Nous avons commencé par corriger une erreur mineure sur nos messages de debuggage de notre programme d'enregistrement d'une combinaison.

Nous avons ensuite dupliqué ce programme et nous l'avons adapaté pour qu'il fonctionne avec le capteur piezo-électrique. Le programme initial fonctionnait avec un bouton pour faciliter les phases de test et les problèmes liés à la sensibilité diu capteur piezo-électrique. Nous avons changé le pin d'entrée et ajouté une variable « threshold » qui est un entier compris entre 0 et 1023 qui est le seuil à partir duquel on considère qu'un coup a été frappé. Nous avons fais plusieurs tests de sensibilité mais pour l'instant cette valeur n'est pas importante car la porte n'est pas dans sa configuration finale, nous l'ajusteront quand la maquette sera assemblée.
Notre programme de détection de combinaison est donc désormais fonctionnel.

Nous nous sommes alors penchés sur la comparaison entre deux combinaisons, la combinaison de référence déterminée par l'utilisateur et la combinaison toquée. Le principe du programme est de vérifier pour chaque valeur du tableau si elle est comprise entre la valeur de référence ± un seuil de tolérance. Pour déterminer ce seuil de tolérance nous avons chacun notre tour tenté de reproduire la même combinaison 10 fois pour avoir une idée de l'écart le plus grand qu'on pouvait avoir. Le tableur de nos essais est disponible sur le GitHub. Nous avons reproduit une combinaison de 3 coups rapides, les valeurs mesurées sont les écarts entre 2 coups en millisecondes. Nous avons fixé ce seuil de tolérance à 60ms. C'est une valeur assez petite mais nous préférons que l'utilisateur ait à retoquer sa combinaison plutôt que notre système ne soit trop tolérant envers les erreurs et accepte potentiellement des faux-positifs.

Nous avons terminé cette séance en stockant le tableau contenant les écarts de notre combinaison dans la mémoire EEPROM. La subtilité vient du fait que les entiers sont codés sur 2 octets. Or les « cases » de l'EEPROM ne peuvent contenir qu'un seul octet. Au moment d'écrire l'entier dans une adresse de l'EEPROM nous devons donc à l'écriture de chaque nouvel entier incrémenter l'indice de l'adresse de 2. Nous avons

testé le programme et après redémarrage de l'Arduino nous avons récupéré la combinaison secrète de l'utilisateur (elle était codée en dur dans ce programme de test). Il est d'ailleurs lui aussi disponible sur le GitHub.