

Rapport de séance n°1 10/02/2020

Lors de cette séance nous nous sommes concentrés sur l'assemblage de notre maquette. Nous avons en effet acheté un verrou, des charnières et une poignée de tiroir. L'après-midi, chez nous, nous avons commencé à réfléchir à la structure du programme principal : KnockiCore.

1 - Assemblage de la maquette

Nous avions déjà la structure de la porte, une boîte ouverte que nous avons découpé au FabLab ainsi que la porte sur laquelle nous avons gravé au FabLab le logo de notre projet et nos prénoms.

Nous avons donc assemblé la porte à la boîte pour placer nos composants dans une configuration proche de la configuration finale et commencer à tester les mécanismes du projet (porte et actionnement du verrou par le servomoteur). Créer une porte a été plus compliqué que nous l'avions imaginé. Nous avons percé des trous dans la porte et dans la boîte. Pour les trous de la boîte nous avons dû en faire un de l'intérieur à travers la charnière pour être certains qu'elle soit bien positionnée, puis le deuxième de l'extérieur car il était trop proche de l'angle de la boîte et la perceuse ne passait pas. Nous nous sommes vite rendu compte qu'il était très facile d'abîmer la planche lors d'un perçage.

De plus, nous n'avons pas pu assembler la porte comme nous l'avions imaginé en plaçant les charnières à l'extérieur (car elle était en contact avec un bord de la boîte ce qui l'empêchait de s'ouvrir). Nous avons donc vissé les charnières à l'intérieur de la boîte mais la porte est légèrement surélevée et son côté gauche n'est pas en contact avec la structure de la boîte donc lorsque l'on tape elle amorti un peu trop les coups ce qui va nous obliger à augmenter la sensibilité du capteur piezo-électrique. Nous avons ensuite vissé la poignée de la porte qui permet de tirer la porte et est principalement un atout esthétique.

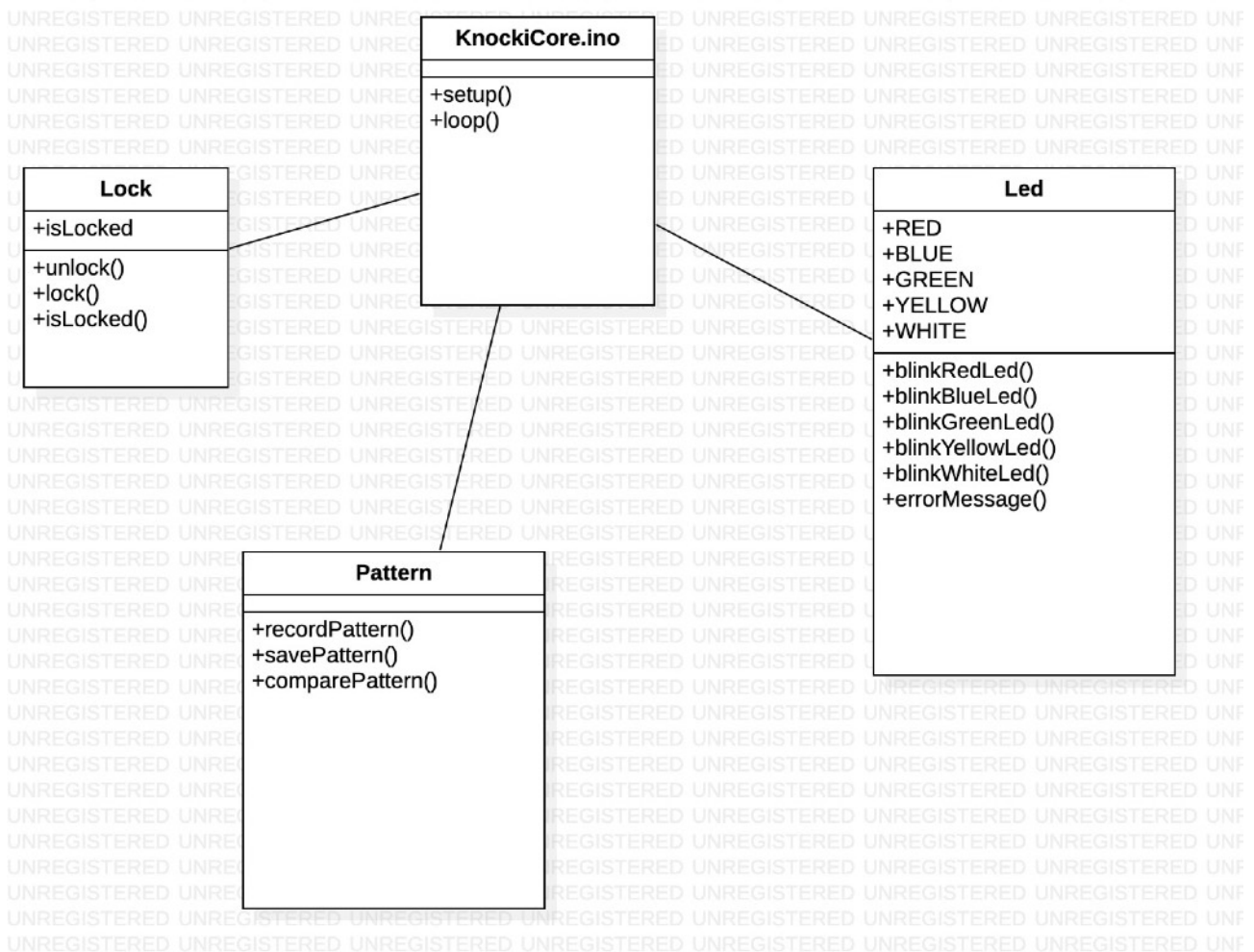
2 - Test du mécanisme d'ouverture

En fin de séance nous nous sommes intéressés au mécanisme qui permet l'ouverture automatique du verrou avec le servomoteur. Nous voulions initialement imprimer une pièce en 3D permettant de relier le servomoteur au verrou afin de transformer le mouvement de rotation en mouvement de translation mais un simple trombone reformé à notre convenance remplit très bien les objectifs. Nous devons percer un trou dans la boîte pour que le verrou puisse bloquer la porte.

3 - Programme principal

L'après-midi nous avons voulu commencer à développer le programme principal. Nous avons en effet réussi à réaliser tous les petits éléments dont nous aurons besoin pour mener à bien notre projet (actionner le servomoteur, détecter un coup, enregistrer une combinaison, sauvegarder une combinaison dans la mémoire EEPROM, comparer des combinaisons etc...). Tous les petits modules de test sont disponibles sur le GitHub. Nous devons donc réfléchir à l'algorithme principal et à l'architecture du programme.

Voici le diagramme UML du programme principal :



Le fichier principal est KnockiCore, c'est littéralement le noyau de notre programme. Ensuite les 3 autres fichiers contiendront toutes les fonctions utiles durant la durée du vie de programme. Le but est de faire en sorte que les noms de fonctions soient explicites au possible pour ainsi faciliter la lecture du KnockiCore et l'alléger. Le fichier lock contiendra les méthodes pour actionner le servomoteur et ouvrir/fermer le verrou. Le fichier Led, les fonctions pour interagir avec chaque led et ainsi permettre un retour visuel avec l'utilisateur. Le fichier Pattern quant à lui, contient les fonctions permettant de gérer les combinaisons secrètes (enregistrer, sauvegarder dans la EEPROM et comparer

deux combinaisons. Nous allons sûrement ajouter un fichier Wireless permettant de gérer les interactions avec la télécommande infrarouge.

Pour le moment les variables ne sont pas définies car nous avons à peine commencé à nous documenter sur la meilleure façon de découper un programme Arduino en plusieurs fichiers, nous n'avons pas trouvé de solution évidente sur Internet. Nous devons encore comprendre la portée qu'on les variables et l'utilité des différentes extensions de fichiers (.h, .cpp, .ino...), en effet, il semble y avoir plusieurs écoles quant à l'utilisation de certaines extensions pour certaines utilisations. Une fois le problème multi-fichier résolu nous pourrions commencer à implémenter les différentes fonctions dont le code est déjà fonctionnel dans nos modules de test.