

Langage C++

**BE: Application du C++ au domaine des objets
connectés**

I4AEIL11

*4ième année - AE - Système Embarqué - Groupe 3
3600024 - Catherine Vaugelade Berg
3600026 - Hermund Døssland
3500008 - Njord Løkebø*

Table des matières

Introduction	2
Déroulement	2
Conception	3
Cas d'utilisation	3
Diagramme de classe	4
Lancement du projet	4
Problèmes rencontrés	4
Conclusion	6

Introduction

Le but de ce BE était de nous familiariser avec des différents concepts dans le langage C++ et l'utilisation de "l'outil de développement" GitHub en concevant notre propre projet de programmation. Nous allons explorer des concepts comme les classes, le héritage, l'utilisation de STL et la gestion d'exceptions. Pour l'utilisation de GitHub il s'agissait d'une introduction à l'outil et donc l'apprentissage de toutes les commandes basiques ainsi que l'utilisation des branches.

Déroulement

Au début du BE nous avons tout d'abord fait un brainstorming dans la groupe pour réfléchir ouvertement à un projet qui pouvait nous intéresser. Le sujet étant libre il fallait trouver quelques idées qui nous semblaient intéressantes et voir ensemble si c'était un travail que nous voulions développer. Les restrictions imposées par la pandémie ont fait que toute implémentation pratique était impossible, il fallait donc aussi trouver un projet qui pourrait se faire complètement en virtuel.

Après avoir trouvé une idée nous avons fait le diagramme de classe pour avoir une vue de tous les éléments qui constituaient notre programme. Ceci nous a permis de répartir le travail entre nous et commencer le codage. Nous utilisions GitHub pendant toute cette partie du projet pour pouvoir travailler chacun sur notre partie sans interférer le travail des autres. Enfin quand toutes les parties étaient finies nous avons tout mis ensemble pour faire le programme final.

Conception

Cas d'utilisation

Nous avons choisi comme projet de faire un *SmartParking* pour des voitures électriques qui donc chargèrent les voitures dans un parking avec une porte (ou barrière) pour fermer et ouvrir le parking. Le concept a les fonctionnalités suivantes:

- Capteur si il y a un voiture devant la porte
- Un servo moteur qui ouvre la porte
- Une batterie qui mesure le niveau de batterie
- Un bouton pour démarrer le chargement
- Un LED jaune qui s'allume si la charge est en cours
- Un LED verte qui s'allume si la charge d'est terminé
- Un LED rouge qui s'allume si le chargeur est éteint
- Le chargeur s'éteint automatiquement si la batterie est à 100%.
- Un bouton pour ouvrir la porte si une voiture doit sortir

Ce projet est réaliser en virtuel, donc aucun matériel n'est utilisé. À cause de la pandémie SARS-CoV-2, nous avons effectuer le travail à distance, en utilisant GitHub.

Diagramme de classe

Notre diagramme de classe montre l'architecture du concept de *SmartParking*, et voici ci-dessous::

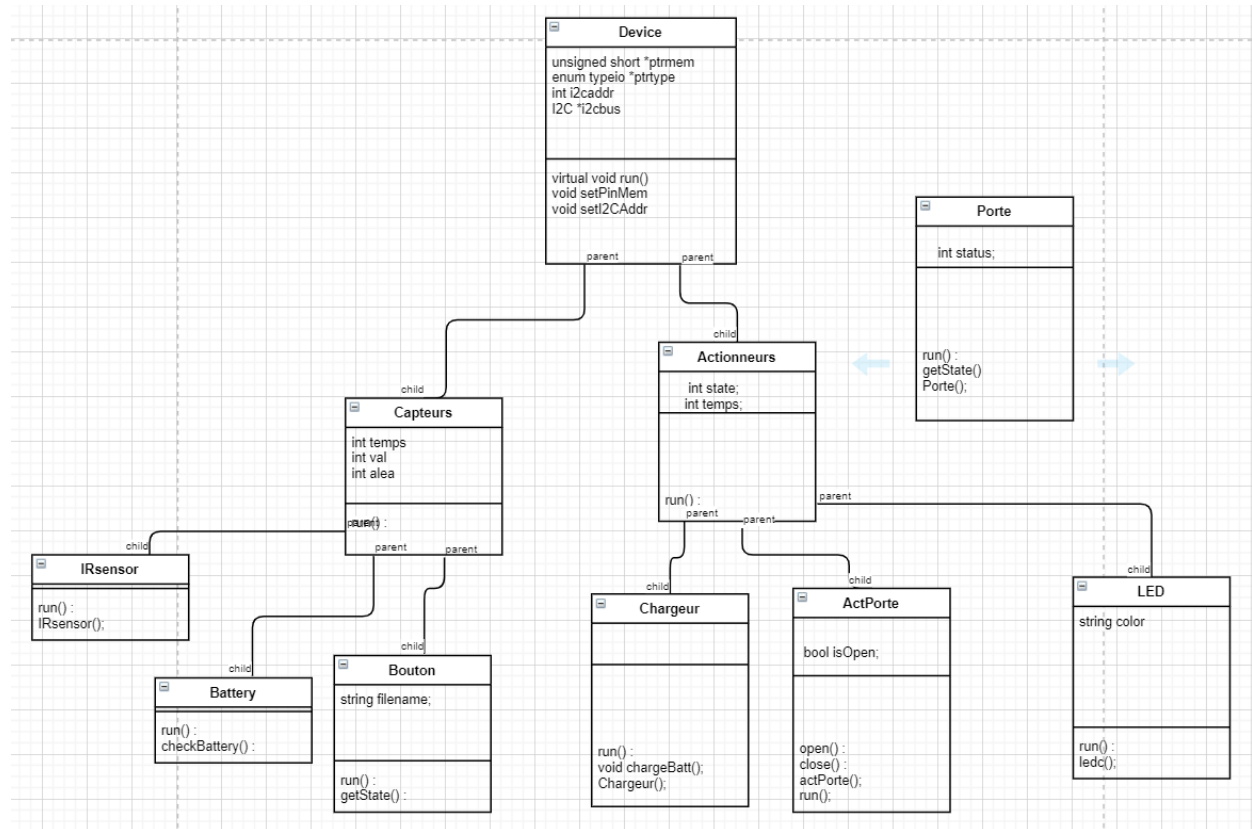


Figure 1: Diagramme de classe de *SmartParking*

Nous avons la classe *Device* qui regroupe les classes *Capteurs* et *Actionneur*. Dans la classe *Capteurs* nous avons comme attributs le *IR-sensor*, *Battery* et le *Bouton*. Ensuite dans *Actionneur* nous avons *Porte*, *ActionneurPorte*, et *LED* comme attributs.

Le capteur infrarouge aurait dû se brancher sur un pin GPIO, il envoie un '0' lorsqu'il détecte un objet et 1 quand il n'y a rien. Le servo-moteur utilise un port GPIO avec PWM afin de contrôler la barrière de notre parking.

Lancement du projet

Pour lancer notre projet, il faut réaliser plusieurs étapes : envoyer les fichiers via le terminal, les compiler à l'aide de la commande `./arduino/compile_linux.sh`. Nous avons créé plusieurs fichiers pour simuler des boutons, une voiture arrivant et une voiture partant.

Problèmes rencontrés

Pendant le projet nous avons rencontré deux types de problèmes. Le premier type concernait les différentes erreurs liées aux classes et l'héritage. Le deuxième était l'utilisation de GitHub, qui nous

```
Serial: -----Le niveau de batterie est : 25
((LED rouge))
The garage door is open
Serial: -----Le niveau de batterie est : 25
((LED rouge))
The garage door is closed
Serial: -----Le niveau de batterie est : 25
-----charging...
The garage door is closed
-----charging...
-----charging...
((LED jaune))
Serial: -----Le niveau de batterie est : 40
-----charging...
The garage door is closed
-----charging...
-----charging...
((LED jaune))
Serial: -----Le niveau de batterie est : 55
-----charging...
The garage door is closed
-----charging...
-----charging...
((LED jaune))
Serial: -----Le niveau de batterie est : 70
-----charging...
The garage door is closed
-----charging...
-----charging...
((LED jaune))
Serial: -----Le niveau de batterie est : 85
-----charging...
The garage door is closed
-----charging...
La batterie est pleine, va conduire!
((LED jaune))
Serial: -----Le niveau de batterie est : 100
La batterie est pleine, va conduire!
The garage door is closed
La batterie est pleine, va conduire!
La batterie est pleine, va conduire!
((LED vert))
Serial: -----Le niveau de batterie est : 100
```

Figure 2: Capture d'écran de l'exécution du programme

à pose plusieurs difficultés. Ceci nous a forcé parmi autres choses à réécrire certaines parties du code qui ont été supprimé plusieurs fois ce qui à amplement augmenté le temps nécessaire pour finir notre produit. Notre manque de familiarisation de cet outil est sûrement l'une des raisons pour ces difficultés et peut être nous auront du nous renseigner encore plus sur son utilisation avant de commencer à coder pour éviter un tel perte de temps.

Conclusion

Au cours de ce BE nous avons pu suivre le processus du développement d'un projet. Il nous a permis de voir comment peut se faire un programme numérique d'une plus grande taille et d'essayer de gérer les contributions de plusieurs personnes sur ce même projet. Le sujet était aussi totalement libre, ce qui nous a poussé d'être créatif et de penser à des nouvelles solutions possibles pour des tâches que nous voyons autour de nous dans la vraie vie. Ce BE nous a aussi permis de voir comment résoudre une grande variété de problèmes concernant la création des codes, une expérience qui est sans doute utile pour un futur ingénieur. Il nous a aussi permis d'avoir une première expérience pratique avec l'outil GitHub, qui est un outil connu dans le monde des développeurs.

La gestion de temps est un autre facteur qui s'est avéré important. Nous avons vu que certaines parties du projet, comme l'assemblage de toutes les différentes parties de code dans GitHub et la résolution des erreurs, pouvaient prendre considérablement plus de temps que prévu. Ceci a entraîné que certaines fonctionnalités planifiées pour le produit final ont dû être abandonnées, comme la mise en place de plusieurs places de parking et l'implémentation d'une gestion d'exceptions. Enfin, ce BE nous avons pu mettre en pratique nos connaissances et c'était une très bonne expérience prise en compte des restrictions d'application physique due aux circonstances actuelles.