Rapport de Séance n°4 : 13/01/2020

Ordre du jour :

1. Oral de mi-projet
2. Design de la porte
3. Code principal permettant de stocker la combinaison

1. Oral de mi-projet

La séance a débuté avec notre oral de mi-projet. Durant celui-ci, nous avons appris que les mémoires flash avaient une durée de vie de 100 000 cycles. Il va donc falloir qu'on réfléchisse et qu'on fasse attention à comment une fait notre code lorsqu'on utilise la mémoire EEPROM.

2. Design de la porte

Aujourd'hui, j'ai commencé et terminé le design de la porte sur le logiciel que j'avais téléchargé la fois dernière. J'ai inséré le logo de Polytech et de notre projet, qu'il a fallu d'abord détourer et noircir (le logiciel ne capte que des nuances de gris et de noir).
Il faut donc qu'on prenne rendez-vous au fablab pour les faire graver sur une planche en bois.
Il ne nous restera ainsi plus qu'à visser une petite poignée de tiroir sur la porte et à fixer celle-ci sur la « boite » avec une charnière.
De plus, j'ai regardé comment nous allons relier le servomoteur au loquet, et pour l'instant nous les relierons grâce à un trombone.

3. Code principal permettant de stocker la combinaison

Ensuite, j'ai rejoint Nathan sur le code principal. C'est le code qui permet de récupérer et stocker la combinaison dans un tableau. Nathan m'a expliqué ce qu'il avait déjà codé et qu'il testé le code avec un bouton poussoir parce que c'était plus simple pour le moment.
Le code qu'il avait déjà écrit ne marchait pas. En effet, il comptait 10 coups lorsqu'on appuyait qu'une fois sur le bouton poussoir.
Ensemble, nous avons réussi à débuguer le code et à le finaliser.
Tout d'abord, nous nous sommes rendu compte que ce qui permet de comptabiliser le nombre de coup était mal placé dans le programme ce qui faisait que le programme comptait 10 coups au lieu d'un.
Après avoir corrigé cette erreur, nous nous sommes rendu compte qu'il comptait encore 3 coups quand on en faisait que 1. C'était en fait du au rebonds. En effet, lorsqu'on a augmenté le delay il comptait le bon nombre de coup.
Ensuite, nous avons ajouté des lignes de débug et nous nous sommes rendu compte que le programme effectuait le mauvais calcul quand il faisait « millis() – reftime ». Il prenait le

reftime du coup qui venait d'être effectue et nous voulions qu'il prenne celui du coup d'encore avant. De ce fait, j'ai soumis l'idée de créer une variable qui stocke le temps actuel millis() ou le coup est effectué juste après qu'il soit effectué et que seulement après le calcul on assigne cette valeur à la variable reftime.

Nous avons ainsi, réussi à finaliser ce code.