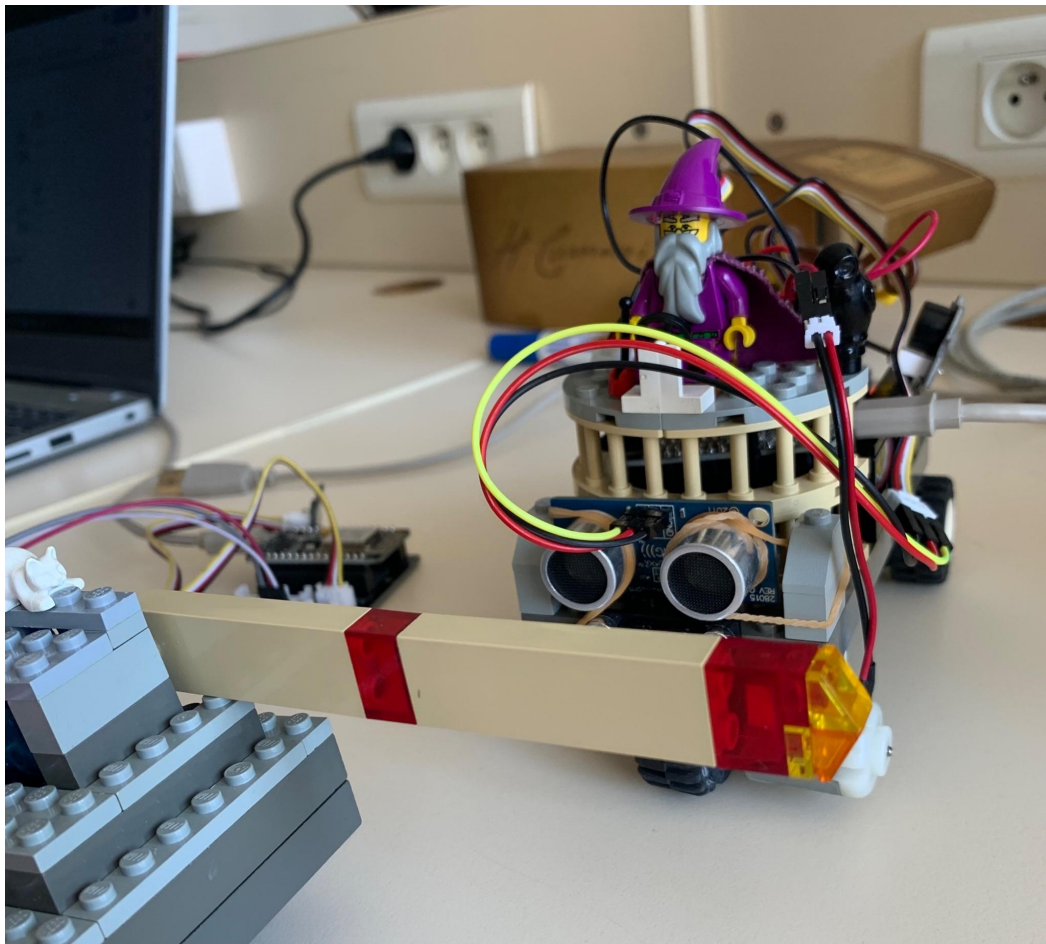


BE C++ - Voiture et péage autonomes

2021-2022

Nidegger Jean-Louis, Drezet Jérémy



Introduction	1
Diagramme de classe	2
Schéma de fonctionnement matériel et logiciel	3
Partie matérielle	3
Partie logicielle	3
Conclusion	4

Introduction

Nous avons choisi d'articuler notre projet dans le langage c++ autour d'une idée d'actualité qu'est le développement d'un véhicule autonome, accompagné d'un péage lui aussi automatisé. Ce sujet au centre des avancées technologiques futures nous tenait à cœur et l'application nous a paru réalisable à partir du matériel Arduino que nous avons à disposition.

La problématique de notre projet était d'arriver à coordonner deux systèmes distincts que sont notre véhicule et notre barrière. Pour se faire, nous avons ainsi utilisé deux cartes arduino esp8266 mises à notre disposition. La voiture a pour objectif de passer un péage en payant avec l'argent qu'elle possède. Elle communique sa quantité d'argent à l'aide du buzzer qui indique au micro de la barrière qu'elle peut payer (3 euros pour passer ce péage). Une fois que la transaction est validée, la barrière s'ouvre et la voiture redémarre étant donné que son moteur DC régulé par un relais intégré ne se déclenche que lorsqu'il n'y a pas d'obstacle détecté par le capteur à ultrason placé à l'avant du véhicule. La barrière se referme automatiquement au bout de 5 secondes après son ouverture, laissant largement le temps à la voiture de passer. Notons que nous avons utilisé des LEGOs afin de construire la structure nous permettant de créer nos éléments.

Nous allons pouvoir à présent vous présenter dans les parties suivantes l'organisation concrète de notre projet à travers différents diagrammes et schémas appropriés.

1. Diagramme de classe

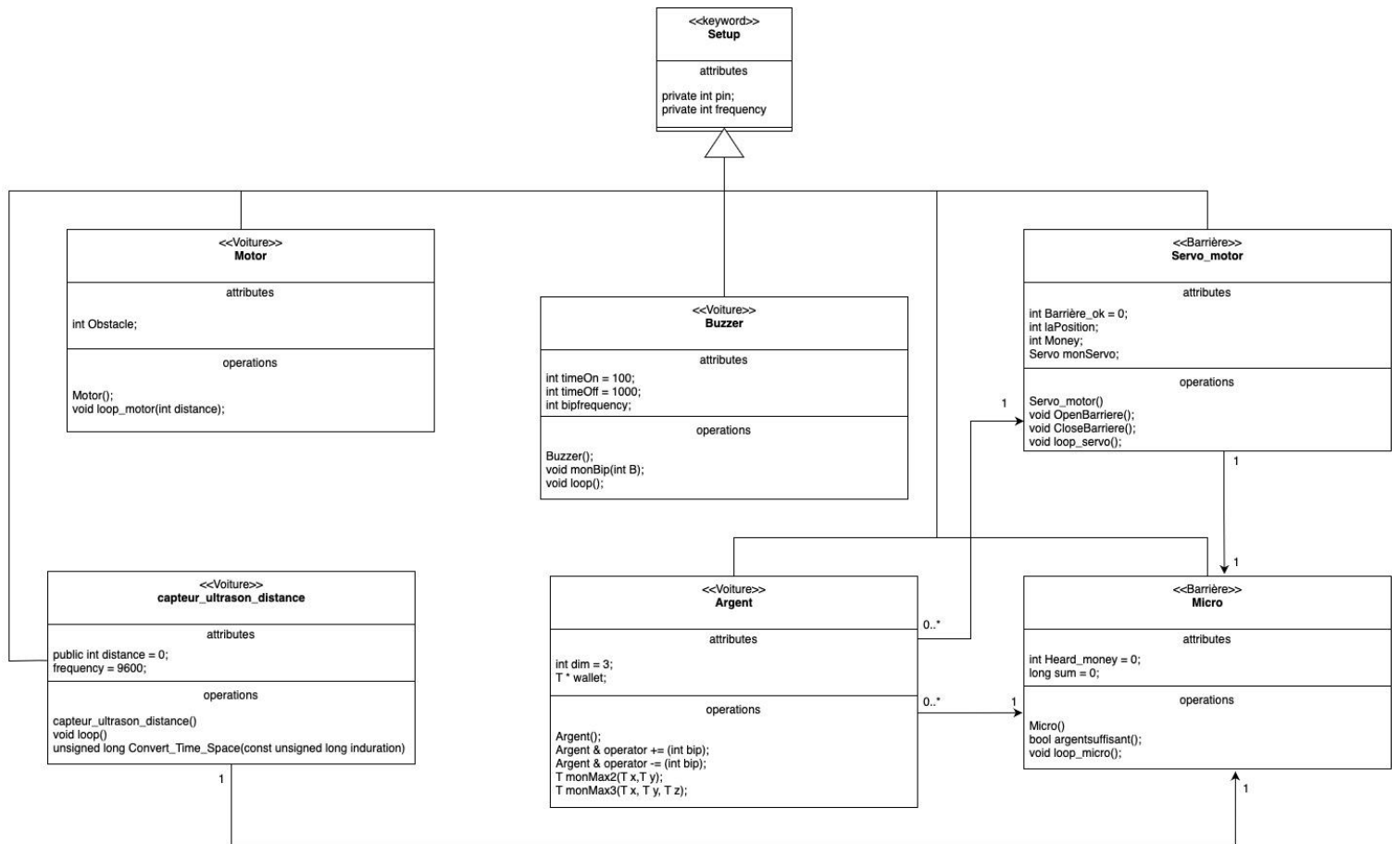


Figure 1 : diagramme de classes du projet, classes mère et filles

2. Schéma de fonctionnement matériel et logiciel

a. Partie matérielle

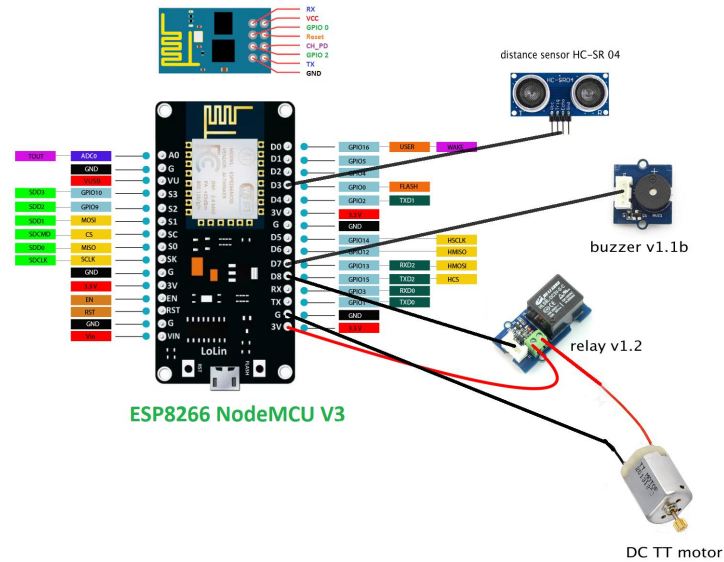


Figure 2 : carte esp8266 liée à l'entité Voiture

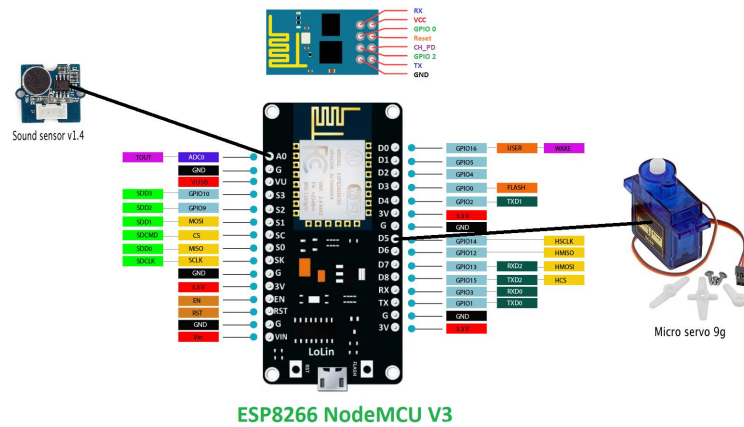


Figure 3 : carte esp8266 liée à l'entité Barrière

b. Partie logicielle

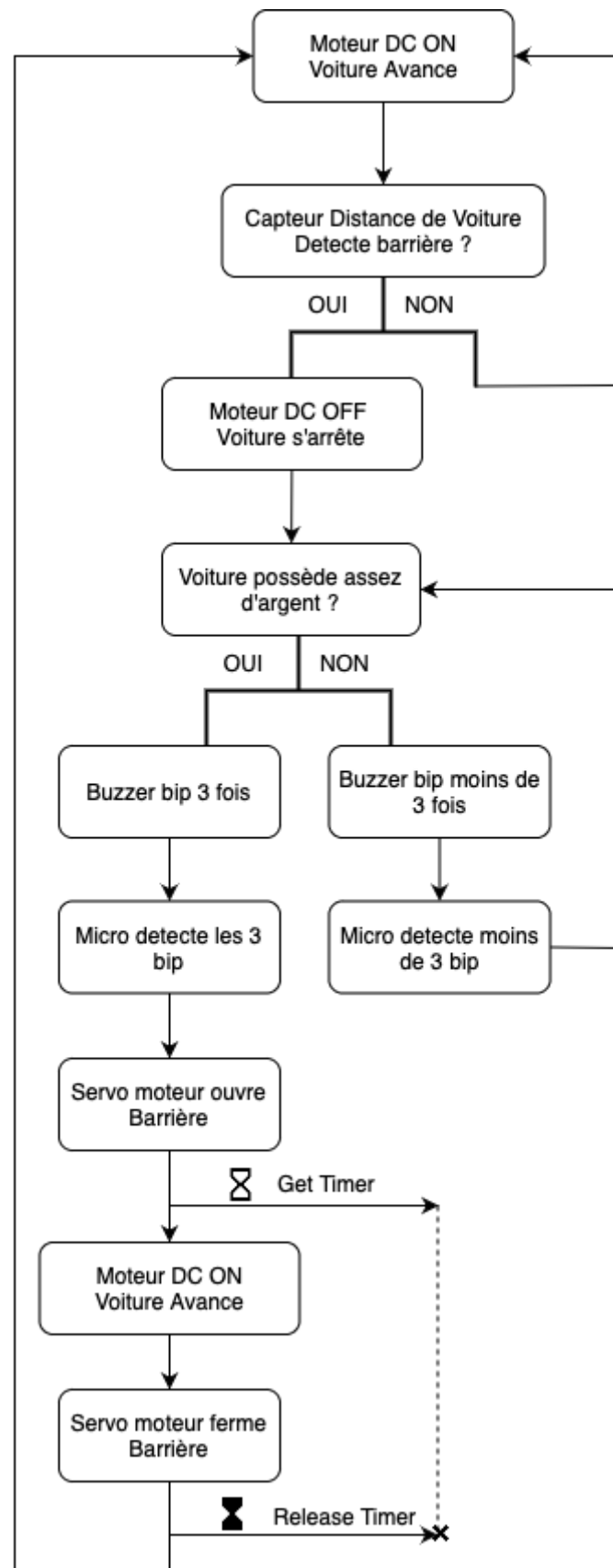


Figure 4 : schéma fonctionnel logiciel du système Voiture et Barrière autonome

Conclusion

Ainsi nous avons atteint l'objectif principal de notre application qui était de faire passer un péage à un véhicule de manière autonome. Cela n'a pas été sans mal notamment lors du réglage des capteurs buzzer et micro : en effet leur synchronisation afin de bien capter les sons émis sans pour autant prendre les bruits parasites extérieurs ne fut pas une mince affaire. Il nous a également fallu un peu de temps afin de comprendre parfaitement comment faire passer nos classes de capteurs et actionneurs en bibliothèques Arduino sur nos machines. De plus, certaines variables nous ont posé problème lorsqu'elles devaient être accessibles par plusieurs classes à la fois notamment pour leur gestion et l'évolution de leur valeur.

L'avantage de notre projet est qu'il peut évoluer de multiples manières. On pourrait par exemple imaginer ajouter des leds à l'arrière et à l'avant du véhicule pour les phares, ou encore améliorer le système de mouvement pour permettre à la voiture de tourner, ou bien même ajouter un capteur de distance sur la barrière pour que cette dernière ne se ferme bien que lorsqu'elle détecte que le véhicule est parti.

C'était un projet très intéressant et constructif qui nous a permis de mettre en application directement nos compétences acquises lors des td et tp de c++ de ce semestre. C'est avec un léger pincement au cœur que nous avons démantelé nos créations au moment de rendre les capteurs utilisés.