

RAMIARA Maxime  
TAMI Théo  
4AE-SE3

# Bureau d'études C++

## Rapport

<https://github.com/TheoTami09/BEPOO>

2020-2021

## 1) Introduction

Après avoir appris les bases de la conception orientée objet et du langage C++, il ne nous restait plus qu'à mettre en pratique nos connaissances. Ainsi, au cours de ce bureau d'études, nous avons réalisé une application du C++ au domaine des objets connectés. Pour ce faire, nous avons utilisé un **microcontrôleur ESP8266** doté d'une carte wifi et programmable en arduino et un ensemble de capteurs et d'actionneurs.

Nous avons choisi de concevoir un **système de sécurité connecté** visant à protéger l'accès à un bâtiment, une pièce ou un coffre fort par exemple. Le but de ce système est de surveiller en temps réel, grâce à la connexion internet, toute tentative d'intrusion.

## 2) Matériel choisi et fonctionnement du système

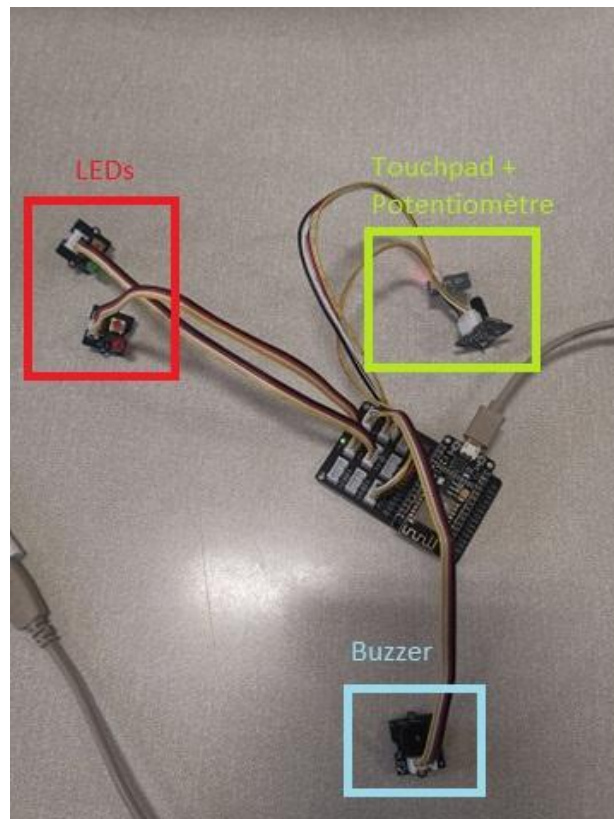


Figure 1 : Photo du système complet

Les capteurs utilisés sont : un **potentiomètre** et un **capteur tactile**.

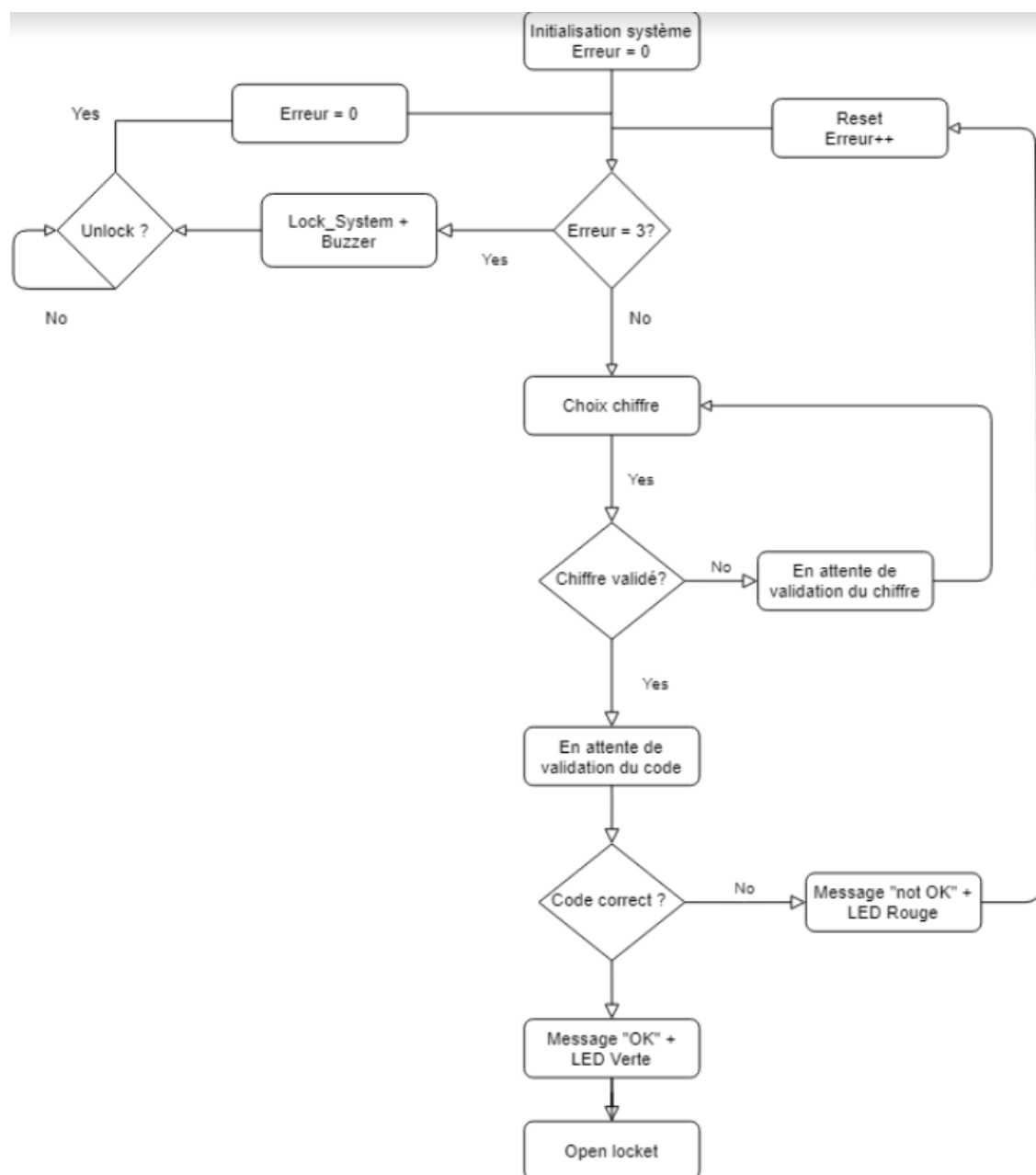
Les actionneurs utilisés sont des **LEDs** et un **buzzer**.

Il y a également la **page web** qui s'actualise en temps réel et qui affiche des messages. Elle permet également d'agir sur le système.

Grâce à un potentiomètre, il faut rentrer un code, **chiffre par chiffre**, qui sera envoyé sur une page web, de plus, il faut valider chaque chiffre avec le bouton tactile. Ensuite, il faut valider le code en entier. S'il est correct, le message "code ok" s'affiche sur la page et la LED verte s'allume. Dans le cas contraire, le message "erreur" s'affiche et la LED rouge s'allume.

Au bout de 3 erreurs, le système se verrouille et un buzzer s'allume signifiant qu'il y a **tentative d'intrusion**. Enfin, sur la page web, on peut déverrouiller le système avec le bouton "unlock", ce qui allume les deux LED en même temps.

### 3) Schéma de fonctionnement logiciel

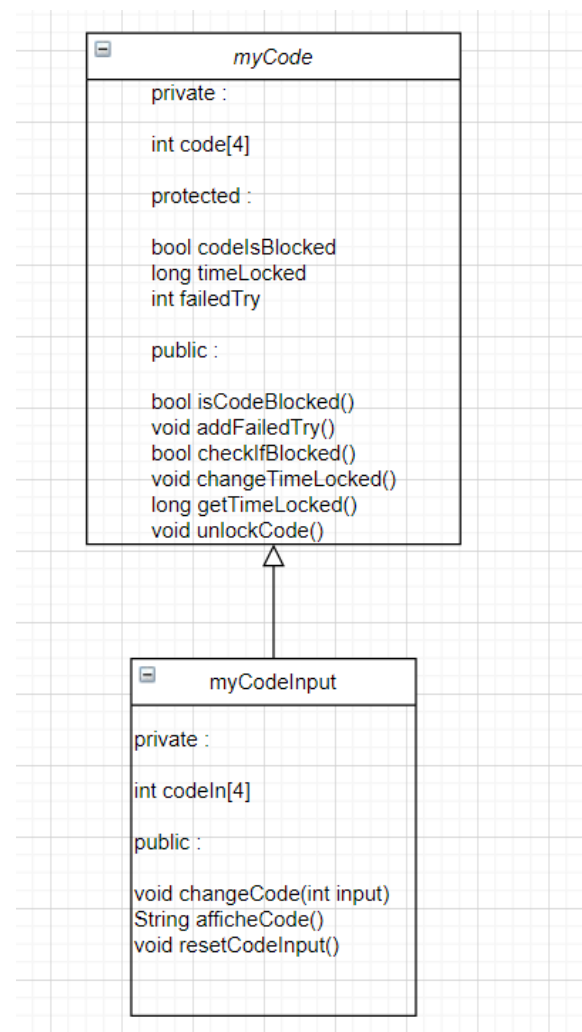


Pour plus de détails concernant le fonctionnement logiciel de notre système, le code est disponible sur notre GitHub.

#### 4) Diagramme de classe

Notre programme comporte plusieurs classes qui sont les suivantes :

- myCode : la classe principale de l'objet "code", qui contient toutes les fonctions nécessaires et les redéfinitions d'opérateurs.  
Le choix de mettre codeIsBlocked, timeLocked et failedTry en protected permet dans une certaine mesure de s'assurer de l'intégrité et de la sécurité des informations, puisqu'elles ne pourront être modifiées que par la classe elle-même ou classes filles.
- myCodeInput : classe fille de myCode, elle sert principalement à gérer l'entrée d'un code, et contient différentes fonctions d'affichage et de modification du code.



## **5) Conclusion**

Tout au long de ce bureau d'études, nous avons mis en application nos connaissances en conception et programmation orientée objet acquises au cours de ce semestre. En outre, nous avons pu nous initier aux concepts de l'internet des objets. Cela nous a permis de mieux comprendre en quoi ce concept offre une multitude de possibilités dans tous les projets de systèmes embarqués.

Par ailleurs, nous regrettons qu'il y ait eu un problème au niveau de la commande d'un de nos capteurs : le capteur RFID. Nous avions à la base imaginé un système de sécurité autour des tags RFID mais nous n'avons reçu le capteur qu'à la dernière séance. Il a donc fallu repenser le projet, ce qui nous a retardé dans notre travail.

Avec le matériel adapté, nous aurions pu rajouter un système de mot de passe personnalisé pour chaque utilisateur possédant un tag RFID. Nous aurions également pu vérifier la présence de clones de tag RFID en affichant leurs clés en hexadécimal sur la page web.

Pour conclure, ce projet était très intéressant et très formateur. Malgré les problèmes rencontrés, nous avons quand même atteint notre objectif de réaliser un système de sécurité fonctionnel.