INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

4AE-SE

RAPPORT DE PROJET DE PROGRAMMATION ORIENTÉ OBJET

SMART PARKING

HOUSBANI Soufiane
BENCHEKROUN Amine

2023-2024

SOMMAIRE

INTRODUCTION :

Dans le cadre de l'Unité de Formation en langage C++, notre projet consiste à concevoir un bureau d'études axé sur le développement d'un système novateur. Après une analyse approfondie, notre choix s'est porté sur la création d'un parking intelligent.

L'implémentation du projet s'effectue en langage orienté objet C++, mettant en œuvre des capteurs et des actionneurs. L'objectif principal de ce bureau d'études est de nous familiariser avec les concepts fondamentaux de la programmation orientée objet, tels que l'héritage, l'encapsulation et le polymorphisme.

Nous utilisons une carte Arduino ESP8266 fournie par Expressif Systems, équipée de la fonctionnalité Wi-Fi. L'IDE Arduino est utilisée pour le développement du code, auquel nous avons intégré divers capteurs et actionneurs pour répondre aux exigences du projet.

Entrées	Sorties
2x Capteurs ultrasons	2x Servomoteurs
	1x Ecran LCD

PRESENTATION DU PROJET :

Nous avons entrepris de simuler la gestion d'un parking dans le cadre de notre projet. Pour ce faire, nous avons mis en œuvre deux barrières, contrôlées par deux Servo Moteurs, deux capteurs de présence de type Capteur à Ultrason, et un afficheur LCD.

Le tableau ci-dessous offre un récapitulatif des capteurs et actionneurs que nous avons intégrés, ainsi que les broches (Pin) auxquelles ils sont connectés :

Capteurs/Actionneurs	Pins
1er capteur d'accès : UltraSonic Sensor HC-SR04	Trig-> GPIO12 (pin D6) Echo-> GPIO14 (pin D5)
2ème capteur de sortie : UltraSonic Sensor HC-SR04	Trig-> GPIO2 (pin D4) Echo -> GPIO13 (pin D8)
1er servo moteur : actionnant la Barriere d'entrée	Connecté à GPIO0 (pin D3)
2ème servo moteur : actionnant la Barriere de sortie	Connecté à GPIO15 (pin D7)
Afficheur LCD	GPIO5 & GPIO4 (pin D1 & D2)

Nous avons conçu un prototype de parking intelligent sur une breadboard, avec la carte ESP8266 au centre. Du côté 1 de la breadboard, nous avons installé un écran LCD, un capteur ultrason et un servomoteur. Du côté 2, nous avons positionné un capteur ultrason et un servomoteur.

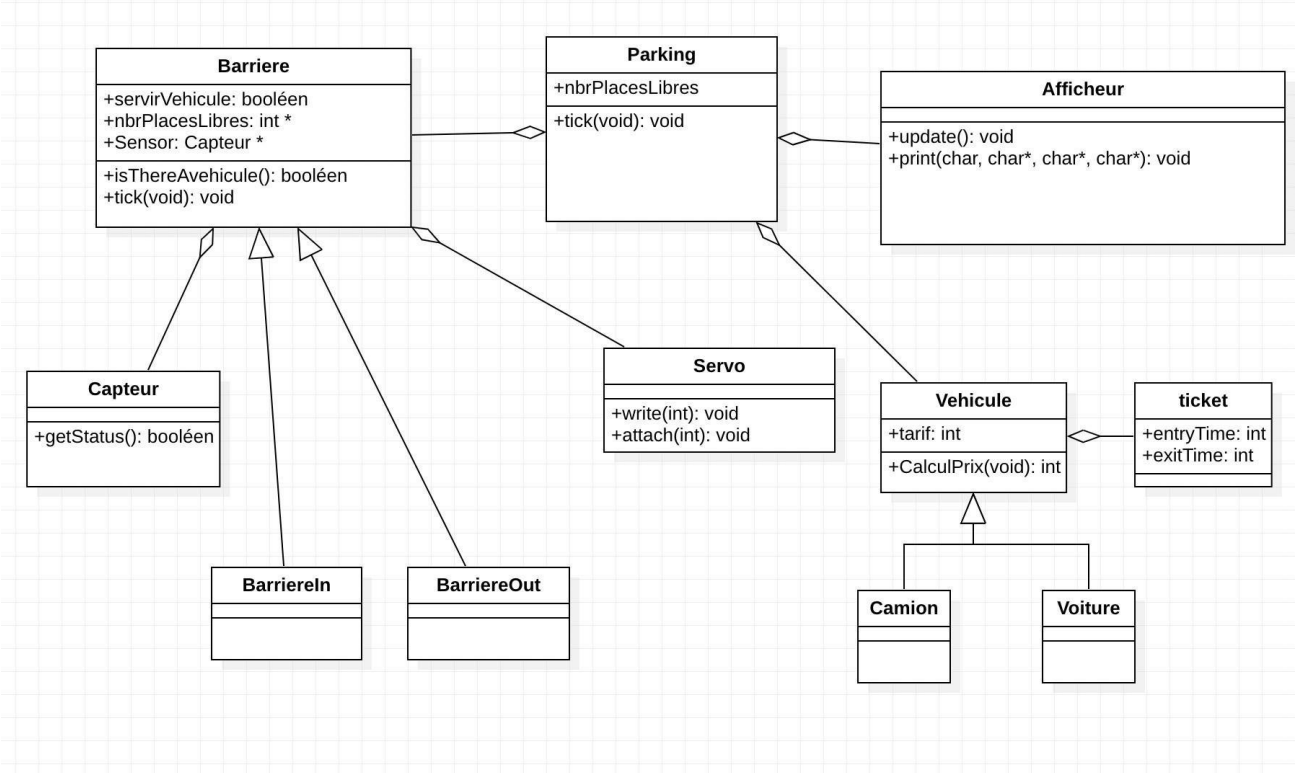
Le côté 1 représente l'entrée de notre parking. L'afficheur LCD informe l'utilisateur du nombre de places disponibles. En cas de disponibilité, le message "Soyez les bienvenus" s'affiche, sinon "Revenez plus tard". Lorsqu'un véhicule s'approche de la barrière d'entrée, le capteur ultrason détecte sa présence, permettant à la barrière de commander le servomoteur pour autoriser l'accès.

En l'absence de places disponibles, l'accès est refusé même si un véhicule attend. Il est nécessaire d'attendre qu'une place se libère.

Le côté 2 représente la sortie du parking, fonctionnant de manière similaire à l'entrée. Lorsqu'un véhicule souhaite quitter le parking, le capteur ultrason détecte sa présence et transmet l'information à la barrière de sortie. Cette dernière commande le servomoteur pour permettre le passage du véhicule.

À chaque entrée, le nombre de places disponibles diminue d'une unité. Lorsqu'un véhicule sort, le nombre de places libres est automatiquement incrémenté d'une place. Cependant, lorsque le nombre de places disponibles atteint la capacité maximale du parking, la barrière de sortie ne s'ouvre plus, même si le capteur de sortie détecte une présence.

DIAGRAMME DE CLASSE :



Notre diagramme de classe comprend plusieurs classes, telles que Parking, Barrière, BarriereIn, BarriereOut, Afficheur, Capteur, Servo, Véhicule, Camion, Voiture, et Ticket. Les classes Camion et Voiture sont des véhicules et héritent de la classe Véhicule, tout comme les classes BarriereIn et BarriereOut héritent de la classe Barrière.

Ces classes sont dotées d'attributs et de méthodes, qui ont été implémentés dans le code pour la réalisation de notre projet. Par exemple, la méthode print() de la classe Afficheur est utilisée pour afficher des messages à l'écran, et la méthode getStatus() de la classe Capteur calcule la distance de l'ultrason et détermine la présence d'un véhicule en renvoyant true ou false.

Des améliorations peuvent être apportées au programme en ajoutant des capteurs ultrasons supplémentaires au-dessus des capteurs existants pour distinguer les camions des voitures. Dans cette configuration, les voitures seraient détectées uniquement par le capteur inférieur, tandis que les camions seraient captés par les deux capteurs. De plus, une classe Ticket peut être ajoutée pour gérer le temps de présence des véhicules dans le parking et calculer le coût d'utilisation, transformant ainsi le parking en un service payant.

Le code intègre des fonctionnalités avancées telles que l'utilisation d'exceptions, la redéfinition d'opérateurs, le polymorphisme au niveau de la classe Barriere, la déclaration anticipée (forward déclaration) avec la classe Parking devant être déclarée avant la classe Afficheur, l'héritage, etc.

Cependant, des difficultés ont été rencontrées pendant le projet, principalement en raison de la limitation du nombre de broches (Pin) sur la carte. Cela nous a empêchés d'ajouter les deux capteurs à ultrasons pour différencier les types de véhicules. De plus, des contraintes temporelles importantes ont également été un défi, en particulier en envisageant une transition vers une carte offrant plus de broches et l'utilisation de la STL pour améliorer le programme

CONCLUSION :

Ce bureau d'études s'est révélé être d'une grande utilité pour nous. Il a joué un rôle essentiel dans l'approfondissement de nos connaissances acquises en classe et dans l'application pratique de ces notions. En outre, il a contribué à combler certaines lacunes en matière de codage et à nous familiariser avec l'IDE d'Arduino.

Cette expérience a également été une opportunité pour exprimer notre créativité et notre esprit d'innovation. Comme mentionné précédemment, plusieurs fonctionnalités auraient pu être ajoutées à notre projet si nous n'avions pas été confrontés à des contraintes technologiques et logistiques.

Travailler en autonomie et bénéficier d'une totale liberté nous a permis de créer notre propre cahier des charges, d'établir un planning de travail et de maîtriser efficacement les différentes phases de la gestion d'un projet. Cela nous a offert une expérience enrichissante en termes de responsabilité et d'organisation.