INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

- **HOUSBANI Soufiane**
- **Benchakroun Amine**

4AE-SE

RAPPORT DE PROJET DE PROGRAMMATION ORIENTÉ OBJET

SMART PARKING

2024-2023

SOMMAIRE

INTRODUCTION :

Dans le cadre de l'Unité de Formation Langage C++, il nous a été assigné la réalisation d'un bureau d'études visant à concevoir un système innovant. Après de minutieuses réflexions, notre choix s'est porté sur la création d'un parking intelligent.

La mise en œuvre de ce projet s'est opérée à travers la programmation en langage orienté objet C++, intégrant l'utilisation de capteurs et d'actionneurs. L'objectif principal de ce bureau d'études est de nous familiariser avec les concepts de la programmation orientée objet, tout en consolidant nos connaissances acquises en cours et travaux dirigés, tels que les notions d'héritage, d'encapsulation, de polymorphisme, etc.

Le support principal employé est une carte Arduino Seeduino Lotus fournie par Expressif Systems, équipée de la fonctionnalité WIFI. Nous avons utilisé l'IDE d'Arduino pour implémenter notre code sur la carte, en y intégrant les divers capteurs et actionneurs nécessaires à la réalisation de notre projet.

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

INFORMATION IMPORTANTE :

Dans l'utilisation de la carte Seeduino LOTUS, les exceptions n'ont pas été utilisées, principalement en raison de l'incompatibilité de la carte avec ce mécanisme. La Seeduino LOTUS ne prend pas en charge les exceptions, ce qui a nécessité l'adoption d'autres stratégies de gestion des erreurs et de contrôle des flux dans le code. Cette limitation technique a été prise en compte lors du développement, et des approches alternatives ont été employées pour assurer un fonctionnement stable du programme.
Cependant vous trouverez a la fin du fichier parking_int.ino la partie de code ou on a utilise les exceptions sur la carte esp86, ainsi que la redéfinition

PRESENTATION DU PROJET :

L'idée de notre projet réside dans la simulation de la gestion d'un parking, mettant en œuvre deux barrières (commandées par deux Servo Moteurs), deux capteurs de présence (Capteurs à Ultrasons), et un afficheur LCD. Le tableau ci-dessous synthétise l'ensemble des capteurs et actionneurs que nous avons employés, ainsi que les broches (pins) auxquelles ils sont connectés :

Capteurs/Actionneurs	Pins
1er capteur d'accès : UltraSonic Sensor HC-SR04	PIN : 3
2ème capteur de sortie : UltraSonic Sensor HC-SR04	PIN : 2
1er servo moteur : actionnant la Barriere d'entrée	PIN : 6
2ème servo moteur : actionnant la Barriere de sortie	PIN : 7
Afficheur LCD	I2C

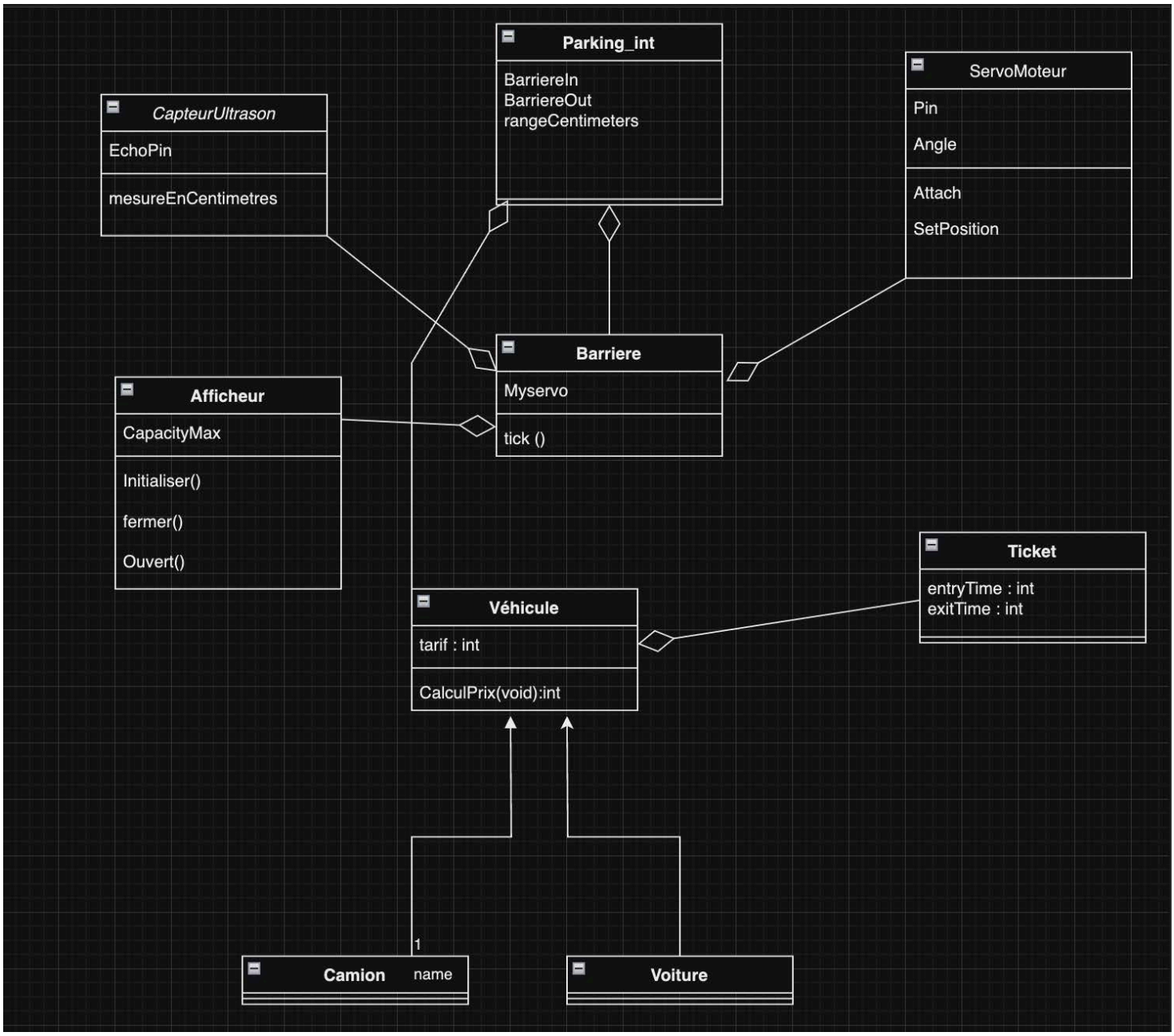
Notre prototype de parking intelligent a été concrétisé sur une breadboard, au centre de laquelle trône la carte Seeduino Lotus. D'un côté (côté 1) de la breadboard, nous avons agencé un écran LCD, un capteur ultrasons et un servomoteur. De l'autre côté (côté 2), nous avons positionné un capteur ultrasons et un servomoteur.

Le côté 1 constitue l'entrée de notre parking, où un utilisateur peut consulter le nombre de places disponibles sur l'afficheur LCD. Si des places sont libres, l'écran affiche le message « Soyez les bienvenus ». En revanche, en cas d'absence de places disponibles, le message « Revenez plus tard » s'affiche.

Lorsqu'il y a des places disponibles, en s'approchant de la barrière d'entrée, un capteur ultrasons détecte la présence d'un véhicule, ce qui permet à la barrière de commander le servomoteur pour autoriser l'accès et permettre au véhicule d'entrer. Si aucune place n'est disponible, même en présence d'un véhicule en attente, l'accès lui est refusé et il doit 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 sortir du parking, le capteur ultrasons détecte sa présence et envoie l'information à la barrière de sortie. Cette dernière commande le servomoteur pour permettre au véhicule de passer.

DIAGRAMME DE CLASSE :



Le schéma de classes que nous avons élaboré comprend diverses entités telles que Parking_int, Afficheur, CapteurUltrasons, ServoMoteur, Véhicule, Camion, Voiture et Ticket. Les classes Camion et Voiture héritent de la classe Véhicule, bénéficiant ainsi de ses propriétés. Le code intègre des attributs et des méthodes cruciaux pour le projet, comme la méthode initialize() de la classe Afficheur, qui affiche des messages à l'écran, et la méthode mesureEnCentimetres() de la classe CapteurUltrason, déterminant la présence d'un véhicule par un retour true ou false en fonction de la distance mesurée.

Pour optimiser notre programme, nous envisageons d'ajouter des capteurs supplémentaires au-dessus des CapteursUltrasons existants. Cette modification permettrait de différencier un camion d'une voiture, par exemple, en détectant la voiture uniquement avec le capteur inférieur et le camion avec les deux capteurs. De plus, afin de rendre le parking payant, nous pourrions introduire une classe Ticket qui enregistre le temps de stationnement des véhicules pour calculer le coût d'utilisation.

Parmi les difficultés majeures rencontrées lors du projet, la limitation du nombre de broches disponibles sur la carte a été un obstacle majeur, nous empêchant d'ajouter les deux capteurs à ultrasons nécessaires pour la distinction des types de véhicules. De plus, le manque de temps a constitué un défi significatif, en particulier lors de l'utilisation de la carte Arduino Seeeduino Lotus dans les derniers jours du projet.

CONCLUSION :

Le bureau d'études a été d'une aide précieuse pour notre équipe. Il a facilité notre compréhension des concepts enseignés en cours et nous a permis de les mettre en pratique. De plus, il a comblé certaines lacunes en matière de codage et nous a enseigné l'utilisation de l'IDE d'Arduino.

Cette expérience a également été l'occasion d'exprimer notre créativité et notre esprit innovant. Comme mentionné précédemment, les contraintes technologiques et logistiques ont limité l'intégration de plusieurs fonctionnalités potentielles dans notre projet.

Travailler de manière autonome et avec une totale liberté nous a donné l'opportunité de rédiger notre propre cahier des charges, d'établir un planning de travail et de maîtriser les différentes phases de gestion de projet.