الجـــــــــــــمـــهـــوريـــــــــــــة
الجـــزائـــريـــة
الديمـــــــــــــــــــــــقـــراطـيـــة الـــشـــــــــــــــــــــــــــعـــبـــيـــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التـــــــــــــــــــــــــعـــــليـــــم العـــــــــــــــــــــالي و
البـــــــــــــــــــــــــــحث العـــــــــــلـــمـــــــــي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جـــــــــــــــــــامعة أبي بـكـــــــــر بـــلـــقـــايـد
- تـــــلمســـــــان-
Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Automatique

Spécialité : Automatique et informatique industrielle

Par : Ghellai Omar Abderrezak Moncef
Krouchi Ahmed Choukri

Sujet

Pilotage d'un véhicule multitâche

Soutenu publiquement, le 27 / 06 / 2023 , devant le jury composé de :

Mr Benariba Hassane	MCB	Université de Tlemcen	Président
Mme Benallel Mounira	MCA	Université de Tlemcen	Encadrant
Mme Benmasour Souhila	MCB	Université de Tlemcen	Examinateur

Année universitaire : 2022 /2023

Dédicace

À nos familles, qui ont été notre soutien constant et notre source d'inspiration tout au long de ce parcours, nous dédions ce travail de mémoire. Vos encouragements, votre amour et votre soutien indéfectible ont été essentiels dans notre réussite.

À notre encadreur, ***Mme Benallel****, nous exprimons notre profonde gratitude pour sa précieuse guidance, son expertise et son dévouement tout au long de ce projet. Ses conseils éclairés et ses encouragements ont été d'une valeur inestimable pour notre développement académique et professionnel.*

À nos amis, qui ont partagé avec nous des moments inoubliables et ont été présents à chaque étape de ce parcours, nous vous remercions du fond du cœur. Votre soutien moral, votre collaboration et votre amitié ont rendu cette expérience encore plus enrichissante.

À notre classe de 2ème année Master Automatique, nous tenons à vous remercier pour cette ambiance de camaraderie, d'échange et de partage qui a marqué notre parcours. Vos discussions passionnantes, vos remises en question et vos contributions ont contribué à notre épanouissement académique et personnel.

Enfin, nous dédions ce travail à toutes les personnes qui ont contribué de près ou de loin à notre réussite, que ce soit par leurs conseils, leur soutien moral, leurs encouragements ou leur présence bienveillante. Vous avez tous joué un rôle important dans notre parcours et nous vous en sommes profondément reconnaissants.

Remerciements

Nous tenons d'abord à DIEU le tout-puissant, pour nous avoir donné la force et la patience nécessaires pour mener à bien ce modeste travail.

Nous tenons à exprimer notre profonde gratitude envers notre promoteur qui nous a encadrés tout au long de la réalisation de notre projet de fin d'études. Nous le remercions pour le temps qu'il nous a accordé, ses précieuses directives et la qualité de son suivi tout au long de cette période.

Nous adressons nos remerciements chaleureux aux membres du jury d'évaluation pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre travail.

Nos remerciements vont également à tous les enseignants qui ont contribué à notre formation et à notre développement académique.

Enfin, nous tenons à exprimer notre gratitude envers toutes les personnes qui ont apporté leur contribution, de près ou de loin, à la réalisation de ce travail.

Résumé

Le travail présenté dans ce mémoire est la conception et le contrôle d'un robot mobile multitâches, doté de plusieurs modes de fonctionnement. En combinant nos connaissances en automatisation, robotique mobile, systèmes embarqués et programmation Arduino, nous avons développé un robot autonome capable de suivre des lignes, d'éviter les obstacles, de se garer de manière autonome, et offrant également la possibilité d'être piloté manuellement grâce à la technologie Bluetooth et infrarouge, permettant à l'utilisateur de contrôler le robot . Notre objectif était de fournir des solutions répondant à des besoins variés, à la fois pratiques et divertissantes.
Mots clés : robot mobile, Arduino, véhicule autonome, suiveur de ligne, évitement d'obstacle, mode Bluetooth, mode Infrarouge, parking autonome.

Abstract

The work presented in this thesis is the design and control of a multitasking mobile robot equipped with multiple operating modes. By combining our knowledge in automation, mobile robotics, embedded systems, and Arduino programming, we have developed an autonomous robot capable of line following, obstacle avoidance, autonomous parking, and also offering the possibility of manual control using Bluetooth and infrared technology, allowing the user to control the robot. Our goal was to provide solutions that meet diverse needs, both practical and entertaining.
Keywords : mobile robot, Arduino, autonomous vehicle, line follower, obstacle avoidance, Bluetooth mode, Infrared mode, autonomous parking.

ملخص

يتمثل هدف هذه الرسالة في تصميم وتحكم في روبوت متحرك متعدد المهام، يتميز بوجود عدة أوضاع عمل مختلفة. من خلال دمج معرفتنا في مجال التحكم الآلي، والروبوتات المتحركة، والأنظمة المضمنة، وبرمجة Arduino ، قمنا بتطوير روبوت مستقل قادر على متابعة الخطوط، وتجنب العوائق، والوقوف ذاتيًا، ويوفر أيضًا إمكانية التحكم اليدوي عن طريق تقنية Bluetooth والأشعة تحت الحمراء، مما يتيح للمستخدم التحكم في الروبوت. كان هدفنا هو توفير حلول تلبي احتياجات متنوعة، سواء في الجانب العملي أو الترفيهي.

الكلمات الرئيسية : روبوت متنقل، Arduino ، مركبة مستقلة، متابعة الخط، تجنب العوائق، وضع Bluetooth ، وضع الأشعة تحت الحمراء، وضع الوقوف ذاتيًا.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Les avancées constantes dans les domaines de l'automatique et de la robotique ont ouvert un éventail de possibilités pour la conception de robots mobiles polyvalents. Ces machines sont capables d'accomplir diverses tâches dans des environnements variés, suscitant un intérêt croissant pour leur développement. Cependant, la réalisation réussie d'un robot mobile multitâches fonctionnel nécessite une compréhension approfondie des concepts d'automatique, de robotique et de systèmes embarqués, ainsi qu'une maîtrise des méthodes de modélisation, de contrôle et d'intégration des fonctionnalités. Notre recherche sur la création d'un robot mobile multitâches pourrait apporter des contributions significatives en ouvrant de nouvelles perspectives de recherche, en repoussant les limites technologiques de la robotique mobile et en enrichissant les domaines de l'automatique et de la robotique grâce à l'introduction de concepts novateurs de coordination, de contrôle et de planification.
À cette fin, notre étude suit une trajectoire méthodique qui permet une progression cohérente et systématique dans la compréhension et la mise en œuvre du robot mobile multitâches. Nous commençons par explorer les concepts fondamentaux de l'automatique et de la robotique, puis nous passons en revue les différentes classifications robotiques pertinentes. Ensuite, nous plongeons dans l'univers de la robotique mobile, en décortiquant les éléments clés de sa conception. Les étapes cruciales de notre cheminement comprennent la modélisation cinématique et dynamique ainsi que les systèmes embarqués. Enfin, nous examinons en détail les différents modes opérationnels du robot, en les analysant sous différentes perspectives pour une compréhension exhaustive. Cette approche méthodique a abouti à la création d'un robot multifonctionnel qui interagit de manière efficace avec son environnement.

Chapitre 1

Automatique et Robotique

1.1 Introduction

L'Automatique est une discipline fondamentale présente dans de très nombreux domaines d'application tels que l'automobile, l'aéronautique et l'aérospatiale, les usines de fabrication, la production et distribution d'énergie électrique, de chauffage, de ventilation et de climatisation, la production de produits chimiques, le papier, l'agro-alimentaire et les métaux, la robotique, les chaînes d'approvisionnement et de logistique, pour n'en citer que quelques-uns. Elle nous donne les bases scientifiques ainsi que la technologie pour analyser et concevoir des systèmes.
Cette discipline développe des méthodes et des outils pour la modélisation des systèmes dynamiques (physiques, chimiques, biologiques, économiques, sociaux) afin de les analyser et pour leur commande pour la réalisation des tâches et/ou d'optimisation de critères. Cette discipline est indispensable pour analyser, concevoir, simuler, optimiser, valider et vérifier les systèmes technologiques et socio-technologiques qui sont amenés dans la prochaine décennie à devenir de plus en plus interconnectés, avec un traitement d'énormes quantités de données et d'informations, et avec de nouvelles formes de synergie entre les humains et les systèmes technologiques .
Le concept de robot date de plusieurs siècles, mais le terme robot fut inventé par le tchèque Karel Capek dans une pièce de théâtre écrite en 1920 : "RUR ou les robots universels de Rossum". Ce terme est dérivé du verbe tchèque "rabota" signifiant travail forcé ou corvée.
En l'année 1959 Georges Devol invente une machine originale, polyvalente et reprogrammable, ce qui a permis au robot d'acquérir une réalité industrielle. A la fin des années 1970 les robots industriels de première génération ont vu le jour.
Le terme robot peut désigner une large variété de réalisations technologiques, allant du simple dispositif mécanique exécutant des mouvements répétés, aux machines analogues morphologiquement aux bras humains et possédant une certaine intelligence. La robotique moderne trouve application dans de nombreux domaines :

- La robotique industrielle
- La robotique de service
- La robotique médicale
- La robotique militaire
- La robotique scientifique, par exemple pour l'exploration de l'espace ou des fonds marins pour la recherche fondamentale (validation de nouveaux algorithmes), etc.
- La robotique de transport (de personnes et de marchandises)

La robotique industrielle est officiellement définie par l'Organisation Internationale de Normalisation (ISO) comme étant un système commandé automatiquement, multi-applicatif, reprogrammable, polyvalent, manipulateur et programmable sur trois axes ou plus. Les applications incluent les robots de soudage, de peinture et d'assemblage etc. ... (voir figure 1.1)

FIGURE 1.1 – Robot industriel

Les robots industriels sont très utilisés dans le secteur de l'automobile. Leur conception nécessite une grande maitrise du domaine de l'ingénierie. Ils ont d'abord été développés pour intervenir dans les milieux à risques (nucléaire, à forte corrosion...) puis dans l'automobile avec le robot Unimate, le premier robot industriel de la société américaine Unimation, au début des années 60.Ils servent aussi beaucoup dans le maniement d'objets lourds, ainsi que l'assemblage de précision sur des petites séries.
L'avantage de la robotique industrielle est sa rapidité d'exécution et sa précision ainsi que la répétition de cette précision dans le temps. L'automatique et la robotique sont deux domaines techniques qui ont connu une croissance exponentielle au cours des dernières décennies grâce aux avancées technologiques des capteurs, des actionneurs, des microcontrôleurs, des systèmes embarqués et des logiciels de contrôle.
Les deux sont étroitement liés, en effet les robots sont souvent des systèmes automatisés complexes qui utilisent des capteurs, des actionneurs et des algorithmes de contrôle pour agir de manière autonome. De plus, les progrès de la robotique ont conduit à de nouveaux développements dans l'automatique, tels que les Algorithmes.

1.2 Historique

L'histoire de l'automatique et de la robotique remonte aux temps anciens où de simples Machines ont été créées pour effectuer des taches répétitives ou dangereuses. Cependant, des progrès significatifs dans ces domaines n'ont commencé qu'au XIXe siècle avec l'invention de la machine à vapeur et l'introduction de l'automatique dans les processus de fabrication.
Au XXe siècle, le développement de l´électronique, des technologies de l'information et de la cybernétique a conduit à des avancées significatives en matière d'automatique et de robotique. Les premiers systèmes de contrôle automatique ont étaient développés dans les années 30 et les premiers robots

industriels ont été introduits dans les années 60 pour des applications aussi diverses que la médecine, l'exploration spatiale et l'observation.
Les premiers robots mobiles ont également été développés, permettant aux machines de se déplacer de manière autonome et d'interagir avec l'environnement. Les progrès de l'intelligence artificielle, de l'apprentissage automatique et de la vision par ordinateur ont permis le développement de robots de plus en plus intelligents et autonomes. Des robots humanoïdes ont également été développés pour offrir des moyens plus humains de se déplacer et d'interagir. A l'heure actuelle, on peut distinguer 3 générations de robots :

- **Le robot passif :** capable d'exécuter une tâche complexe de manière répétitive, sans modifications de l'environnement. De nombreux robots sont encore de cette génération.
- **Le robot actif :**Il a une image de son environnement et choisit le bon comportement (Différentes configurations sont prévues). Il peut se calibrer tout seul.
- **Le robot (intelligent) :**Le robot est capable d'établir des stratégies, ce qui fait appel à des capteurs sophistiqués, et souvent à l'intelligence artificielle

1.3 Utilité de l'automatique et de la robotique

L'automatisation et la robotique sont des domaines d'une grande utilité dans de nombreuses industries et domaines, permettant aux tâches d'être effectuées de manière plus efficace, précise et sûre que par un humain.
Une automatisation est une technique ou un ensemble de techniques ayant pour but de réduire ou de rendre inutile l'intervention d'opérateurs humains dans un processus où cette intervention était coutumière. Il n'y a évidemment pas automatisation lorsque l'opérateur humain est remplacé par la force animale, ni lorsqu'un processus artificiel est substitué à un processus naturel. L'automatisation désigne uniquement une transformation de processus exclusivement créés par l'homme : techniques ou ensemble de techniques. Elle tend donc à économiser l'intervention humaine sous toutes ses formes (apport d'énergie mis à part) :

- appréciation, mesure et surtout évaluation de grandeurs (substitution d'un automatisme aux perceptions sensorielles)
- décision simple à partir de critères (substitution d'un traitement d'information au jugement de l'intelligence)
- organisation, gestion, optimalisation (substitution d'un traitement d'information, de mémoires auxiliaires, de systèmes autodidactiques, au jugement de l'intelligence éduquée et assistée d'une documentation).

Ainsi, l'automatisation peut s'appliquer à des processus qui ne mettent en œuvre aucune énergie physique appréciable : détection ,contrôle et mesures ,calculs en temps réel.
L'Automatique a joué un rôle important dans pratiquement toutes les évolutions technologiques majeures jusqu'à aujourd'hui, de la machine à vapeur aux fusées, aux avions à haute performance, aux vaisseaux spatiaux, aux trains à grande vitesse, aux voitures (vertes), aux caméras numériques, aux téléphones intelligents, aux technologies de productions modernes, aux équipements médicaux, et bien d'autres [32] .
Les systèmes de contrôle embarqués interagissent avec des dispositifs physiques et les systèmes, petits et grands, allant du téléphone mobile aux automobiles, aux robots, jusqu'à l'ensemble des installations industrielles. Plus de 90% de l'unité centrale de traitement (CPU) sont occupés par des systèmes de contrôle embarqués. Récemment les systèmes de contrôle ont donc évolué vers les systèmes de contrôle embarqués et en réseau, appelés aussi "systèmes cyber-physiques", prenant ainsi en compte la forte interaction des transmissions et du traitement de l'information en temps réel avec le dispositif physique [13] .
Dans l'industrie, les systèmes de contrôle automatique améliorent la qualité et la Précision des pro-

cessus de production, réduisant les coûts et le risque d'erreur humaine. Les robots industriels sont capables d'effectuer une grande variété de tâches, de la manipulation de pièces à l'assemblage, au soudage, à la peinture et au contrôle de la qualité. Grâce à leur précision et leur efficacité, ils peuvent améliorer considérablement la productivité de entreprises et contribuer à leur compétitivité sur le marché.
Dans le domaine de la santé, les robots peuvent aider les chirurgiens à effectuer des Interventions chirurgicales complexes et délicates, réduisant ainsi le risque d'erreurs. Ils peuvent également être utilisés pour les soins à domicile des personnes âgées ou handicapées, en les aidant dans leur vie quotidienne.
Dans l'exploration spatiale, les robots autonomes sont des outils nécessaires pour explorer des environnements extrêmes tels que les planètes, les comètes et les astéroïdes.
Les robots peuvent être envoyés dans des endroits où il est impossible ou dangereux d'envoyer des humains et de collecter des données scientifiques précieuses pour comprendre l'univers.
Dans le domaine de sécurité et de la défense, ils peuvent être utilisés pour des missions de surveillance et de reconnaissance ainsi que pour la neutralisation d'engins explosifs. Ils peuvent également être utilisés dans des opérations de sauvetage pour localiser des personnes dans des environnements dangereux tels que : des bâtiments effondrés ou des zones sinistrées.

1.4 Conclusion

En résumé, l'automatique et la robotique sont deux domaines étroitement liés, ont considérablement évolué au cours des dernières décennies grâce aux avancées technologiques en informatique, en électronique et en mécanique. Les systèmes de Contrôles automatisés et les robots ont considérablement amélioré l'efficacité, la qualité, la sécurité et la flexibilité des processus de fabrication et des tâches répétitives tout en réduisant le temps, les coûts ainsi que les risques pour les travailleurs.
L'automatique des processus industriels permet la surveillance continue des paramètres critiques, la détection rapide des anomalies, la correction des erreurs en temps réel, l'optimisation des performances et la réduction des temps d'arrêt. Les robots, quant à eux, sont utilisés dans de nombreux domaines pour effectuer des tâches spécifiques, telles que, la production, maintenance, surveillance, logistique, soins médicaux et recherche scientifique.
Malgré les avantages importants de l'automatique et de la robotique, ces technologies soulèvent également des inquiétudes quant à leur impact sur les emplois, la sécurité et l'éthique.Il est donc important de développer ces technologies de manière responsable et d'assurer leur intégration harmonieuse dans les processus de production et dans la societé.
En résumé, l'automatique et la robotique sont en evolution et continueront de transformer notre monde dans les années à venir, offrant à la fois des opportunités et des défis pour les travailleurs, les entreprises et la société dans tout son ensemble.
C'est dans ce contexte que nous nous tournons désormais vers l'examen plus approfondi de la robotique mobile dans le chapitre suivant qui nous permettra de voir comment ces avancées dans les domaines de l'automatique et de la robotique.

Chapitre 2

Robotique mobile

2.1 Introduction

La robotique est un domaine qui a plusieurs classifications. Elle évolue constamment et combine différentes disciplines pour créer des robots intelligents et autonomes. Un domaine important au sein de la robotique est la robotique mobile, qui se concentre sur les robots qui peuvent se déplacer et interagir avec leur environnement, ce qui en fait un aspect crucial. En mettant l'accent sur les robots à roues, nous plongerons dans le monde de la robotique mobile tout au long de ce chapitre. Se déplacer à travers les roues confère à ces robots stabilité et maniabilité. Notre objectif en explorant la robotique mobile est de tirer parti des capacités de ces robots à roue capables de naviguer et d'interagir avec leur environnement.

2.2 Classification des robots

Il existe plusieurs façons de classifier les robots, en fonction de différents critères telles que la fonction, le degré d'autonomie, la cinématique, etc. Dans notre part, Il existe 03 types de robots :

Les manipulateurs

- Les robots suivent des trajectoires spécifiques dans l'espace en fonction de leurs contraintes physiques et géométriques, ainsi que de l'environnement qui les entoure.
- Les positions du robot sont souvent représentées par deux ou trois valeurs discrètes pour chaque axe.
- La commande du robot est séquentielle, ce qui signifie qu'elle est exécutée étape par étape en suivant une séquence prédéfinie d'instructions.

Les télémanipulateurs

Ce sont des équipements de manipulation à distance tels que les pelles mécaniques et les ponts roulants, qui ont été introduits aux États-Unis vers 1945.

- Ils offrent la possibilité de déplacer les objets dans l'espace selon des trajectoires arbitraires.
- Les trajectoires sont définies en temps réel par l'opérateur à partir d'un pupitre de commande équipé d'un joystick.

Les robots manipulateurs industriels

Ils ont pour mission de charger divers éléments, notamment :

- **Des pièces** : stockage et déstockage, palettisation et dépalettisation, chargement et déchargement de machines-outils, manipulation d'éprouvettes, assemblage de pièces, etc.
- **Des outils** : soudage en continu ou par points, peinture, collage, ébavurage, etc.

Les robots didactiques

Ils sont des versions réduites des robots industriels mentionnés précédemment. Ils sont fabriqués par des constructeurs différents et utilisent une technologie distincte. Leur fonction principale est de servir comme outils de formation et d'enseignement, mais ils peuvent également être utilisés pour effectuer des tests de faisabilité pour des postes de travail robotisés.

Les robots mobiles

Ils offrent des possibilités plus vastes en raison de leur capacité à se déplacer. Ils peuvent être utilisés dans des zones dangereuses telles que les environnements nucléaires, les incendies, les situations de sécurité civile et les opérations de déminage. Ils peuvent également être utilisés dans des endroits inaccessibles comme l'océanographie et l'espace. Ces types de robots requièrent des capteurs et des logiciels sophistiqués pour leur fonctionnement.[16],[27] et [23] . Ces derniers vont être étudiés plus en détails au cours des chapitres suivants.

2.3 Robotique mobile

2.3.1 Généralité sur la robotique mobile

2.3.2 Introduction

De nombreuses disciplines sont impliquées dans le domaine pluridisciplinaire de la robotique ; des thèmes tels que la mécanique, la mécatronique, l'électronique, l'automatique, l'informatique et l'intelligence artificielle. En général, les différentes définitions du terme robot gravitent autour de ceci :
Une machine dotée de capacités de perception, de décision et d'action est ce qu'on appelle un robot. Ce qui lui permet d'agir de façon autonome dans son environnement en fonction de la perception qu'il en a.(voir figure 2.1)

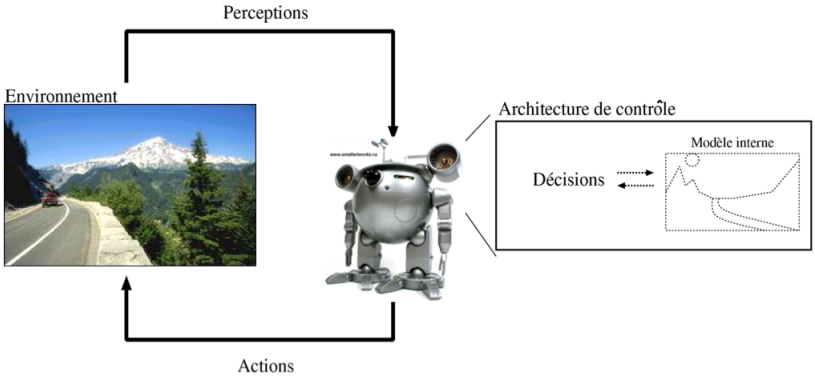

FIGURE 2.1 – Schéma des interactions d'un robot avec son environnement

2.3.3 Bref historique sur la robotique mobile

L'un des premiers robots mobiles autonomes a été la tortue construite par Grey dans les années 1950. Bien que Grey Walter n'utilise que quelques composants analogiques, y compris des tubes à vide, son robot est capable de se diriger vers une lumière qui marque un but, de s'arrêter face à des obstacles et de recharger ses batteries lorsqu'il arrive dans sa niche.
(voir figure 2.2) Les fonctions sont effectuées dans un environnement préparé, mais restent toujours des fonctions de bases pour les sujets de recherche à venir.
Avec l'apparition du transistor dans les années 60, les recherches en électronique ont conduit à des robots plus complexes, mais qui vont réaliser des tâches similaires. Ainsi le robot Beast (figure 2.3) de l'université John Hopkins peut se déplacer au centre des couloirs à l'aide de capteurs ultrasons, chercher des prises électriques (noires sur des murs blancs) grâce à des photo-diodes et de s'y recharger.

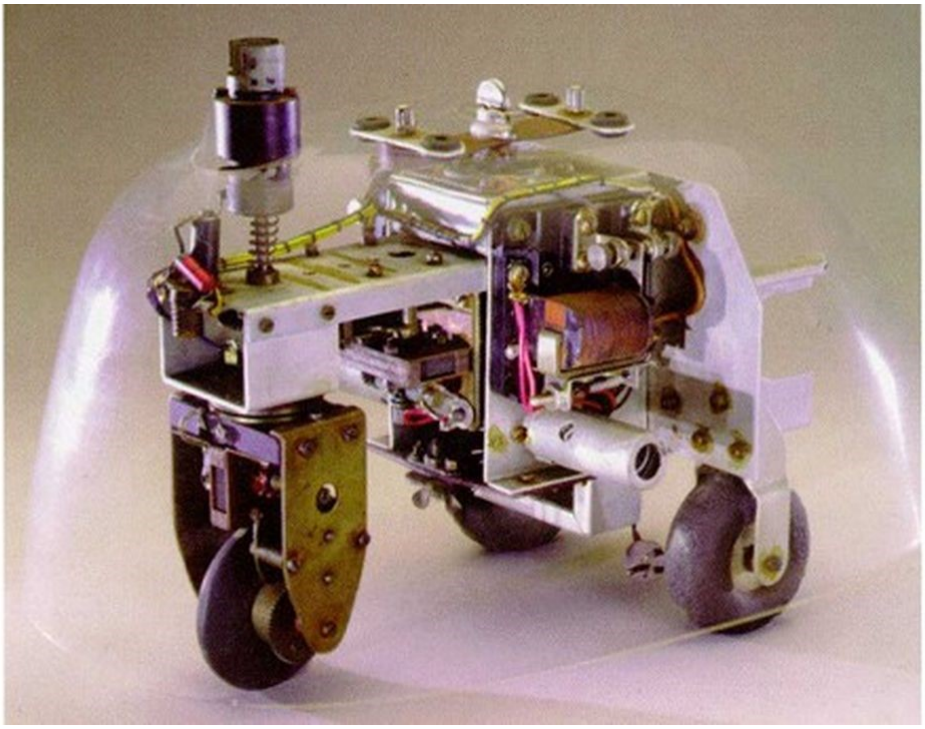

FIGURE 2.2 – La cyber tortue de William Grey Walter

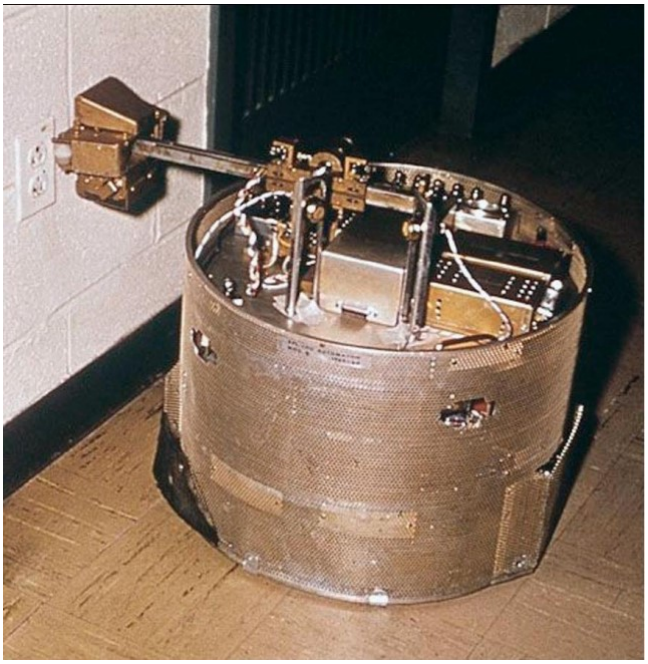

FIGURE 2.3 – The hopkins beast

Quelques années après, en 1969 ,les premières idées sur la recherche en intelligence artificielle dans la robotique mobile est apparu avec le robot Shakey qui utilise des télémètres et une caméra .Ce robots offre des approches symboliques de la planification , mais posait un problème sur la perception d'environnement .Cela a poussé les ingénieurs à poursuivre leurs recherches et développer de nouvelles techniques. Ainsi, dans les années 70 , apparaissent les premières utilisations de la stéréo-vision pour la détection d'obstacles et la modélisation de l'environnement avec par exemple le robot Hilare construit en 1977. (voir figure 2.4)

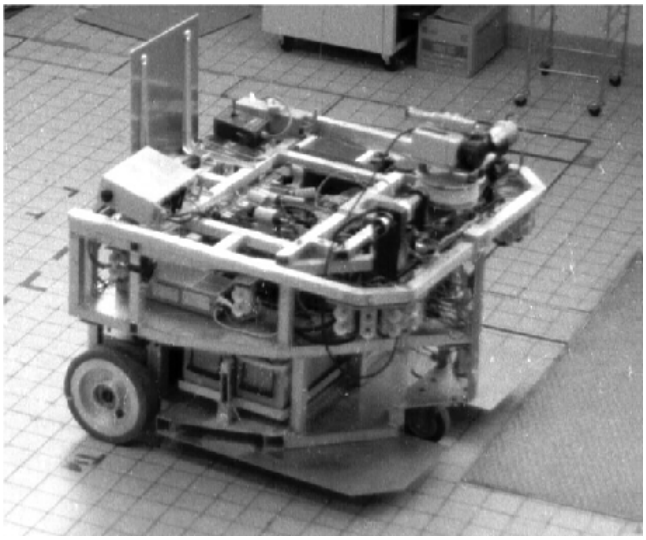

FIGURE 2.4 – Robot hillare

Dans les années 90, la robotique réactive a marqué une étape cruciale avec l'apparition de Rodney Brooks, l'un de ses représentants majeurs. Cette approche novatrice de la robotique, qui place la perception au cœur de la problématique, a permis de passer de grands robots très lents à de petits robots beaucoup plus réactifs et adaptés à leur environnement . Contrairement aux méthodes précédentes, ces robots n'ont pas besoin de modéliser leur environnement de manière complexe pour fonctionner de manière efficace. (voir figure 2.5)

FIGURE 2.5 – Genghis, développé par Rodney Brooks au MIT au début des années 90

Depuis les années 90, ces développements ont continué et ont abouti à l'apparition sur le marché de plates-formes intégrées telles que le Pioneer de la société Mobile Robots. Cela a permis à de nombreux laboratoires de se consacrer à la robotique mobile et a entraîné une diversification importante des thèmes de recherche. Même si les problèmes liés aux déplacements dans l'espace et à la modélisation de l'environnement restent complexes et cruciaux, certains laboratoires ont pu explorer des approches multi-robots, la problématique l'apprentissage ou encore les interactions entre les humains et les robots [22].

2.3.4 Domaine d'application de la robotique mobile

La robotique mobile est un domaine en constante évolution qui vise à développer des robots capables de se déplacer et d'interagir dans leur environnement. Les robots mobiles sont utilisés dans de nombreux domaines d'application, tels que l'industrie, la médecine, l'agriculture, la surveillance, la sécurité et l'exploration de l'espace. voici des exemples :

- **Industrie nucléaire**
  - surveillance de sites
  - manipulation de matériaux radioactifs
  - démantèlement de centrales
- **Sécurité civile**
  - neutralisation d'activité terroriste
  - déminage
  - pose d'explosif
  - surveillance de munitions
- **agriculture**
  - cueillette de fruits
  - traite, moisson, traitement des vignes.
- **industrie**
  - convoyage
  - surveillance

- **militaire**
  - surveillance
  - pose d'explosif
  - manipulation de munitions
- **Exploration spatiale**
  - exploration des environnements extraterrestres
  - collecte des échantillons
  - mener des expériences scientifiques [31].

2.3.5 Avantages dans l'utilisation des robot mobiles

La robotique mobile présente de nombreux avantages dans différents domaines. Voici quelques exemples :
Productivité accrue : les robots mobiles peuvent effectuer des tâches répétitives plus rapidement et plus efficacement que les humains, ce qui augmente la productivité et réduit les coûts.
Sécurité accrue : les robots mobiles peuvent être utilisés dans des environnements dangereux ou difficiles d'accès pour les humains, réduisant ainsi les risques pour la sécurité.
Flexibilité : les robots mobiles peuvent être programmés pour effectuer une variété de tâches différentes, ce qui les rend utiles dans de nombreux domaines différents.
Précision : les robots mobiles peuvent être équipés de capteurs avancés pour mesurer avec précision leur environnement et effectuer des tâches avec une grande précision.
Coût réduit : bien que les coûts initiaux d'un robot mobile puissent être élevés, leur utilisation à long terme peut réduire les coûts en réduisant les besoins en main-d'œuvre et en augmentant la productivité.
Amélioration de la qualité de vie : les robots mobiles peuvent être utilisés pour effectuer des tâches difficiles ou répétitives, ce qui permet aux humains de se concentrer sur des tâches plus intéressantes et créatives.
Dans l'ensemble, la robotique mobile offre de nombreux avantages qui peuvent améliorer notre vie quotidienne et transformer la façon dont nous interagissons avec notre environnement.

2.3.6 Architecture des robots mobiles

L'architecture des robots mobiles se structure en quatre éléments

La structure mécanique et la motricité

La structure mécanique se réfère à l'ensemble des éléments mécaniques du robot, tels que les moteurs, les roues, les bras mécaniques, les capteurs, etc. qui permettent au robot d'interagir avec son environnement. La motricité se réfère à la façon dont le robot se déplace dans son environnement, par exemple, en utilisant des roues, des chenilles, des jambes ou des ailes.

Les organes de sécurité

Les organes de sécurité des robots mobiles se réfèrent aux dispositifs de sécurité qui sont intégrés à la structure mécanique du robot et qui visent à protéger les personnes et les équipements dans son environnement. Les organes de sécurité sont essentiels pour garantir la sécurité de tous les acteurs impliqués, y compris les opérateurs humains, les passants et les autres machines.
Les organes de sécurité des robots mobiles peuvent comprendre des capteurs de proximité pour détecter la présence d'objets ou de personnes dans leur environnement immédiat, des barrières lumineuses

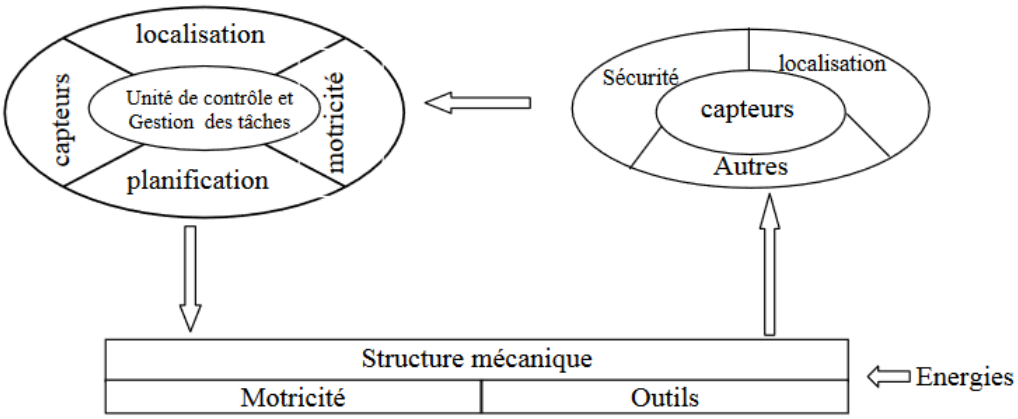

FIGURE 2.6 – Schéma représentant l'architecture des robots mobiles

pour délimiter des zones de sécurité, des systèmes de verrouillage de sécurité pour empêcher le mouvement du robot lorsqu'une porte est ouverte, ou encore des systèmes de désactivation d'urgence pour stopper le robot en cas de danger imminent.

Le système de traitement des informations et gestion des tâches

Le système de traitement des informations et gestion des tâches des robots mobiles est l'ensemble des composants électroniques et logiciels qui permettent de contrôler le mouvement et les actions du robot, de collecter et de traiter les données de l'environnement, et de planifier et d'exécuter les tâches assignées. Ce système tâches comprend généralement des capteurs pour détecter l'environnement, des actionneurs pour contrôler le mouvement du robot, des processeurs pour traiter les données et planifier les tâches, et des algorithmes pour prendre des décisions en temps réel.

Le système de localisation

Il permet au robot de connaître sa position et son orientation dans l'environnement, ce qui est essentiel pour naviguer, éviter les obstacles et effectuer des tâches précises.
Il existe plusieurs méthodes de localisation pour les robots mobiles, telles que la localisation basée sur les capteurs, la localisation basée sur les signaux radio et la localisation basée sur la vision.
Une fois que la position du robot est connue, elle peut être utilisée pour planifier des trajectoires de mouvement, éviter les obstacles et effectuer des tâches spécifiques. Les méthodes de localisation peuvent également être combinées pour améliorer la précision et la fiabilité du système de localisation [10].

2.3.7 Capteurs utilisés en robotique mobile

Les capteurs sont des dispositifs électroniques qui permettent aux robots mobiles de percevoir leur environnement. Ils sont utilisés pour capturer des données sur les caractéristiques physiques et les propriétés de l'environnement, telles que la lumière, le son, la température, la distance, la pression, etc. Les données collectées par les capteurs sont ensuite traitées et analysées par les algorithmes de perception afin de créer une représentation numérique de l'environnement, qui est utilisée pour planifier des trajectoires et prendre des décisions.
Les capteurs sont généralement classés en deux catégories en fonction de ce qu'ils mesurent : l'état

du robot lui-même ou l'état de son environnement. Dans le premier cas, semblable à la perception chez les êtres vivants, on parle de proprioception et donc de capteurs proprioceptifs. Ces derniers comprennent, par exemple, des capteurs de position ou de vitesse des roues ainsi que des capteurs de charge de la batterie. Les capteurs qui fournissent des informations sur l'état de l'environnement, c'est-à-dire sur ce qui est à l'extérieur du robot lui-même, sont appelés capteurs extéroceptifs. Il s'agit de capteurs qui mesurent la distance du robot à l'environnement, la température, qui signalent le contact du robot avec l'environnement, etc(voir figure 2.7) [14].

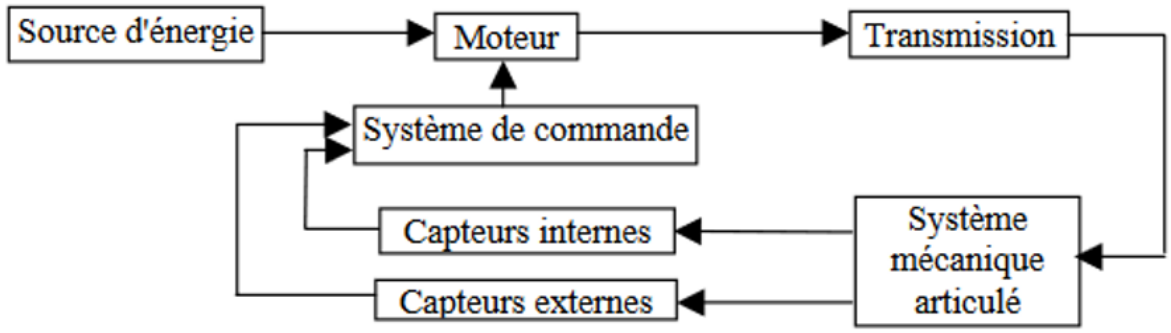

FIGURE 2.7 – Boucle de commande

1-Les capteurs proprioceptifs

Les capteurs proprioceptifs fournissent des informations sur les paramètres cinématiques du robot, tels que les positions, les vitesses, les accélérations, les angles et les forces. Ces capteurs sont situés à l'intérieur du corps du robot et mesurent les mouvements et les forces internes. D'autres types de capteurs doivent donc être utilisés conjointement pour obtenir une localisation plus précise et fiable [19].

Odométrie

L'odométrie permet d'estimer le déplacement de la plateforme en mesurant la rotation des roues (ou le déplacement des pattes). Cette mesure de rotation est généralement effectuée à l'aide d'un codeur optique placé sur l'axe de la roue ou sur le système de transmission (par exemple, à la sortie de la boîte de vitesses pour une voiture). Cependant, cette mesure est sujette à une incertitude importante, car l'estimation du déplacement dépend largement de la qualité du contact entre la roue (ou la patte) et le sol.

Les systèmes radar doppler et optiques

Une alternative à la mesure de déplacement via les roues est l'utilisation d'un radar dirigé vers le sol, qui permet de mesurer la vitesse du véhicule en utilisant l'effet Doppler. Des systèmes optiques, similaires à ceux des souris d'ordinateur, peuvent également être utilisés pour mesurer le déplacement du véhicule en analysant le mouvement relatif du sol. Ces méthodes sont plus précises que la mesure par les roues car elles ne sont pas affectées par les dérapages possibles des roues, mais elles sont généralement plus coûteuses et volumineuses et ne sont pas souvent utilisées sur les petites plates-formes [22].

Les capteurs d'attitude

Les capteurs d'attitude sont des dispositifs qui permettent d'estimer l'orientation du robot en mesurant les angles de cap, de roulis et de tangage. Ces capteurs sont souvent de type inertiel, ce qui signifie qu'ils mesurent les changements de vitesse et d'orientation à l'aide d'accéléromètres et de gyroscopes. Toutefois, ces capteurs peuvent être coûteux et sensibles aux interférences électromagnétiques et mécaniques, ce qui peut limiter leur utilisation dans des environnements hostiles. De ce fait, ils sont moins souvent intégrés dans les systèmes embarqués que les odomètres [31].

Les gyroscopes

Ils permettent de mesurer les variations angulaires. Ils peuvent être utiles en robotique mobile car ils peuvent compenser les erreurs des odomètres. Les erreurs d'orientation odométrique peuvent entraîner une erreur de position cumulative, qui peut être réduite, voire compensée, par l'utilisation de gyroscopes en combinaison avec les odomètres. Cependant, les gyroscopes très précis sont généralement trop coûteux pour être utilisés en robotique mobile. Cependant, les gyroscopes à fibre optique, qui sont connus pour leur grande précision, sont devenus une solution attrayante pour la navigation en robotique mobile car leur coût a diminué [31] et [17] .

Les gyromètres

Les gyroscopes sont des capteurs qui permettent de mesurer la vitesse angulaire d'un objet. Dans le contexte de la robotique mobile, deux technologies sont couramment utilisées : les gyromètres piézoélectriques et les gyromètres à fibres optiques. Les gyromètres piézoélectriques sont moins chers mais ont des dérives plus importantes et sont sensibles aux variations de température. Les gyromètres à fibres optiques sont plus précis et ont des dérives moins importantes, mais sont plus chers et ont une vitesse de fonctionnement minimum plus élevée. Dans tous les cas, les mesures d'orientation fournies par l'intégration des vitesses angulaires sont sujettes à des erreurs croissantes avec le temps, qui doivent être corrigées par des techniques de filtrage temporel.

Les gyrocompas

Le gyrocompas est un capteur qui permet de mesurer le cap. Il est composé d'un gyroscope et d'un compas magnétique. Le gyrocompas conserve le nord magnétique durant tout le déplacement du véhicule, après l'avoir initialement déterminé de façon autonome [31].

Le magnétomètre

Le magnétomètre, également connu sous le nom de compas magnétique, permet de mesurer l'orientation du robot en mesurant la direction du champ magnétique terrestre[35] .

Les accéléromètres

Ce sont capteurs inertiels mesurent l'accélération linéaire, c'est-à-dire le changement de vitesse par unité de temps, dans différentes directions. Les accéléromètres sont souvent utilisés pour détecter les mouvements, les inclinaisons et les changements de vitesse dans diverses applications, y compris la robotique, la navigation, les dispositifs portables, les véhicules, etc.

2- Les capteurs externes

Les capteurs extéroceptifs sont utilisés pour détecter l'environnement dans lequel évolue le robot. Ils sont souvent utilisés en complément des capteurs présentés précédemment. Des techniques de

fusion de données sont alors mises en œuvre pour traiter et combiner les informations sensorielles de natures différentes.
Ces capteurs sont utilisés pour différentes actions, notamment :

- Vérifier et améliorer la trajectoire suivie par le robot.
- Mesurer les interactions entre le robot et son environnement.
- Percevoir l'environnement dans lequel le robot évolue.
- Assurer la prévention et la sécurité du robot et des personnes autour [31].

2.1-Les télémètres

Il existe différents types de télémètres utilisant différents principes physiques pour mesurer la distance aux éléments de l'environnement. Parmi les principaux types de télémètres, on peut citer :

2.1.1 **Les télémètres à ultrasons :** Les télémètres à ultrasons sont des capteurs qui utilisent des ondes sonores de haute fréquence pour mesurer la distance entre un objet et le capteur. Le temps nécessaire pour que les ondes sonores se réfléchissent sur l'objet et reviennent au capteur est mesuré. Cette méthode est souvent utilisée pour les robots mobiles peu coûteux.(voir figure 2.8)

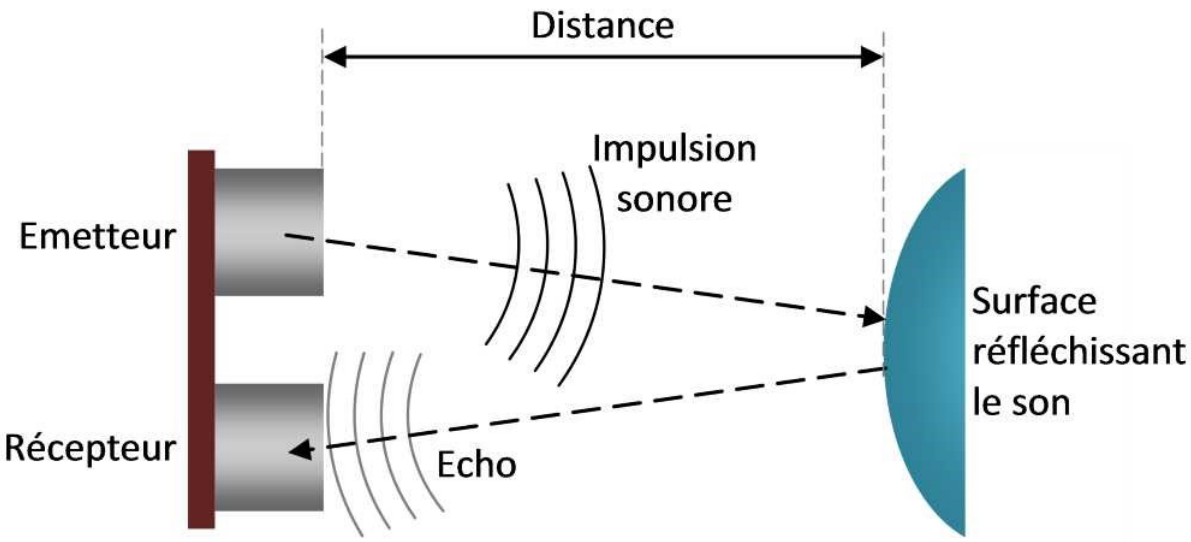

FIGURE 2.8 – Fonctionnement d'un capteur ultrason

2.1.2 **Les télémètres à infrarouge :**Ce type de télémètre est caractérisé par un faisceau de détection plus étroit que celui des télémètres à ultrasons. Les télémètres infrarouges utilisent une lumière infrarouge plutôt qu'une onde sonore pour la détection, ce qui leur permet de recueillir différentes informations en fonction des techniques utilisées.(voir figure 2.9)

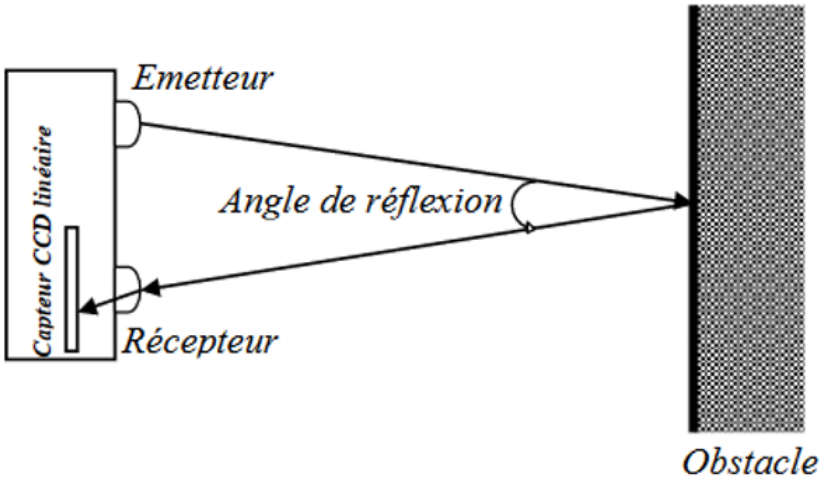

FIGURE 2.9 – Fonctionnement d'un capteur IR à mesure d'angle de réflexion

2.1.3 **Les télémètres Laser :**Les télémètres laser se servent d'un faisceau lumineux très fin et d'un temps de réflexion pour mesurer la distance d'un objet. Ils sont couramment utilisés dans des applications nécessitant une grande précision[6].
Il existe deux autres systèmes laser qui sont utilisés d'une manière différente :
Les capteurs goniométriques qui sont utilisés pour mesurer les angles[13] et les capteurs de source de lumière structurée sont souvent employés pour réaliser une modélisation 3D de l'environnement. Dans ce cas, une source laser est généralement couplée à une caméra pour fournir des informations sur la profondeur [24] .

2.1.4 **Les télémètres radar :** Les télémètres radar mesurent la distance entre un objet et le télémètre en utilisant des ondes électromagnétiques. Ils sont adaptés pour des applications nécessitant des mesures de distances longue portée ou dans des environnements difficiles.

2.2-les caméras

L'utilisation de caméras pour la perception de l'environnement est une approche prometteuse en robotique, car elle fournit une grande quantité d'informations similaires à celles utilisées par les humains. Cependant, le traitement de données complexes et volumineuses issues de ces capteurs peut être difficile. Cette méthode est donc une voie de recherche explorée et étudiée de près pour ses avantages potentiels en robotique. Ils existent les différentes techniques et méthodes utilisées pour traiter les données des caméras en robotique et leur application dans diverses applications robotiques.

2.2.1 **Les caméras simple** : Les caméras simples peuvent être utilisées pour détecter des points de repère, des guides de navigation et pour caractériser une position ou un point de vue dans l'environnement. Le flot optique permet d'estimer la distance des objets lorsque le robot est en mouvement, ce qui peut être utilisé pour réaliser un évitement d'obstacles ou une reconstruction tridimensionnelle de l'environnement.

2.2.2 **Les caméras stéréoscopiques :**Les caméras stéréoscopiques permettent d'estimer la distance des objets et d'obtenir une image de profondeur, mais cela dépend de la qualité de la texture et des conditions de luminosité.

2.2.3 **Les caméras panoramiques :**Les caméras panoramiques sont utiles pour la navigation, car elles permettent de caractériser une position indépendamment de la direction du robot. Elles peuvent également être utilisées pour détecter des points de repère et pour estimer le

flot optique.

2.3-Autres capteurs

2.3.1 **Les capteurs tactiles :**Les robots peuvent être pourvus de capteurs tactiles pour des situations d'urgence lorsqu'ils rencontrent un obstacle non détecté par le reste du système de perception. Les capteurs peuvent être des contacteurs simples situés sur le pourtour du robot, ne détectant le contact qu'au dernier moment. Alternativement, de petites tiges arquées peuvent être utilisées pour servir d'intermédiaire à ces contacteurs, ce qui permet une détection plus précoce et une marge supplémentaire pour arrêter le robot [22].

2.3.2 **Les balises :** Les balises sont des dispositifs émetteurs de signaux radiofréquences qui permettent de localiser des objets ou des personnes équipés de récepteurs appropriés. Le principe de fonctionnement des balises repose sur l'émission de signaux radiofréquences qui sont captés par les récepteurs pour déterminer leur distance et leur direction par rapport aux balises. Ces données sont ensuite utilisées pour déterminer la position de l'objet ou de la personne équipée du récepteur.

2.3.3 **GPS :**Le système de positionnement par satellite appelé GPS (Global Positioning System) permet de localiser avec précision n'importe quelle position sur Terre ainsi que de connaître l'heure, et ce, en tout lieu et à tout moment. Ce système repose sur un réseau de satellites orbitant autour de la Terre, qui envoient des signaux radio captés par des récepteurs GPS situés sur le sol.

2.3.8 Classification des robots mobiles

Les robots mobiles peuvent être classés de différentes manières en fonction de leurs caractéristiques et de leurs applications. Voici quelques exemples de classifications possibles :

1. **Selon leur structure**

1.1 *Robots à roues :* Ils se déplacent sur des roues, souvent de manière autonome. Exemples : les robots aspirateurs, les robots de sécurité intérieure.(voir figure 2.10)

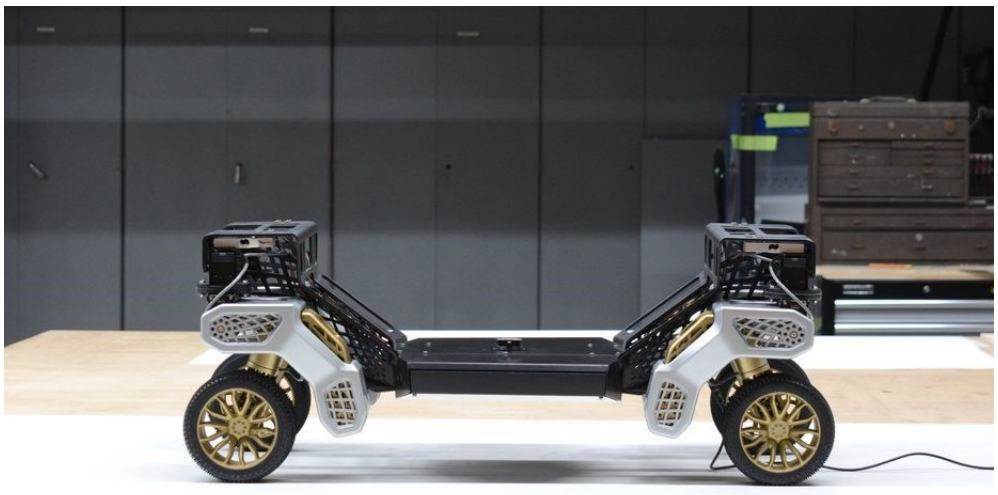

FIGURE 2.10 – Le robot mobile à roue Tiger

1.2 *Robots à chenille :* Ils ont une structure similaire à celle des tanks, avec des chenilles qui leur permettent de se déplacer dans des environnements difficiles. Exemples : les robots militaires, les robots d'exploration en terrain accidenté.(voir figure 2.11)

FIGURE 2.11 – Exemple d'un robot mobile à chenille

1.2 *Robots à pattes :* Ils imitent la marche des animaux pour se déplacer, souvent dans des environnements où les roues ou les chenilles seraient inefficaces. Exemples : les robots quadrupèdes, les robots arthropodes. (voir figure 2.12)

FIGURE 2.12 – Le robot dog

2. **Selon leur fonction**

2.1 *Robots de surveillance et de sécurité :* Ils sont utilisés pour surveiller des zones spécifiques, détecter des intrus, prévenir les incendies, etc. Exemples : les robots de sécurité intérieure, les drones de surveillance.

2.2 *Robots de livraison :* ils sont utilisés pour transporter des colis ou des marchandises d'un point à un autre. Exemples : les robots livreurs autonomes, les drones de livraison.

2.3 *Robots d'exploration :*Ils sont utilisés pour explorer des environnements difficiles ou dangereux, tels que des planètes extraterrestres ou des fonds marins. Exemples : les robots d'exploration

spatiale, les robots sous-marins.

2.4 *Robots de production :* ils sont utilisés dans les usines et les entrepôts pour effectuer des tâches de production ou de logistique. Exemples : les robots de soudage, les robots de palettisation.

3. **Selon leur niveau d'autonomie**

3.1 *Robots télé-opérés :* Ils sont contrôlés à distance par un opérateur humain.

3.2 *Robots semi-autonomes :* Ils peuvent effectuer certaines tâches de manière autonome, mais nécessitent encore une intervention humaine pour d'autres tâches.

3.3 *Robots autonomes :* Ils sont capables de se déplacer et de prendre des décisions de manière autonome, sans intervention humaine.(voir figure 2.13)

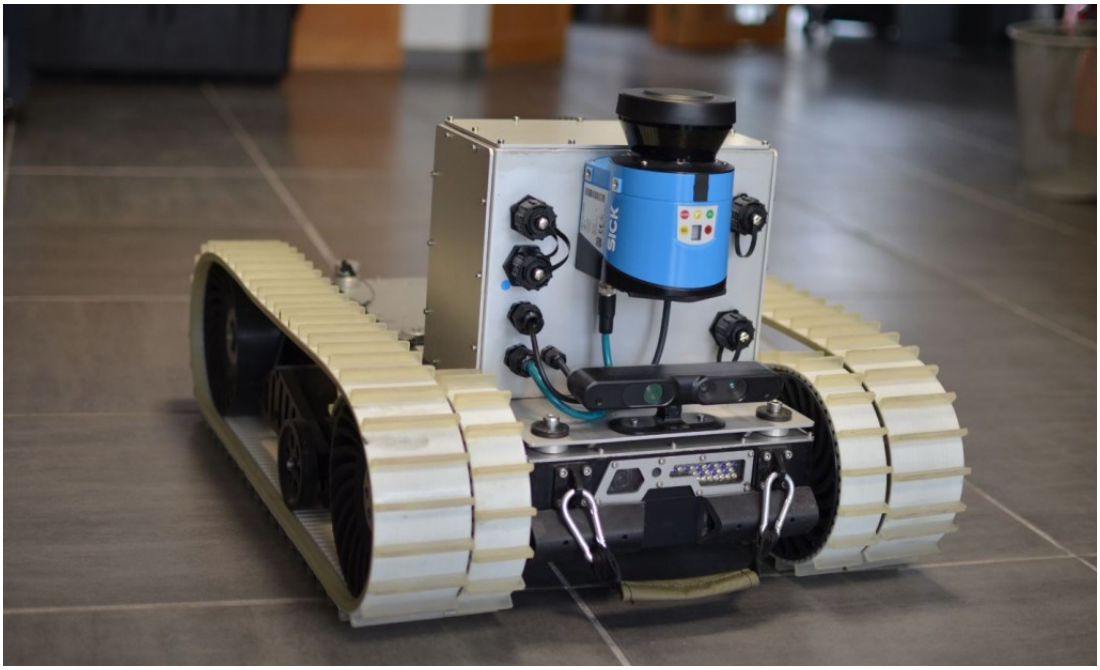

FIGURE 2.13 – Robot autonome

4. **L'holonomie et la non holonomie**
   Les systèmes mécaniques peuvent être soumis à des contraintes de position ou de vitesse, qualifiées de holonomes lorsque les relations entre les positions ou les vitesses des différents points du système peuvent être intégrées et doivent être respectées tout au long du mouvement, aboutissant à des relations algébriques liant les paramètres de configuration. Si ces relations peuvent être éliminées par un changement de variables approprié, le système est dit holonome. En revanche, si ces contraintes ne sont pas intégrables et ne peuvent être éliminées, le système est considéré comme non holonome [31] [34].
   Dans un scénario idéal, un robot peut se déplacer dans toutes les directions, ce qui signifie qu'il possède autant de degrés de liberté que de paramètres de configuration. Le nombre de degrés de liberté représente l'ensemble des mouvements que le robot peut effectuer[30].
5. **Robots à roues**
   - *Robot unicycle* : Le robot de type unicycle est équipé de deux roues motrices indépendantes et peut avoir des roues folles pour maintenir sa stabilité. Son centre de rotation se trouve entre les deux roues motrices. Il est considéré comme un robot non-holonome car il ne peut pas se déplacer dans une direction perpendiculaire à ses roues de locomotion. La

commande de ce robot est relativement simple car il peut être déplacé d'un point à un autre en effectuant des rotations et des déplacements en ligne droite. (voir figure 2.14)

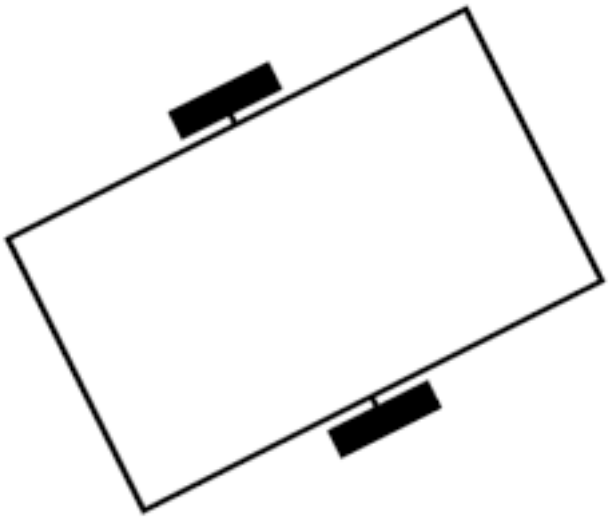

FIGURE 2.14 – Schéma d'un robot unicycle

- *Robot tricycle :* Un robot de type tricycle est composé de deux roues fixes situées sur un même axe et d'une roue orientable centrée placée sur l'axe longitudinal. Le mouvement du robot est déterminé par la vitesse des deux roues fixes et l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les deux roues fixes et de l'axe de la roue orientable. Ce type de robot est non-holonome, ce qui signifie qu'il ne peut pas se déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus complexe et il est souvent impossible d'effectuer des rotations simples en raison d'un rayon de braquage limité de la roue orientable. (voir figure 2.15)

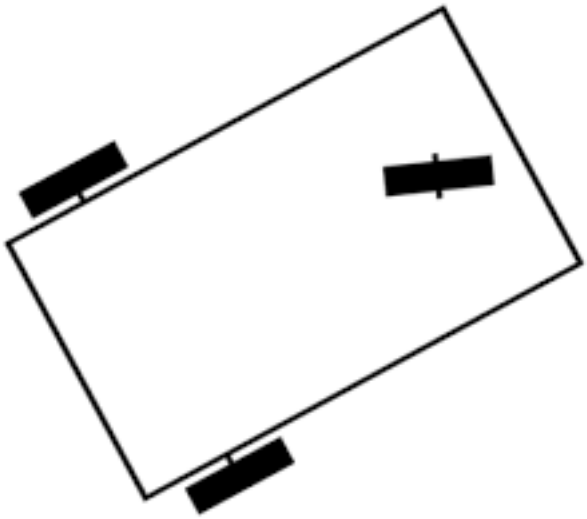

FIGURE 2.15 – Schéma d'un robot tricycle

- *Robot voiture :* Le robot de type voiture est similaire au robot tricycle, car il est composé de deux roues fixes placées sur un même axe et de deux roues centrales orientables placées sur un autre axe commun. Toutefois, il est plus stable grâce à un point d'appui supplémentaire. Toutes les autres caractéristiques du robot voiture sont les mêmes que celles du robot tricycle. Ce dernier peut être transformé en robot voiture en remplaçant simplement les deux roues avant par une seule roue centrale placée sur l'axe, de sorte que le centre de rotation reste inchangé. (voir figure 2.16)

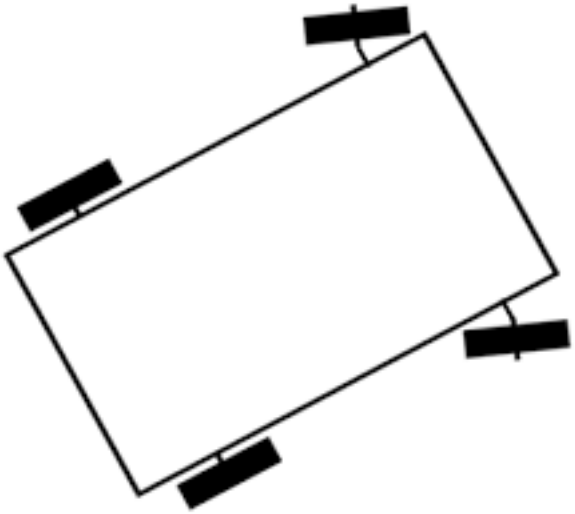

FIGURE 2.16 – Robot voiture

- *Robot omnidirectionnel :* Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral.
  L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande [25] et [28].(voir figure 2.17)

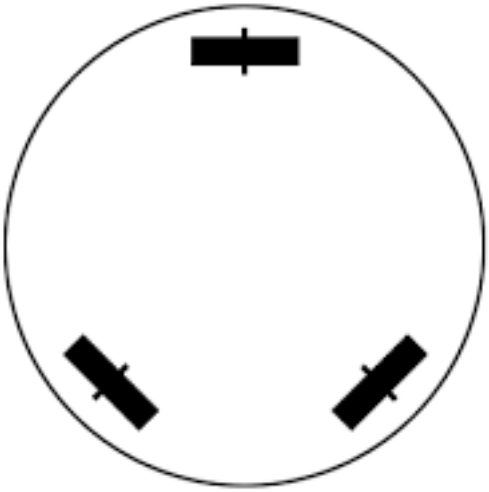

FIGURE 2.17 – Robot omnidirectionnel

2.3.9 Avantage et inconvénient de chaque type de robot

Type de robot	Avantages	Inconvénient
Unicyle	-Stable -Rotation sur soi-même -complexité mécanique faible	Non-holonome
Tricycle	-Complexité mécanique moderée	-Non-holonome -Peu stable -Pas de relation sur soi-même
voiture	-solide -complexité mécanique modérée	-Non-holonome -pas de relation sur soi-même
Omnidirectionnel	-Holonome -Stable -Relation sur soi-même	-Complexité mécanique importante

TABLE 2.1 – Avantage et inconvénient de chaque type de robot

2.4 Conclusion

La robotique mobile est un secteur industriel en pleine expansion qui intègre de multiples disciplines telles que l'ingénierie, l'informatique, les sciences cognitives et l'intelligence artificielle. Les robots mobiles ont la capacité de se déplacer et d'effectuer des tâches de manière autonome, ce qui permet une utilisation efficace dans divers secteurs. L'autonomie est un aspect clé de la robotique mobile, qui se définit par l'alimentation en énergie du robot. Bien que la robotique mobile soit encore en développement, les innovations en matière d'intelligence artificielle laissent présager une croissance exponentielle de son utilisation dans les années à venir. Les caractéristiques clés de la robotique mobile incluent la perception de l'environnement grâce aux capteurs, la capacité d'adaptation, la navigation, la planification et l'accomplissement de tâches autonomes. La coopération de multiples disciplines est essentielle pour relever les défis complexes de la robotique mobile.
La robotique mobile possède des caractéristiques clés, notamment la perception de l'environnement grâce aux capteurs, la capacité d'adaptation en cas de changement de situation, la navigation et la planification autonome, ainsi qu'un logiciel et une programmation orientés vers les tâches à accomplir.
Après avoir donné un aperçu sur la robotique mobile, nous allons dans le prochain chapitre passer à la modélisation, car cette dernière est une étape clé dans la compréhension et la réalisation de notre robot mobile.

Chapitre 3

Modélisation

3.1 Introduction

Nous allons considérer notre voiture comme un robot mobile à 4 roues fixes sans barre de directions non holonomes afin de le modéliser.
Dans le cas de notre modélisation nous avons opté pour le modèle d'un véhicule autonome à roues pour cela on a inclut la cinématique et la dynamique de notre voiture. Le mouvement du véhicule est décrit par le modèle cinématique sans prendre en compte les forces qui le causent. Cependant, les forces responsables du mouvement vont être prises en compte par le modèle dynamique.

3.2 Modèle cinématique

Nous allons présenter notre modèle cinématique. Tout mouvement ne peut être décrit sans un repère référentiel. La figure ci-dessous illustre le schéma du véhicule terrestre autonome. Il est possible de réaliser l'analyse cinématique d'un véhicule autonome à roues différentielles dans un plan en deux dimensions en utilisant des coordonnées cartésiennes.(voir figure 3.1)

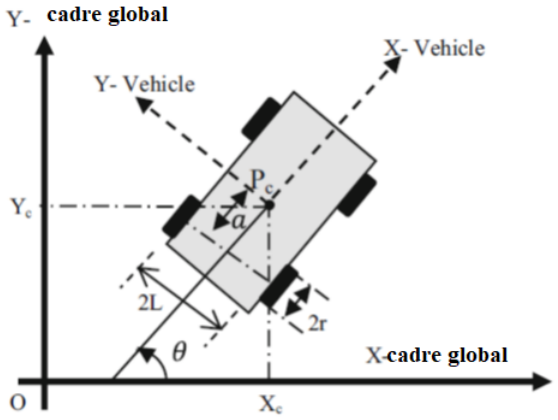

FIGURE 3.1 – Schéma de principe du véhicule

Dans notre modèle cinématique on suppose que le véhicule autonome se déplace sans glissement sur un plan, ce qui implique un contact pur de roulement entre les roues et le sol, ainsi qu'aucun glissement latéral entre la roue et le plan.
Le mouvement de direction à patinage entraîne d'une manière différentielle le véhicule qui possède quatre roues fixes standard. Les deux roues motrices sont entraînées indépendamment par deux moteurs pour acquérir le mouvement et l'orientation. Les roues ont le même rayon "r". Les roues motrices sont séparées par une distance "2L". La posture du véhicule dans le plan en deux dimensions à tout moment est définie par la situation dans les coordonnées cartésiennes et l'orientation par rapport à un cadre de référence global.
La configuration du véhicule est représentée par des coordonnées généralisées, soit :$Pc = (Xc, Yc, \Theta)$ comme indiqué dans la figure 3.1 . La cinématique du véhicule est donnée par les équations suivantes comme cité par Al-Mayyahi (2014)(2016) [12] et [11] :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\Theta} \end{pmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_L \end{bmatrix} \tag{3.1}$$

$$v = r.\left[\frac{\omega_r + \omega_L}{2}\right] \tag{3.2}$$

$$\dot{\Theta} = r.\left[\frac{\omega_r - \omega_L}{L}\right] \tag{3.3}$$

Les équations suivantes donnent la cinématique dans le cadre mondial :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\Theta} \end{pmatrix} = \begin{bmatrix} cos\Theta & 0 \\ sin\Theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.4}$$

$$\dot{x_c} = v.cos\Theta \tag{3.5}$$

$$\dot{y_c} = v.sin\Theta \tag{3.6}$$

$$\dot{\Theta} = \omega \tag{3.7}$$

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} \dot{\Theta}_r \\ \dot{\Theta}_L \end{bmatrix} \tag{3.8}$$

De plus, on a supposé que le mouvement du véhicule est soumis à des contraintes cinématiques qui impliquent que le contact entre les roues et le sol est un roulement pur et non glissant comme cité par Fierro and Lewis (1998) [21] et [12].
Cas 1 : Pas de contrainte de glissement

$$\dot{y_c}.cos\Theta + \dot{x_c}.sin\Theta = \alpha.\dot{\Theta} \tag{3.9}$$

Cas 2 :Pure contrainte de roulement

$$\dot{x_c}.cos\Theta + \dot{y_c}.sin\Theta + L\dot{\Theta} = r.\dot{\Theta}_r \tag{3.10}$$

$$\dot{x_c}.cos\Theta + \dot{y_c}.sin\Theta - L\dot{\Theta} = r.\dot{\Theta}_r \tag{3.11}$$

La forme suivante permet d'écrire les trois contraintes de non- holonomie en une représentation plus compacte à des fins de contrôle et de simulation :

$$J(q)\dot{q} = 0 \tag{3.12}$$

$$J(q) = \begin{bmatrix} -sin\theta & cos\theta & -\alpha & 0 & 0 \\ cos\theta & sin\Theta & L & -r & 0 \\ cos\theta & sin\theta & -L & 0 & -r \end{bmatrix} \tag{3.13}$$

$$\dot{q} = \begin{bmatrix} \dot{x}c & \dot{y}_c & \dot{\Theta} & \dot{\phi}_r & \dot{\phi}_L \end{bmatrix}^T \tag{3.14}$$

Dans cette transformation, nous essayons de trouver un moyen d'éliminer le terme de contrainte de l'équation. La matrice cinématique est définie par la transformation suivante :

$$\dot{q} = \begin{bmatrix} \dot{x}c \\ \dot{y}_c \\ \dot{\Theta} \\ \dot{\phi}_r \\ \dot{\phi}_L \end{bmatrix} = \begin{bmatrix} cos\Theta & -\alpha sin\Theta \\ sin\Theta & \alpha cos\Theta \\ 0 & 1 \\ \frac{1}{r} & \frac{L}{r} \\ \frac{1}{r} & -\frac{L}{r} \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{3.15}$$

avec :
V : vitesse du véhicule en mouvement [m/s]
Θ :orientation du véhicule en mouvement [degré]
ϕ_r :l'angle d'orientation de la roue droite
ϕ_L : l'angle d'orientation de la roue gauche
ω_r :vitesse angulaire de la roue droite [rad/s]
ω_L : vitesse angulaire de la roue gauche [rad/s]
ω : vitesse angulaire du véhicule [rad/s]
r : rayon de la roue [m]
a : distance entre le centre de masse et l'axe des roues motrices [m]
2L : distance entre chaque roue motrice et l'axe de symétrie du véhicule [m].
q : le vecteur de configuration
J : le Jacobien
Ce modèle est le modèle cinématique du véhicule qui décrit les vitesses mais pas les forces ou les couples qui ont un effet sur la vitesse. Dans la section suivante, le modèle dynamique sera présenté et analysé.

3.3 Le modèle dynamique

L'étude de la relation entre les différentes forces agissant sur un mécanisme robotique et leurs accélérations est représentée par le modèle dynamique d'un véhicule terrestre autonome.
Notre principale préoccupation dans la modélisation dynamique est la simulation et l'analyse de la conception du véhicule, ainsi que la conception d'un contrôleur de mouvement pour le véhicule. Pour cela le mécanisme de mouvement du robot est décrit en termes de ses composants : corps, joints et les paramètres qui les caractérisent.
Afin de définir le modèle dynamique d'un corps rigide donné, plusieurs paramètres sont nécessaires, tels que l'inertie, le centre de masse et les forces appliquées.
Il est possible d'utiliser l'approche lagrangienne basée sur l'énergie pour obtenir le modèle dynamique du véhicule autonome, qui se présente sous la forme générale suivante comme cité par Fierro and Lewis (1997) [11] et [20] :

$$M(q) + \dot{\omega} + C(q, \dot{q})\omega + F(\dot{q}) + G(q) = B(q)T \tag{3.16}$$

ou :
M(q) est la matrice d'inertie symétrique définie positive
$C(q, \dot{q})$est la matrice centrifuge et de Coriolis
$F(\dot{q})$est la matrice de frottement de surface
G(q)est le vecteur gravitationnel
B(q)est la matrice de transformation d'entrée
T est le vecteur d'entrée.
Le mouvement plan du véhicule conduit à l'élimination des termes de gravité dans l'équation dynamique :

$$\text{G(q) et } F(\dot{q})=0$$

Cependant, l'équation(1) peut être réécrite d'une autre façon :

$$\bar{M}(q)\dot{\omega} + \bar{C}(q, \dot{q})\omega = B(q)T \tag{3.17}$$

Les équations ci-dessus représentent les équations dynamiques du robot en prenant en compte les contraintes non holonomes, et peuvent être transformées en les équations matricielles simplifiées suivantes :

$$\begin{bmatrix} m & 0 \\ 0 & ma^2 + l_c \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \ddot{\Theta} \end{bmatrix} + \begin{bmatrix} 0 & -ma\dot{\Theta} \\ ma\dot{\Theta} & 0 \end{bmatrix} \begin{bmatrix} \omega \\ \dot{\Theta} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix} \begin{bmatrix} T_r \\ T_l \end{bmatrix} \tag{3.18}$$

Les paramètres physiques pertinents du véhicule autonome sont indiqués dans le Tableau suivant :

parametre	Description	Unité
r	Rayon des roues	m
L	La distance entre la roue motrice et l'axe de symétrie	m
a	La distance entre le centre de masse et l'axe de la roue motrice	m
m	La masse du véhicule avec les roues motrices et les moteurs DC	Kg
lc	*Le moment d'inertie de masse par rapport au centre de gravité*	Kgm^2
Tr,Tl	*Les couples du moteur de la roue droite et gauche*	Nm

TABLE 3.1 – Tableau les paramètres du véhicule terrestre

3.4 Conclusion

La modélisation cinématique et dynamique de notre robot à roues est cruciale pour notre projet. Ces modèles nous aident à analyser et prédire le comportement du robot en mouvement. La modélisation cinématique explique la relation entre les mouvements des roues et les déplacements du robot dans l'espace, tandis que la modélisation dynamique tient compte des forces et moments agissant sur le robot. Bien que ces modèles simplifient la réalité et nécessitent des approximations, ils fournissent une base solide pour optimiser les performances du robot. Ils guident la conception des algorithmes de contrôle et de navigation, assurant un fonctionnement efficace et précis du robot dans diverses situations.
Afin de maitriser le contrôle des robots, on va passer au prochain chapitre à la présentation des systèmes embarqués particulièrement les microcontrôleurs.

Chapitre 4

Système embarqué

4.1 Introduction

Le robot que nous utilisons est un exemple de système embarqué. Les systèmes embarqués sont des systèmes informatiques intégrés dans un appareil ou un équipement pour effectuer des tâches spécifiques. Ils fonctionnent de manière autonome et souvent en temps réel, avec des ressources limitées telles que la puissance de traitement, la mémoire et l'énergie. De nos jours, les systèmes embarqués se trouvent dans une grande variété d'appareils et d'équipements tels que les téléphones portables, les voitures et les équipements médicaux.

FIGURE 4.1 – Différents exemples des systèmes embarqués

Les microprocesseurs ou microcontrôleurs sont les éléments principaux des systèmes embarqués, assurant le contrôle des fonctions et tâches de l'appareil, avec la possibilité d'inclure des circuits électroniques, capteurs, interfaces utilisateur et logiciels. Ces systèmes sont omniprésents dans divers secteurs, tels que l'automobile, l'aérospatiale, l'électronique grand public, la défense et la sécurité, la santé, l'industrie et l'énergie, et ont un impact grandissant sur notre quotidien avec l'avènement de l'IoT et des appareils connectés. Leur conception, complexe, nécessite une connaissance approfondie des spécifications, exigences de performance et contraintes de conception pour assurer leur autonomie en temps réel malgré des ressources limitées.(voir figure 4.1)

4.2 Les microcontrôleurs

Un microcontrôleur est un circuit intégré qui contient les éléments fondamentaux d'un ordinateur tels qu'un processeur, des mémoires, des interfaces d'entrées-sorties et des unités périphériques. Les microcontrôleurs ont un degré d'intégration plus élevé, consomment moins d'énergie et ont une vitesse de traitement moins élevée (allant de quelques mégahertz à plus d'un gigahertz) et un coût inférieur aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels. Les microcontrôleurs permettent de réduire la taille, la consommation électrique et le coût des produits par rapport aux systèmes électroniques à base de microprocesseurs et d'autres composants séparés. Ainsi, ils ont favorisé l'usage de l'informatique dans un grand nombre de produits et de processus . [29]

4.2.1 Application des microcontrôleurs

Les domaines d'application des systèmes embarqués incluent :

- Informatique (souris, modems, etc.)
- Contrôle des processus industriels (régulation, pilotage, supervision, etc.)
- Appareils de mesure (affichage, calcul statistique, mémorisation, etc.)
- Automobile (ABS, injection, GPS, airbag, etc.)
- Multimédia (téléviseur, carte audio, carte vidéo, MP3, magnétoscope, etc.)
- Téléphonie (fax, téléphone portable, modem, etc.)
- Électroménager (lave-vaisselle, lave-linge, four à micro-ondes, etc.) [10]

4.2.2 Architecture du microcontrôleur

Les composants de base d'un microcontrôleur comprennent :

- **Unité centrale de traitement (CPU)** : c'est le cerveau de microcontrôleur son rôle est d'exécuter les instruction et les fonctions de programmations .
- **Mémoire** : il y a deux types de mémoires,La mémoire Morte (ROM),ou mémoire non volatile ,est une mémoire qui stocke les instruction de maniéré permanente y compris en cas de coupure de courant ,Et la mémoire vivre (RAM) qui stocke les fichiers de manière temporaire.
- **Convertisseurs Analogique-Numérique (CAN) et Numérique-Analogique (CNA) :**
  - **Convertisseurs Numérique-Analogique (CNA) :** est dispositif électronique (généralement un circuit intégré) permettent d'obtenir en sortie une tension dont la valeur est représentative du mot binaire présente en entrée .
  - **Convertisseurs Analogique-Numérique (CAN) :** est dispositif électronique (généralement un circuit intégré) permettent d'obtenir en sortie une grandeur numérique codée sur n bits dont la valeur est représentative de la grandeur analogique (tension) presenté en entrée .

  Les CAN et CNA permettent au microcontrôleur de communiquer avec des signaux analogiques, tels que les capteurs et les actionneurs.
- **Interfaces d'entrées/sorties (E/S) :**Les interfaces E/S sont utilisées pour interagir avec le monde extérieur. Les E/S comprennent des ports d'entrée/sortie généraux (GPIO) pour les signaux numériques, des ports série pour les communications série, des interfaces pour les bus de communication, tels que le SPI et l'I2C, et des interfaces pour les capteurs et les actionneurs spécifiques.
- **Horloge :** Son role est pour synchroniser les opérations du microcontrôleur et pour réguler la vitesse de la CPU.

4.2.3 Horloge

Dans le domaine de l'électronique et de l'informatique, une horloge désigne un oscillateur qui produit un signal périodique permettant la synchronisation des opérations au sein d'un système électronique ou informatique. Dans les microcontrôleurs, l'horloge interne joue un rôle important en régulant la vitesse de la CPU et des autres périphériques. La fréquence de l'horloge, mesurée en hertz (Hz), indique le nombre de cycles d'oscillation par seconde, et détermine la capacité du microcontrôleur à exécuter des instructions. Dans le contexte des systèmes embarqués, la régulation de l'horloge est cruciale pour réduire la consommation d'énergie et prolonger l'autonomie de la batterie. Certains microcontrôleurs disposent d'une horloge à basse consommation pour économiser l'énergie en mode veille ou inactif. En somme, l'horloge est un élément fondamental des microcontrôleurs, permettant de synchroniser les opérations et de réguler la vitesse de traitement, tout en offrant des options d'optimisation pour les systèmes embarqués.

4.3 Arduino

4.3.1 Définition du module Arduino

Le module Arduino est une plateforme de contrôle open source qui est construite autour d'un circuit imprimé dont les plans de conception sont publiés sous une licence libre. Bien que certains des composants de la carte, tels que le microcontrôleur et les composants complémentaires, ne soient pas en licence libre. Ce module est programmable et peut analyser et produire des signaux électriques pour effectuer une grande variété de tâches. Il est largement utilisé dans des domaines tels que l'électrotechnique industrielle et embarquée, le modélisme, la domotique, l'art contemporain, la robotique, la commande de moteurs, les jeux de lumière, la communication avec l'ordinateur, et la commande d'appareils mobiles. Chaque module d'Arduino est équipé d'un régulateur de tension de 5V et d'un oscillateur à quartz de 16 MHz (ou d'un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise le logiciel IDE Arduino.
Actuellement, il existe plus de 20 versions de module Arduino.Parmi ces types, nous avons choisi une carte Arduino UNO (voir figure 4.2) que nous étudierons un peu plus.
L'Arduino fournit un environnement de développement qui utilise des logiciels open source pour programmer le microcontrôleur de la carte. Il est facile de transférer un programme converti en code HEX dans la mémoire du microcontrôleur en utilisant une connexion USB. La carte Arduino est équipée d'un microcontrôleur ATmega 328 et d'autres composants, et dispose d'une mémoire morte de 1 kilo. Elle possède également 14 entrées/sorties digitales, dont 6 peuvent être utilisées pour la sortie PWM, ainsi que 6 entrées analogiques et un cristal de 16 MHz. La carte dispose également d'un port USB, d'un bouton de remise à zéro et d'une prise jack d'alimentation.
La communication entre le PC et la carte se fait via le port USB en installant le pilote approprié sur le PC [33].

4.3.2 Pourquoi choisir Arduino Uno

Parmi les nombreuses cartes électroniques disponibles pour la programmation électro-nique basée sur des microcontrôleurs ; Arduino UNO simplifie la façon de travailler avec les microcontrôleurs et offre de nombreux avantages. Tout d'abord, les cartes Arduino sont relativement peu coûteuses comparées aux autres plates-formes. De plus, le logiciel Arduino écrit en JAVA est multiplateforme

FIGURE 4.2 – Carte Arduino Uno

et peut fonctionner sous Windows, Macintosh et Linux, tandis que la plupart des systèmes à microcontrôleurs sont limités à Windows. L'environnement de programmation Arduino IDE est facile à utiliser pour les débutants, mais suffisamment flexible pour que les utilisateurs avancés puissent en tirer profit également. Le logiciel et le langage Arduino sont publiés sous licence open source, ce qui permet aux programmeurs expérimentés de les compléter et de les étendre. Les cartes Arduino sont basées sur des microcontrôleurs Atmel et leurs schémas de modules sont publiés sous une licence Creative Commons, permettant aux concepteurs de circuits expérimentés de réaliser leur propre version des cartes Arduino en les complétant et en les améliorant. Enfin, même les utilisateurs relativement inexpérimentés peuvent fabriquer leur propre version sur plaque d'essai pour économiser les coûts et mieux comprendre comment fonctionne la carte Arduino.(voir figure 4.2)

4.3.3 Description de la carte

Alimentation secteur

L'Arduino peut être alimenté soit via le port USB, soit via le connecteur externe. Si l'on utilise le connecteur externe, un régulateur de tension 5V est nécessaire pour stabiliser et réguler la tension à 5 volts, car la tension d'entrée peut varier entre 7 et 12 volts. Le régulateur de tension est plus volumineux que les autres composants, comme les CMS, et est utilisé comme dissipateur de chaleur.

Les broches d'entrée analogiques (A0 à A6)

Elles permettent de mesurer des tensions qui peuvent être utilisées dans les programmes. Elles ont une impédance d'entrée très élevée, ce qui signifie que le courant d'entrée est très faible.

Les broches numériques d'entrée/sortie (D0 à D13)

Elles peuvent être utilisées en tant que sorties pour envoyer des signaux à 5V pour un niveau logique '1' et 0V pour un niveau logique '0'. Chaque broche numérique peut supporter un courant maximal de 40mA sous une tension de 5V, ce qui est suffisant pour alimenter une diode, mais pas assez pour alimenter un moteur électrique. Les deux premières broches D0 et D1 sont réservées pour la communication avec l'USB RX/TX.

La liaison série UART

L'Arduino utilise un UART (Universally Asynchronous Receiver/Transmitter) présent sur la plupart des microcontrôleurs actuels, qui permet d'émettre et de recevoir des données à travers une liaison série aux niveaux TTL (Transistor-Transistor Logic). Cette terminologie fait référence à l'une des premières familles de circuits logiques. **Alimentation et régulateurs**

Il y a deux types de régulateurs sur la carte Arduino : un régulateur avec une sortie de 5 volts (MC33269) et un autre avec une sortie de 3,3 volts (LP2985-33DBVR). Pour que le régulateur 5 V (MC33269) fonctionne correctement et fournisse une sortie de 5 volts, l'alimentation doit être d'au moins 7 volts.Quant au composant LP2985-33DBVR, c'est est un régulateur de tension qui produit une sortie de 3,3 volts.La carte peut être alimenté à partir des entrées USB, Vin et connecteur jack.

Horloge, quartz

L'Arduino Uno utilise un oscillateur à quartz 16 MHz pour générer une horloge précise pour les opérations de calcul et de synchronisation de la communication série. Cet oscillateur est connecté à deux broches de la puce du microcontrôleur, qui sont utilisées pour réguler la vitesse d'exécution du code et la synchronisation des communications. Le quartz fournit une horloge stable et précise, ce qui est essentiel pour les applications nécessitant une synchronisation précise ou des calculs rapides et précis [4].(voir figure 4.3)

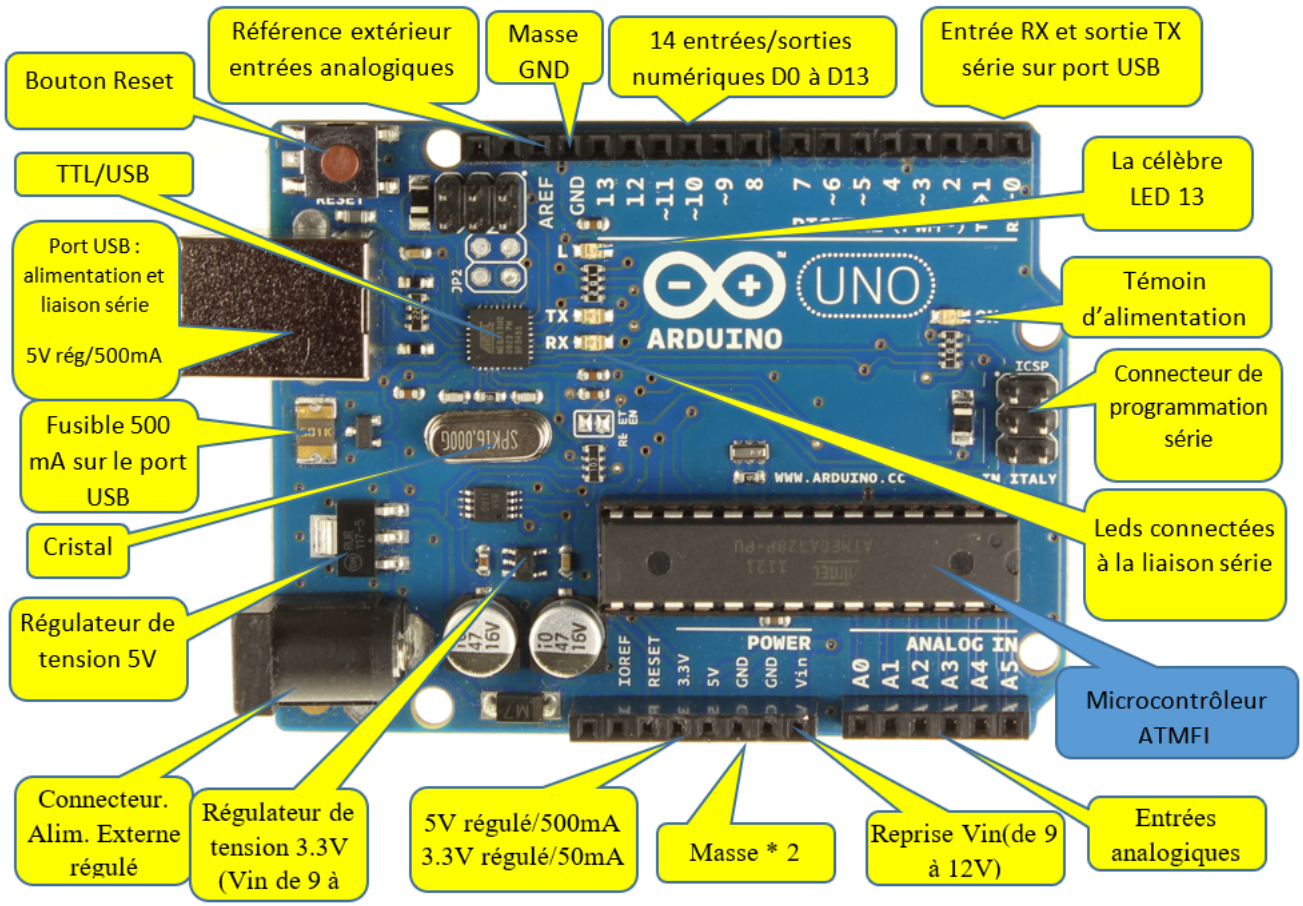

FIGURE 4.3 – Les éléments de la carte Arduino Uno

4.3.4 Conclusion

En conclusion de ce chapitre consacré aux systèmes embarqués, nous avons exploré cette discipline essentielle dans le domaine de la robotique et de l'automatisation.
Nous avons défini les systèmes embarqués comme des systèmes informatiques intégrés dans des appareils ou des machines, leur permettant de fonctionner de manière autonome et de répondre à des besoins spécifiques.
Nous avons également étudié les microcontrôleurs, qui jouent un rôle crucial dans le contrôle et la gestion des périphériques des appareils. Parmi eux, l'Arduino se démarque par sa popularité, sa facilité d'utilisation et sa polyvalence.
Les concepts abordés dans ce chapitre nous ont permis de mieux appréhender les systèmes embarqués, les microcontrôleurs et en particulier l'Arduino. Cette connaissance approfondie sera précieuse pour exploiter pleinement cette technologie lors de la conception et de la réalisation de notre robot à roues.
En résumé, ce chapitre nous a fourni les fondements nécessaires pour comprendre notre environnement de développement et les notions essentielles pour mener à bien notre projet.
Une fois toutes les données théoriques réalisées et présentées dans les différents chapitres, on va entamer notre réalisation pratique qui n'est autre que la conception et la réalisation d'un robot mobile multitâches .

Chapitre 5

Réalisation

5.1 Introduction

Notre véhicule est un robot mobile à quatre roues motrices fixes non holonome. Comparé à un robot équipé de roues omnidirectionnelles, le robot avec des roues classiques a des capacités de mouvement plus limitées. Cependant, il reste capable d'effectuer plusieurs types de mouvements pour se déplacer et accomplir des tâches :

- **Translation pure** : Le robot peut se déplacer en ligne droite sans rotation, en faisant tourner toutes les roues à la même vitesse.
- **Rotation pure** : Le robot peut tourner sur place autour de son axe de rotation en arrêtant la rotation des roues d'un côté tandis que les roues de l'autre côté continuent à tourner.
- **Mouvement en arc** : Le robot peut effectuer un mouvement en arc en faisant tourner les roues d'un côté plus rapidement que celles de l'autre côté, créant ainsi une trajectoire courbe.
- **Déplacement courbe** : Le robot peut réaliser un mouvement courbe en ajustant la vitesse de rotation des roues de chaque côté de manière différente.

En combinant ces différents mouvements, le robot peut se déplacer dans des environnements complexes et accomplir des tâches spécifiques en adaptant sa trajectoire en conséquence. Bien que les options de mouvement soient moins diverses que celles d'un robot à roues omnidirectionnelles, le robot peut toujours se déplacer de manière efficace dans de nombreux scénarios.

5.2 Cahier de charge

Projet de robot mobile commandé par une carte Arduino Uno.

5.2.1 Fonctionnement général

- Le robot doit être capable de se déplacer selon une translation vers l'avant et vers l'arrière, ainsi qu'un pivotement vers la gauche ou la droite.
- Le système de déplacement est constitué de 4 roues motrices actionnées par un module double pont en H.

5.2.2 Modes de fonctionnement

5.2.3 Mode Manuel

- Le robot peut être commandé à distance par deux méthodes :

- **Par infrarouge** : à l'aide d'une télécommande qui envoie un signal capté par un récepteur infrarouge.
- **Par bluetooth** : à l'aide d'une application sur téléphone qui se connecte au robot via un module Hc-06 et envoie les commandes à exécuter

5.2.4 Mode suiveur de ligne

- Le robot est équipé de capteurs suiveurs de ligne qui lui permettent de détecter une ligne et de la suivre

5.2.5 Mode détecteur d'obstacle

- Le robot doit avancer en autonomie tout en étant capable de détecter des obstacles grâce à un capteur ultrason.
- Si un obstacle est détecté, le robot doit l'éviter en prenant un autre chemin.

5.2.6 Mode parking

- Le robot doit être capable de trouver une place de stationnement libre et de calculer la distance à laquelle il se trouve de cette place.
- Selon la distance, le robot doit être capable de se garer en créneau ou en bataille.
- Les capteurs ultrasons et de vitesse sont utilisés pour effectuer cette manœuvre.

5.3 Présentation du matériel et logiciel

5.3.1 Carte arduino

Le système Arduino se compose essentiellement de deux éléments le matériel et le logiciel :
le matériel :
Le matériel consiste en une carte électronique qui repose sur un microcontrôleur Atmega fabriqué par Atmel. Malgré son prix abordable, cette carte offre une grande variété d'applications possibles. [18] (voir figure 5.1)

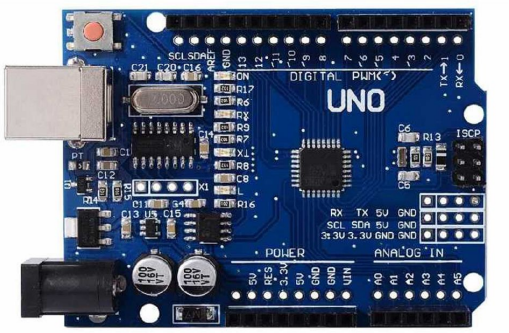

FIGURE 5.1 – Carte arduino UNO

Le logiciel :
Le logiciel est utilisé pour programmer la carte Arduino. Il propose de nombreuses fonctionnalités.(voir figure 5.2)

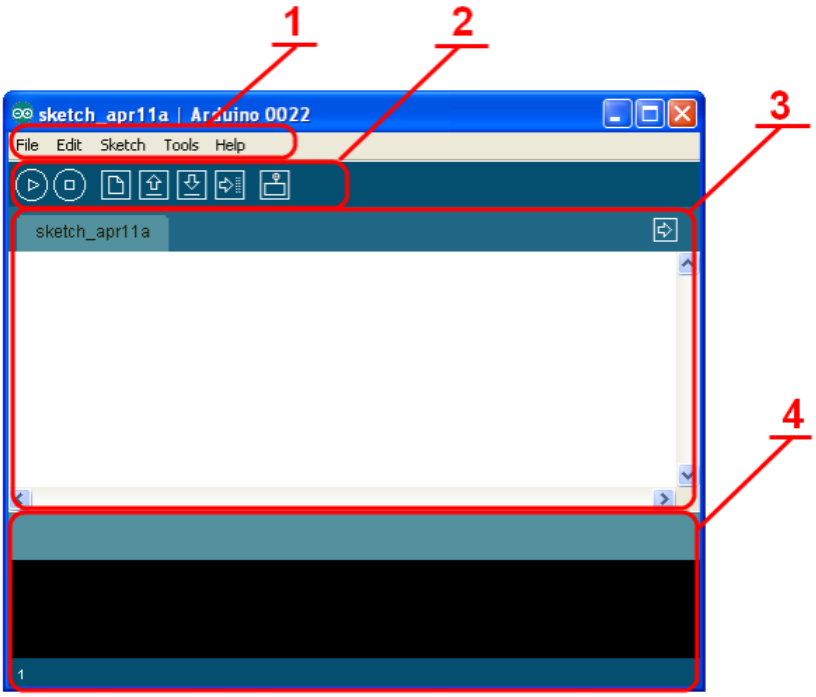

FIGURE 5.2 – L'interface du logiciel Arduino IDE

1 : Barre des options de configuration du logiciel
2 : Barre des boutons pour la programmation des cartes
3 : Fenêtre d'édition des programmes
4 : Fenêtre de la console (affiche les messages de la compilation) [18]

5.3.2 Proteus

Isis Proteus est un logiciel qui facilite le développement et la simulation d'applications grâce à une interface graphique simple et interactive
Lorsque vous lancez PROTEUS, vous accédez à un environnement familier de type Windows,composé d'une fenêtre principale et d'un ensemble de barres d'outils. En plus du menu traditionnel qui permet de gérer les fichiers, l'affichage et les options des projets,la fenêtre principale comprend une zone de travail dédiée au développement de circuits à simuler et tester. Une bibliothèque d'objets affiche une liste d'objets tels que des circuits électriques et électroniques utilisés dans l'application en cours. Les différentes touches magnétoscope sont des raccourcis qui permettent de lancer certaines actions. [26](voir figure 5.3)

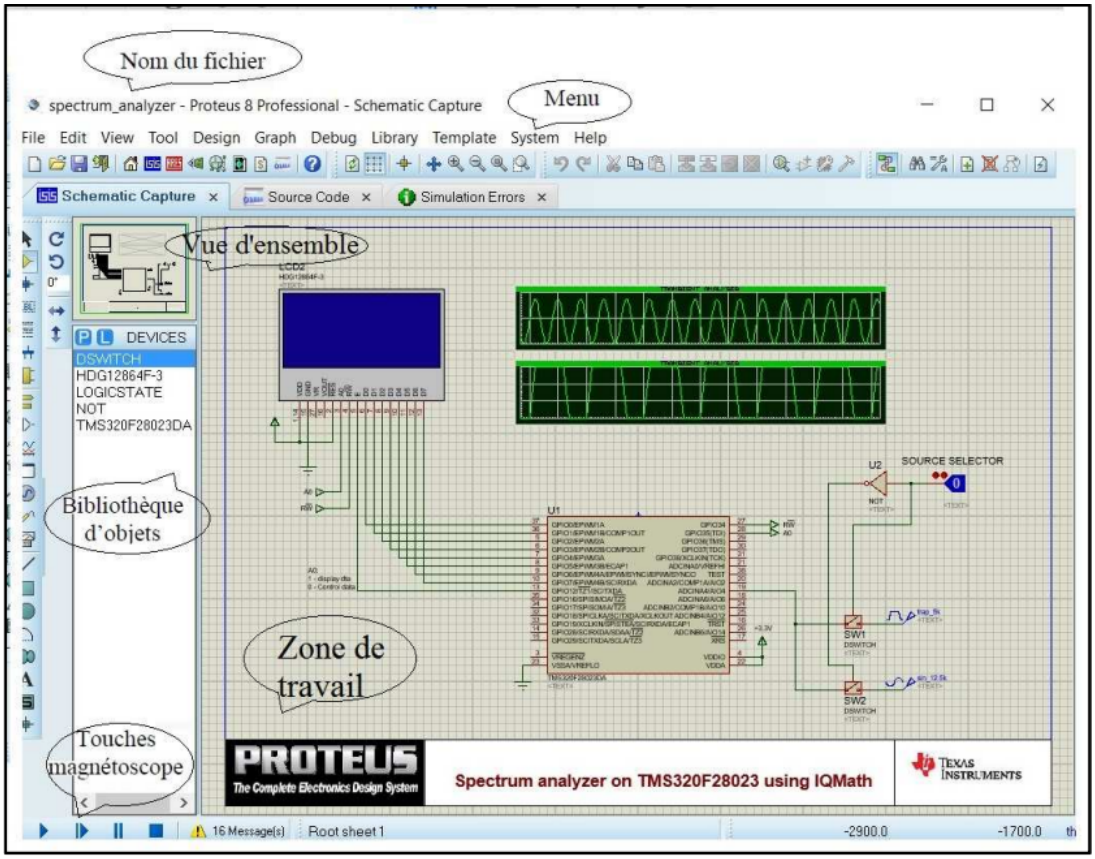

FIGURE 5.3 – Interface du logiciel Proteus

5.3.3 Fritzing

Fritzing est un logiciel open source spécialement conçu pour les personnes qui souhaitent créer des projets électroniques, en particulier des matériels libres, mais qui ne disposent pas du matériel requis. Il offre également la possibilité de concrétiser vos conceptions, de capturer des exemples à des fins de tutoriels, et bien plus encore. De plus, cet outil bénéficie d'une large communauté engagée dans sa mise à jour et prête à vous aider en cas de problème. Il représente également un excellent outil pour les cours d'électronique, tant pour les étudiants que pour les professeurs, permettant aux utilisateurs de partager et de documenter leurs prototypes, et même pour les professionnels. [5](voir figure 5.4)

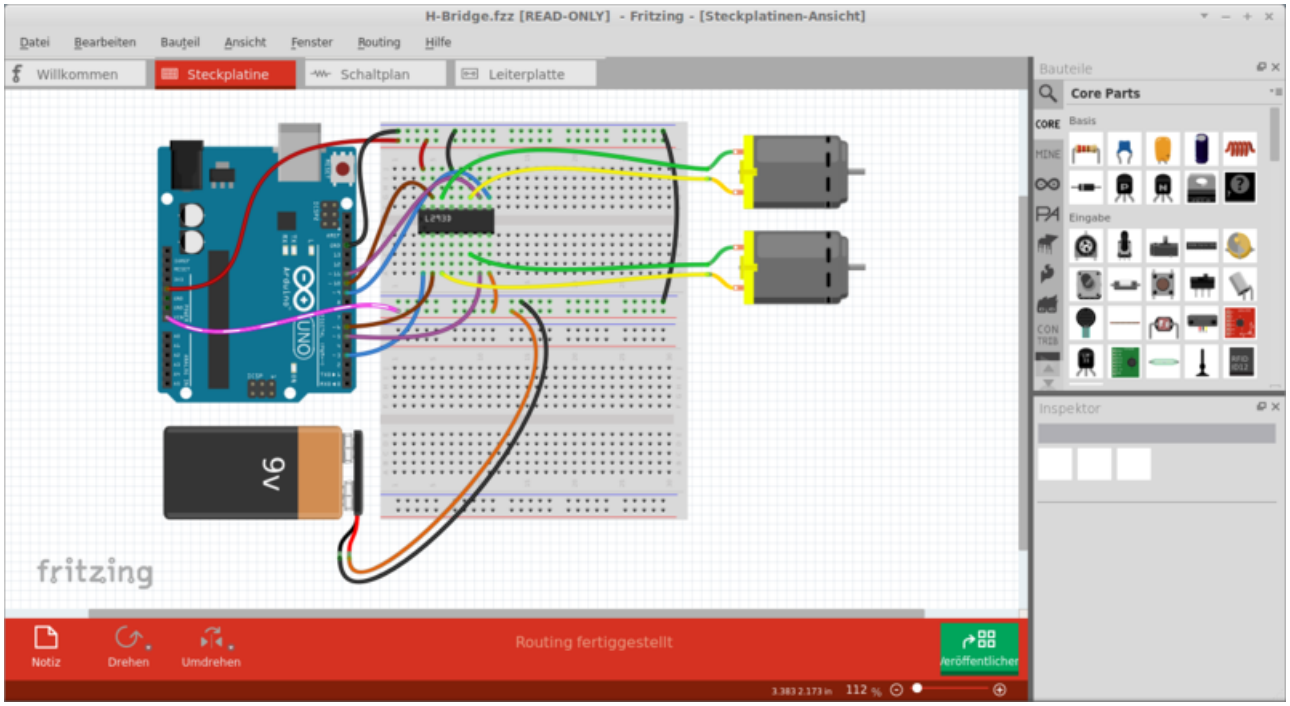

FIGURE 5.4 – Interface du logiciel Frizting

5.3.4 Module double pont en H L298N

Le circuit intégré L298N est composé de deux ponts en H réalisés à l'aide de transistors à jonction. Il est couramment utilisé pour contrôler des moteurs à courant continu de faible puissance en utilisant une modulation de largeur d'impulsion (PWM).(voir figure 5.5)

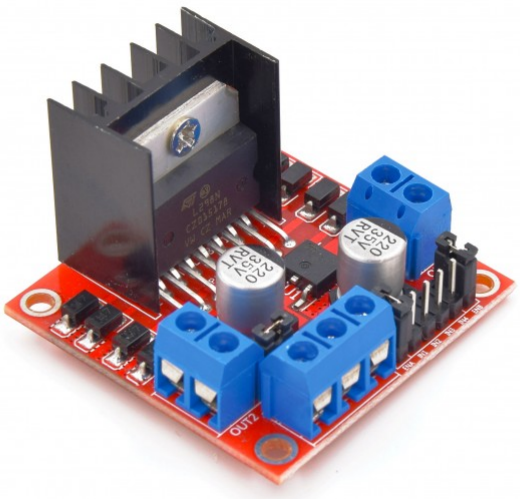

FIGURE 5.5 – Module driver l298N

Le L298N est disponible en deux versions : une version à montage traversant dans un boîtier Multiwatt15 et une version à montage en surface dans un boîtier PowerSO20. Il existe de nombreuses cartes préassemblées qui intègrent un L298N. Par exemple, l'Arduino Motor Shield est une carte d'extension qui se connecte directement à une carte Arduino, mais elle est limitée à la commande d'un seul moteur. [15] Vss représente la tension d'alimentation de la partie logique du circuit (maximum

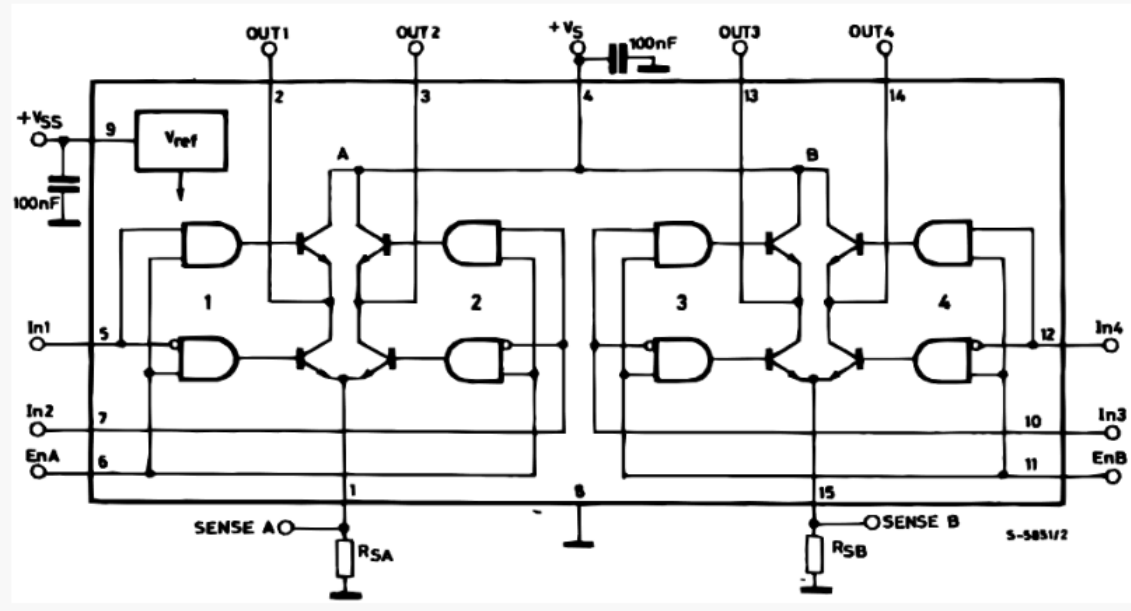

FIGURE 5.6 – Le schéma interne (simplifié) du L298N

+7 V), c'est-à-dire la tension de commande du pont. Vs correspond à l'alimentation de la charge (maximum 50 V), dont la puissance doit être adaptée en fonction de la charge utilisée. La capacité de courant maximale supportée par le pont en continu est de 2 A, ce qui limite son utilisation aux charges de faible puissance. Il est important de noter la présence des résistances RSA et RSB dans ce schéma (qui ne font pas partie du L298N) et qui sont utilisées pour mesurer le courant. Cependant, leur valeur est si faible que les points 1 et 15 peuvent être considérés comme étant à la masse. Dans la plupart des cas, les bornes 1 et 15 sont simplement reliées à la masse.
Nous nous concentrons sur le pont situé à gauche, avec les bornes de sortie OUT1 et OUT2. Les trois ports de commande de ce pont sont IN1, IN2 et ENA (Enable). Chacun des quatre transistors du pont est contrôlé par la sortie d'une porte ET. Le cercle à l'entrée de deux de ces portes représente une inversion.
Lorsque ENA est égal à 0, les sorties des quatre portes ET sont à l'état bas, indépendamment des états de IN1 et IN2. Les transistors sont donc bloqués (équivalents à des interrupteurs ouverts). Lorsque ENA est égal à 1, l'état des transistors dépend des valeurs de IN1 et IN2. IN1 contrôle le demi-pont 1, composé des deux transistors situés à gauche. IN2 contrôle le demi-pont 2, composé des deux transistors situés à droite. Ainsi, lorsque IN1 est égal à 1, le transistor supérieur du demi-pont 1 est en état de saturation (équivalent à un interrupteur fermé), tandis que le transistor inférieur est bloqué (interrupteur ouvert). La fonction de IN2 est similaire pour le demi-pont 2. Pour appliquer la tension Vs à la charge, il est nécessaire de mettre ENA à 1 et de choisir entre IN1=1, IN2=0 ou IN1=0, IN2=1 en fonction de la polarité souhaitée. [15] (voir figure 5.6)

5.3.5 Moteur à courant continu

Le moteur à courant continu est un dispositif électromécanique qui permet la conversion d'énergie. Son rôle est de convertir l'énergie électrique continue en énergie mécanique, ce qui permet de faire fonctionner une charge en mouvement. (voir figure 5.7)

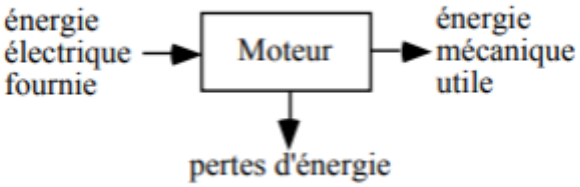

FIGURE 5.7 – La conversion d'énergie dans un moteur à courant continu

Moteur de type TT

Les moteurs à courant continu TT sont une solution pratique et économique pour faire progresser les projets. Ces moteurs sont équipés d'une boîte de vitesses avec un rapport de transmission de 1 :48 et sont livrés avec deux fils de 200 mm dotés de connecteurs mâles de 0,1 pouce, ce qui les rend parfaitement adaptés pour être branchés sur une breadboard ou des borniers.
On peut alimenter ces moteurs avec 3VDC jusqu'à 6VDC, ils iront bien sûr un peu plus vite aux tensions plus élevées(voir figure 5.8)

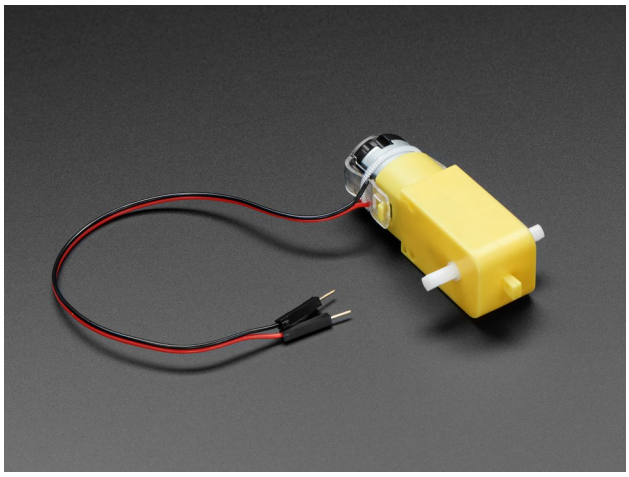

FIGURE 5.8 – Moteur de type TT

Caractéristiques

- Tension nominale : 3 6V
- Courant permanent à vide : 150mA +/- 10%.
- Min. Vitesse de fonctionnement (3V) : 90+/- 10% RPM
- Min. Vitesse de fonctionnement (6V) : 200+/- 10% RPM
- Couple : 0.15Nm 0.60Nm 0.60Nm
- Couple de décrochage (6V) : 0.8kg.cm.
- Rapport d'engrenage : 1 :48
- Dimensions du corps : 70 x 22 x 18mm
- Longueur des fils : 200mm et 28 AWG
- Poids du produit : 30,6g / 1,1 oz [1]

5.3.6 Servo moteur

Fourni avec trois cornes (bras) et du matériel de fixation. Ce servo est petit et léger, mais offre une puissance de sortie élevée. Il permet une rotation d'environ 180 degrés (90 degrés dans chaque direction) et fonctionne de manière similaire aux servos standards, mais dans un format plus compact. on peut utiliser n'importe quel code, matériel ou bibliothèque de servo pour le contrôler. Il convient parfaitement aux débutants qui souhaitent faire bouger des objets sans avoir à construire un contrôleur de moteur avec rétroaction et boîte de vitesses, d'autant plus qu'il s'adapte facilement dans des espaces restreints. (voir figure 5.9)

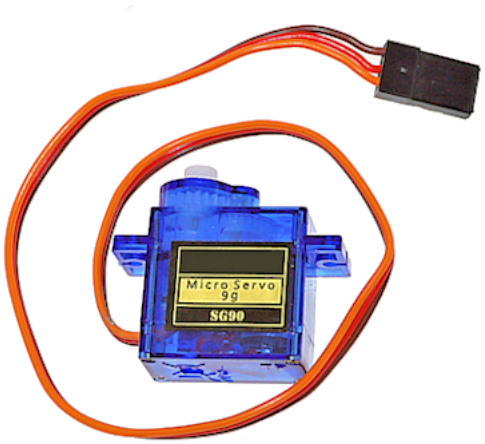

FIGURE 5.9 – Servo moteur

Les caractéristiques du SG90 sont les suivantes :

- Dimensions : 22 x 11.5 x 27 mm
- Poids : 9 g
- Tension d'alimentation : 4.8v à 6v
- Vitesse : 0.12 s / 60° sous 4.8v
- Couple : 1.2 Kg / cm sous 4.8v
- Amplitude : de 0 à 180 degré [2]

Le servo est équipé d'une prise de type Graupner à 3 fils.La correspondance des fils est la suivante :

Marron	Masse
Rouge	+5v
Orange	Commande

TABLE 5.1 – Les fils du SG90

5.3.7 Capteur ultrason

Le capteur à ultrasons HC-SR04 permet de mesurer avec précision la distance des objets situés entre 2 cm et 400 cm du capteur, avec une précision de 3 mm. Ce capteur est composé d'un émetteur d'ultrasons, d'un récepteur et d'un circuit de commande. Le principe de fonctionnement est le suivant :

1- Un signal numérique est envoyé à l'émetteur pendant 10 µs, mettant le signal à l'état haut.
2- Le capteur envoie automatiquement 8 impulsions d'ultrasons à une fréquence de 40 kHz et détecte les signaux de retour.
3- Si un signal de retour est détecté, la durée de l'état haut du signal reçu correspond au temps écoulé entre l'émission et la réception des ultrasons.

Le calcul de la distance s'effectue selon la formule suivante : Distance = (temps à l'état haut du signal reçu * vitesse du son) / 2, en tenant compte de la vitesse du son dans l'air (340 m/s).[7] (voir figure 5.10)

FIGURE 5.10 – Capteur ultrason HC-RS04

Paramètres électriques :

Tension d'alimentation	5V DC
Courant d'alimentation	15 mA
Fréquence de travail	40 Hz
Distance maximale de détection	4m
Distance minimale de détection	2 cm
Angle de détection	15 degrés
Signal d'entrée de l'émetteur	Impulsion a l'état haut de 10us
Signal de sortie de l'émetteur	Signal numérique a l'état haut et la distance proportionnellement
Dimension	45*20*15mm

TABLE 5.2 – Paramètres électriques

5.3.8 Module du capteur suiveur de ligne KY-033

Le module capteur est capable de détecter la présence d'une surface qui réfléchit ou absorbe la lumière devant lui. Lorsqu'une telle surface est détectée, le module émet une sortie numérique, comme illustré dans les images ci-dessous. La sensibilité du capteur, c'est-à-dire la distance minimale à laquelle il peut détecter les objets, peut être ajustée à l'aide du contrôleur.
Le fonctionnement d'un capteur suiveur de ligne sur l'utilisation d'une LED infrarouge pour illuminer l'obstacle, et sur la mesure de la lumière réfléchie à l'aide d'une photodiode ou d'un phototransistor. [8] (voir figure 5.11)

FIGURE 5.11 – Module suiveur de ligne KY-033

Spécifications Module KY-033 :

- Tension de fonctionnement : DC 3.3 V-5 V

- Courant de travail :$\geq 20mA$
- Distance de détection : 2-40 cm
- Interface IO : interfaces 3 fils
- Angle efficace : 35 degré

Broches de connexion

- Signal : pour transmettre le signal mesuré
- Vcc : pour l'alimentation
- Gnd : la masse [8]

5.3.9 Module Bluetooth HC-06

Le module HC-06, qui est un module Bluetooth, permet d'établir une connexion sans fil (liaison série) entre une carte Arduino et d'autres appareils disposant d'une connexion Bluetooth, tels qu'un smartphone, une tablette ou une autre carte Arduino. Contrairement au module HC-05 qui agit en tant que "maître", le module HC-06 est un module "esclave". Cela signifie qu'un module "maître" peut initier une demande de jumelage avec un autre périphérique Bluetooth, tandis qu'un module "esclave" ne peut recevoir que des demandes de jumelage.(voir figure 5.12)

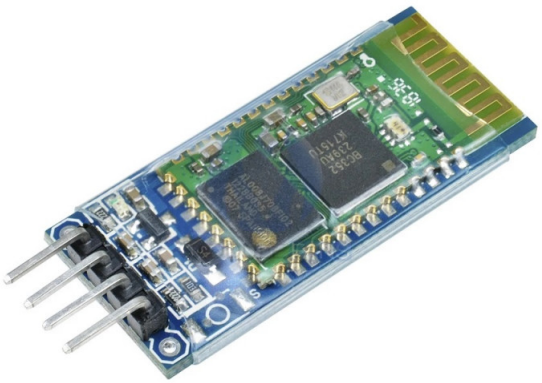

FIGURE 5.12 – Module bluetooth HC-06

Spécifications :

- Tension d'alimentation : 3V3 à 5V
- Tension niveau I/O : 3V3 Maximum (fortement déconseillé de mettre du 5V sur les I/Os du module)
- Consommation : $\sim$ 50mA @ 3V3
- Fréquence : Bande ISM 2.400 - 2.4835GHz (antenne incluse dans PCB)
- Data rate : Jusqu'à 2Mbps
- Modulation Radio : GFSK
- Interface UART : TTL 3Mbps Max - 3V3 Maximum
- Portée : $\sim$ 10m (non garantie) [6]

5.3.10 Bloc d'alimentation

Il s'agit d'un support de piles(9v).Il va nous permettre d'alimenter la Carte Arduino Uno et le Driver L298N. [9](voir figure 5.13)

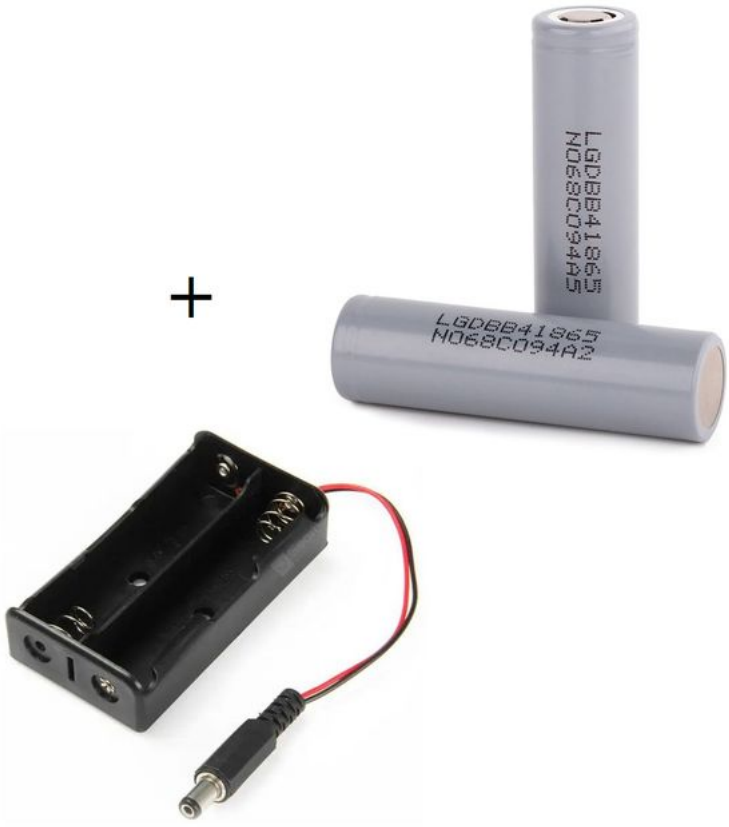

FIGURE 5.13 – Bloc d'alimentation + piles

5.3.11 Capteur de vitesse

Ce composant est conçu pour mesurer le taux d'un moteur électrique. Ce module peut être utilisé en association avec un microcontrôleur pour détecter la vitesse du moteur, compter les impulsions, limiter la position, etc. Fondamentalement, un compteur de vitesse mesure simplement la fréquence à laquelle un événement se produit. Généralement, cela est réalisé en comptant le nombre d'événements pendant une période de temps donnée (intervalle d'intégration) et en divisant ce nombre par le temps pour obtenir le taux.
Le module de capteur de vitesse de moteur décrit ici est un dispositif simple compatible avec les microcontrôleurs. Il génère des trains d'impulsions traités lorsque le trajet optique de son capteur est physiquement interrompu par une roue fendue ou un mécanisme similaire (un capteur optique typique comprend une diode électroluminescente qui fournit l'éclairage et un phototransistor qui détecte la présence ou l'absence de cet éclairage). Le capteur optique utilisé ici est transmissif et se compose d'une diode électroluminescente infrarouge et d'un phototransistor. Cette configuration permet d'éviter les interférences provenant de sources de lumière extérieures indésirables, et en ajustant les deux composants pour une fréquence spécifique de rayonnement, ils deviennent encore plus résistants aux interférences indésirables[3].(voir figure 5.14)

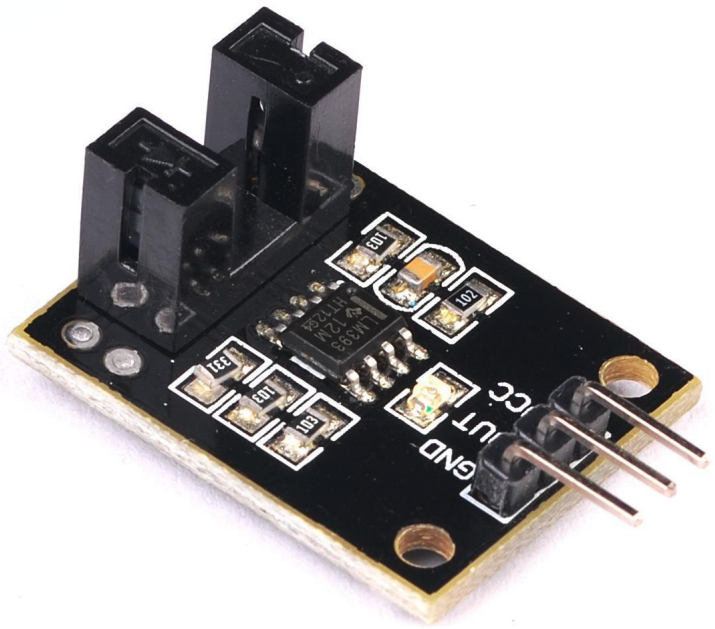

FIGURE 5.14 – Capteur de vitesse

5.4 Simulation de quelques composants

5.4.1 Servomoteur

Pour cette simulation, on a besoin de trois composants :

- Une carte arduino uno
- Un potentiomètre
- Un servo moteur

La connexion est faite ainsi :
Pour l'alimention, le servomoteur et le potentiomètre sont reliés au GND et au 5v.
Pour la communication avec l'Arduino, le servo est connecté au port digital 7, tandis que le potentiomètre est connecté au port analogique 0.

Le code :

```
#include <Servo.h>

Servo monservo;
void setup ()
{
  monservo.attach(7);
  }
void loop(){
 int v=analogRead (A0);
 float angle = v*180.0/1023 ;
 monservo.write(angle);
 delay(100);
  }
```

Résultat

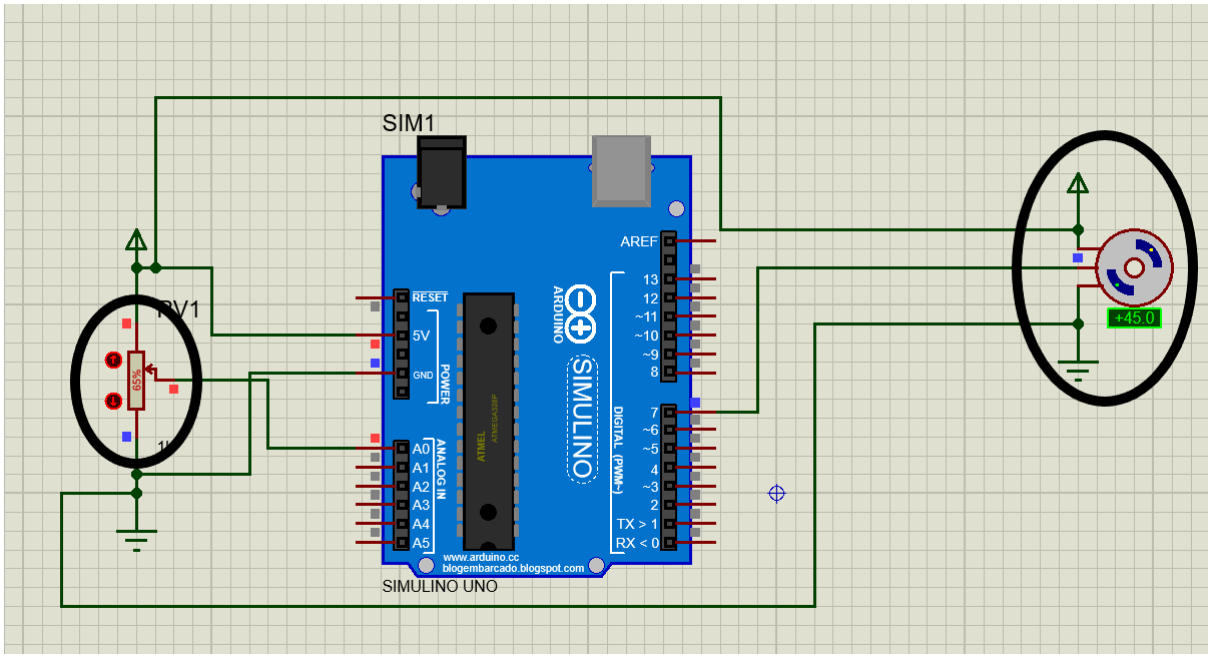

FIGURE 5.15 – Simualtion servomoteur

Commentaires

Le potentiomètre est une résistance variable qui permet de mesurer un changement de résistance. En envoyant un courant entre les bornes extrêmes du potentiomètre, nous pouvons lire la valeur de la tension qui se crée sur la borne du milieu grâce au pont diviseur formé.

- Dans notre programme, nous devons inclure la bibliothèque « servo.h » pour utiliser les fonctions prédéfinies afin de piloter le servomoteur.
- La fonction « Monservoattach() » permet de connecter le servo au pin désiré.
- La fonction « Analogread() »permet de lire la valeur analogique (variation de tension) du potentiomètre. Cette valeur, en volts, est automatiquement transformée en une nouvelle valeur équivalente, sur une échelle d'entre 0 et 1023 selon la valeur reçue (0 V= 0 / 5V= 1023) :

$$5v \rightarrow 1023 \tag{5.1}$$

$$x \rightarrow valeur \tag{5.2}$$

$$X = (valeur * 1023)/5 \tag{5.3}$$

Ici on abesoin la valeur de l'angle en degré donc on procède à ce calcul :

$$Angle = (valeur * 180)/1023 \tag{5.4}$$

- « monservo.write() » sert à faire tourner le servo à l'angle désiré.

5.4.2 Capteur ultrason

Pour cette simulation on a besoin de ces composants :

- Une carte Arduino Uno
- Un capteur ultrason SR-04
- Un Visual terminal
- Un potentiomètre

Pour la connexion :

- Capteur ultrason
- Le port trig connecté au pin 7
- Le port echo connecté au pin 6
- Potentiomètre
- La résistance variable est connecté au pin test du capteurs ultrason .
- Tous les composants sont alimentés au 5v.

Le code :

```
const int trigPin = 7;
const int echoPin = 6;

long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);
}
```

```
void loop() {

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  delay(1000);
}
```

Résultat

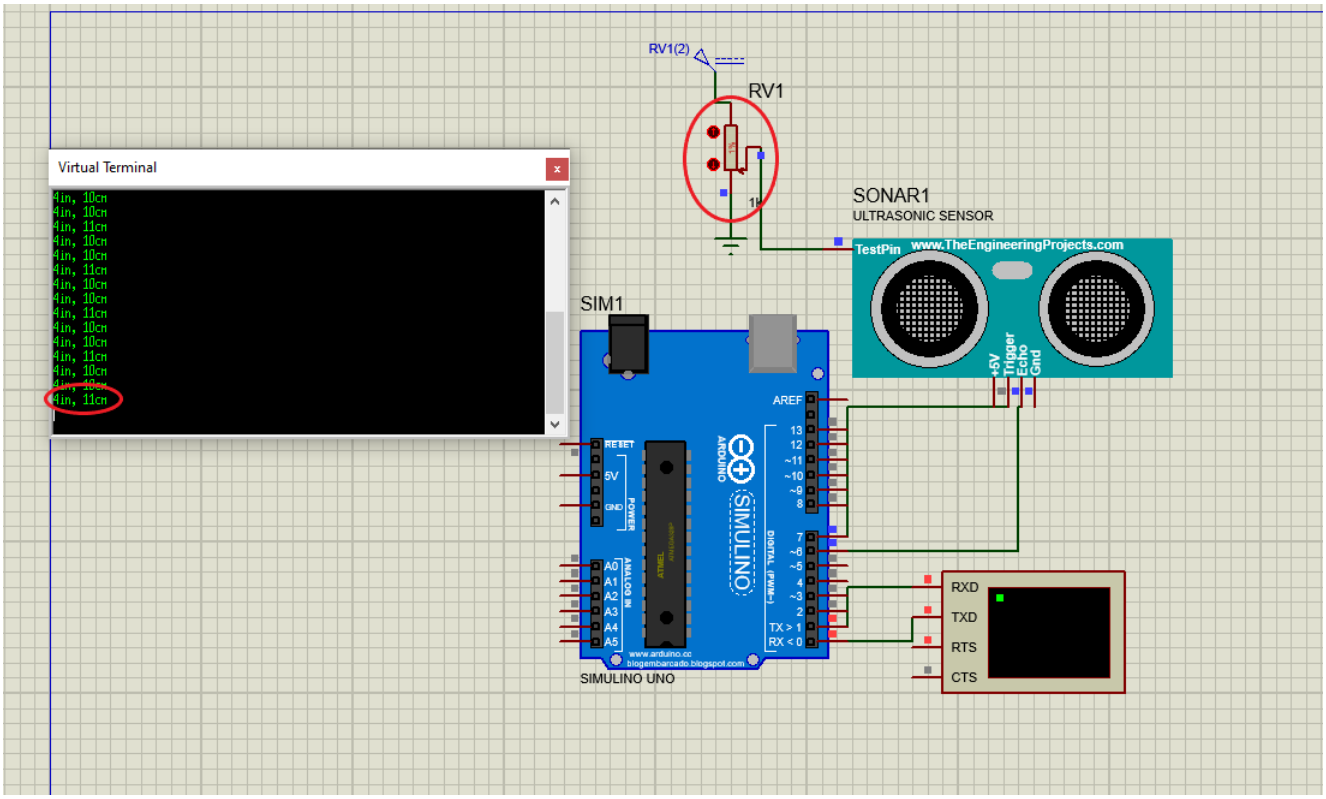

FIGURE 5.16 – Simulation du capteur ultrason

Commentaires

- Nous définissons les broches utilisées par le capteur ultrason SR-04. La broche trigPin est utilisée pour émettre les impulsions de déclenchement, tandis que la broche echoPin est utilisée pour recevoir les signaux d'écho.
- Nous déclarons les variables duration (durée) et distance (distance) qui serviront à stocker les valeurs mesurées
- la fonction « setup() » est une fonction d'initialisation qui s'exécute une seule fois au démarrage du programme. À l'intérieur de cette fonction, nous configurons les modes des broches utilisées par le capteur : « trigPin » en sortie pour l'envoi des impulsions de déclenchement, et « echoPin » en entrée pour la réception des signaux d'écho. De plus, nous initialisons la communication série à une vitesse de 9600 bauds à l'aide de la fonction « Serial.begin(9600) », ce qui nous permettra d'afficher les résultats sur le moniteur série.
- la fonction « loop() » est la boucle principale du programme qui s'exécute en continu après l'exécution de la fonction « setup() ». À l'intérieur de cette boucle, nous effectuons les opérations de mesure et d'affichage de la distance mesurée.
- nous envoyons une impulsion de declenchement en mettant la broche « trigPin » » à LOW pendant 2 microsecondes, puis à HIGH pendant 10 microsecondes, et enfin à LOW à nouveau.
- La fonction « pulseIn() » mesure la durée de l'écho en attendant le signal HIGH sur la broche « echoPin ». La valeur de cette durée est ensuite stockée dans la variable duration
- nous calculons la distance en multipliant la durée de l'écho par un facteur de conversion appropriée.

Pour le calcul de la distance : la vitesse de son dans l'air est généralement considérer comme étant d'environ 340 mètres par seconde (ou 34 000 centimètres par seconde) .Dans notre calcule ,nous utilisons une approximation de la vitesse du son de 34cm/ms ou(0.034cm/us)pour simplifier les

calculs .
La durée de l'écho mesurée est stockée dans la variable "duration" en microsecondes.
Pour obtenir la distance réelle en centimètres, nous multiplions la durée de l'écho (duration) par la vitesse du son approximative (0.034cm/us) et divisons le résultat par 2. Ceci est dû au fait que le signal ultrasonique parcourt deux fois la distance entre le capteur et l'objet réfléchissant.
Ainsi, la formule distance = duration * 0,034 / 2 permet de calculer la distance en centimètres.

5.4.3 Driver moteur L298n

Pour cette simulation on a besoin de trois composants :

- Une carte arduino uno
- Un driver moteur L298n
- Deux moteurs DC

Pour le connexion :

- L298N
- Motor A :
  - enA → pin 9
  - in1 → pin 8
  - in2 →pin 7
- Motor B :
  - int enB →pin 3 ;
  - int in3 → pin 5 ;
  - int in4 → pin 4 ;

  Le driver moteur est alimenté en 12v

Le code :

```
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;
// Motor B connections
int enB = 3;
int in3 = 5;
int in4 = 4;

void setup() {
  // Set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}

void loop() {
  directionControl();
  delay(1000);
  speedControl();
  delay(1000);
}

// This function lets you control spinning direction of motors
void directionControl() {
// Set motors to maximum speed
// For PWM maximum possible values are 0 to 255
  analogWrite(enA, 255);
  analogWrite(enB, 255);

// Turn on motor A & B
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  delay(2000);

// Now change motor directions
  digitalWrite(in1, LOW);
```

```
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(2000);

  // Turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}

// This function lets you control speed of the motors
void speedControl() {
  // Turn on motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);

// Accelerate from zero to maximum speed
  for (int i = 0; i < 256; i++) {
    analogWrite(enA, i);
    analogWrite(enB, i);
    delay(20);
  }

// Decelerate from maximum speed to zero
  for (int i = 255; i >= 0; —i) {
    analogWrite(enA, i);
    analogWrite(enB, i);
    delay(20);
  }

  // Now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

Résultat :

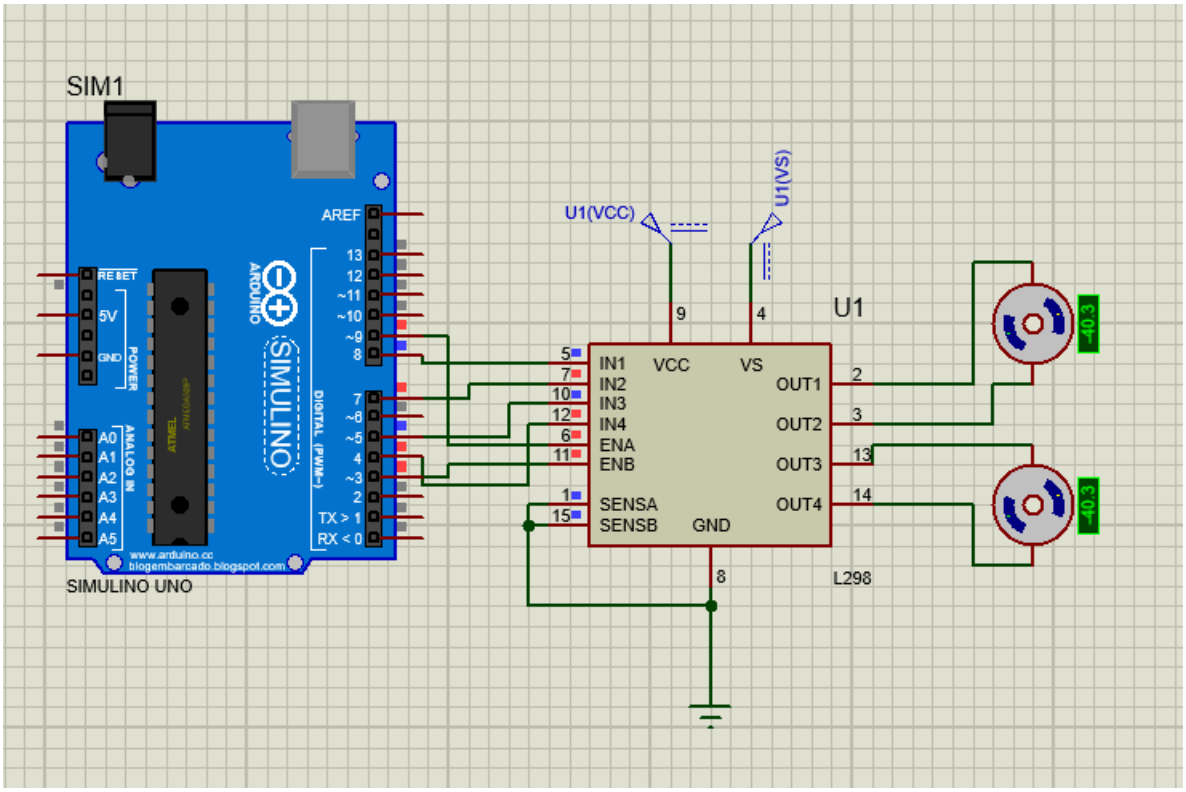

FIGURE 5.17 – Simulation driver moteur l298n

Commentaires :

Nous définissons les broches utilisées pour le driver moteur L298N. Les broches enA et enB sont les broches de commande de la vitesse des moteurs, tandis que les broches in1, in2, in3 et in4 sont les broches de commande de la direction des moteurs

La fonction loop() est la boucle principale du programme. Elle appelle deux fonctions, directionControl() et speedControl(), qui contrôlent respectivement la direction et la vitesse des moteurs.

La fonction directionControl() est utilisée pour contrôler la direction de rotation des moteurs. Dans cet exemple, les moteurs tournent d'abord dans un sens pendant 2 secondes, puis dans l'autre sens pendant 2 secondes. Les broches de contrôle de la vitesse (enA et enB) sont réglées sur la valeur maximale de 255. Les broches de direction (in1, in2, in3, in4) sont utilisées pour définir le sens de rotation des moteurs en les mettant à HIGH ou LOW ; le tableau suivant montre le sens de rotation du moteur en fonction des états des sorties digitales in1, in2, in3 et in4

La vitesse est contrôlée en utilisant la fonction analogWrite() pour générer une modulation de largeur d'impulsion (PWM) sur les broches enA et enB.

Le signal PWM (modulation de largeur d'impulsion) est utilisé pour contrôler la puissance fournie à un composant électronique en modulant la largeur des impulsions d'un signal.Sur Arduino, les broches marquées avec le symbole " " peuvent générer un signal PWM. Bien qu'elles soient des sorties numériques, elles peuvent être utilisées pour générer un signal PWM en utilisant la fonction analogWrite(). Cette fonction nécessite deux arguments : le numéro de la broche PWM et la valeur de la largeur d'impulsion. La valeur de la largeur d'impulsion varie de 0 (0% de devoir) à 255 (100% de devoir). Une valeur de 0 correspond à une impulsion constamment basse (0V), tandis que 255 correspond à une impulsion constamment haute (5V pour Arduino Uno). En utilisant des valeurs intermédiaires, il est possible de générer une impulsion avec une largeur variable, permettant ainsi de contrôler la puissance fournie à un composant tel qu'un moteur ou une LED.

5.4.4 Scénario du mode suiveur de ligne

On va utiliser des boutons poussoir au lieu des modules suiveur de lignes connecté à l'arduino Le scénario du véhicule sera établi selon l'état (haut ou bas) des boutons, quatre situations seront affichées sur le moniteur .le tableau représente les cas différents :(voir figure 5.18)

Position de ligne	Etat du capteur			Action du robot
	gauche	milieu	droite	
	bas	haut	bas	aller tout droit
	haut	bas	bas	tourner à gauche
	bas	bas	haut	tourner à droite
	bas	bas	bas	s'arrêter

FIGURE 5.18 – Différents scénarios du mode suiveur de ligne

Le code :

```
void setup() {
  // put your setup code here, to run once:
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
int R=digitalRead(3);
int M=digitalRead(4);
int L=digitalRead(5);
if(R==0 & M==1 & L==0)
{Serial.print("Avancer tout droit ");
Serial.println("");
  }
else if(R==1 & M==0 & L==0)
{Serial.print("Tourner vers la droite ");
Serial.println("");

  }
  else if(R==0 & M==0 & L==1)
{
  Serial.print("Tourner vers la gauche");
  Serial.println("");
  }
else {
    Serial.print("ARRET");
     Serial.println("");
    }
  delay(1000);
}
```

Résultat :

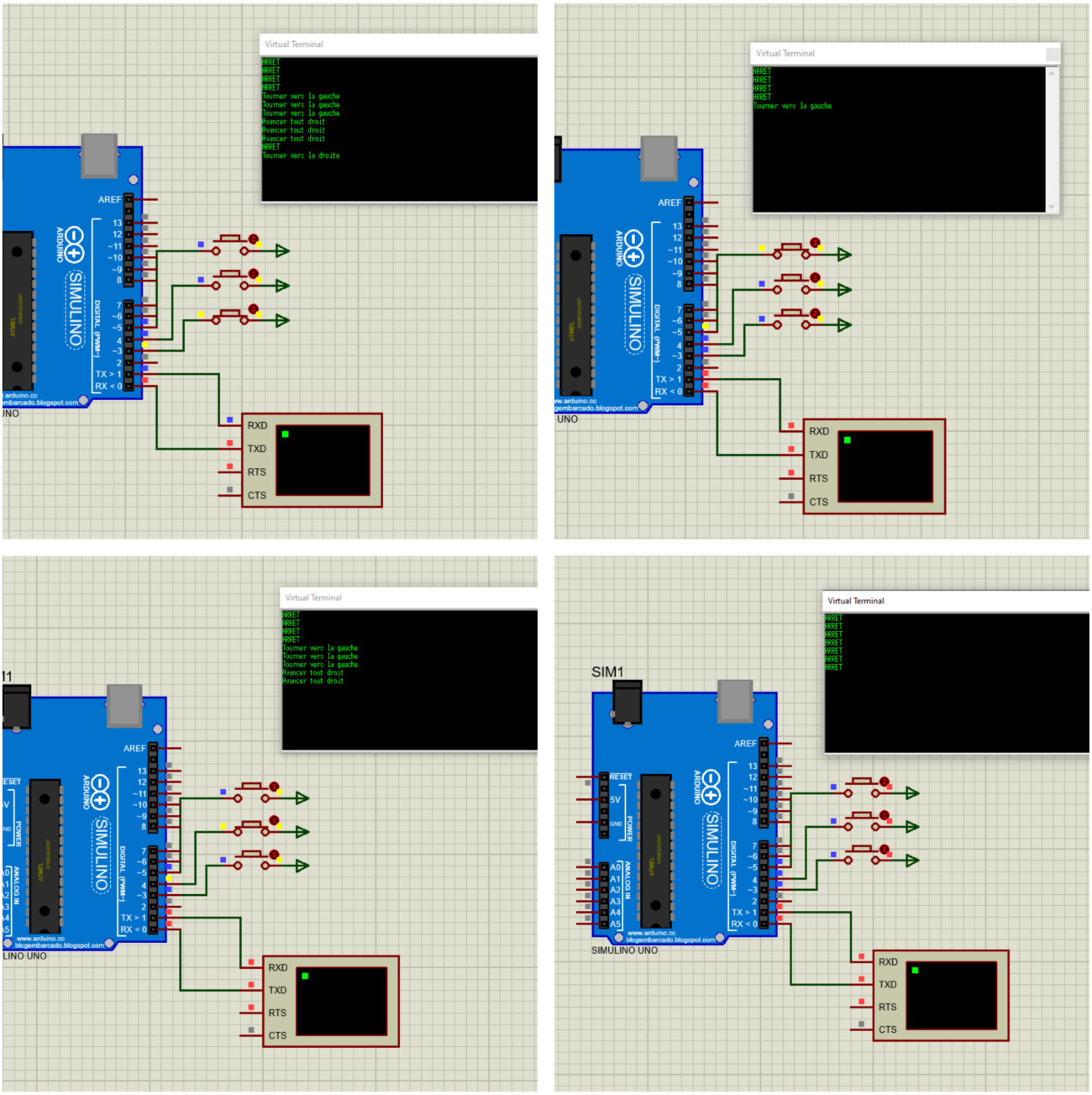

FIGURE 5.19 – Simulation des différents scénarios de suiveur de ligne

5.5 Partie pratique

5.5.1 Mode manuel IR

Le système de contrôle à distance infrarouge universel se compose de deux parties distinctes : l'émetteur et le récepteur. L'émetteur est constitué d'une télécommande infrarouge, tandis que le récepteur est composé d'un tube récepteur infrarouge. Les signaux émis par la télécommande infrarouge sont sous la forme d'une série de codes binaires d'impulsions. Pour éviter les interférences provenant d'autres signaux infrarouges pendant la transmission sans fil, ces signaux sont généralement modulés à une fréquence porteuse spécifique, puis émis à travers un phototransistor infrarouge.
Le tube récepteur infrarouge filtre les autres ondes de bruit, ne captant que les signaux à la fréquence désirée, et les démodule pour reconstituer les codes binaires d'impulsions d'origine. Les signaux lumineux transformés par le tube récepteur sont convertis en signaux électriques faibles et envoyés à une diode électroluminescente infrarouge. À l'intérieur du circuit intégré, ces signaux sont amplifiés par un amplificateur et passent par un contrôle automatique du gain, un filtrage passe-bande, une démodulation et une mise en forme d'onde, ce qui permet de restaurer le signal original émis par la télécommande. Ce signal est ensuite reconnu et transmis à la carte ARDUINO via une broche de sortie du module de réception infrarouge.
Avant de d'établir le code principal, nous devons connaitre le code hexadécimal des signaux de notre télécommande, pour cela il nous faut ce code :

```
#include <IRremote.h>
 int RECV_PIN = 12;
 IRrecv irrecv(RECV_PIN);
 decode_results results;
 void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}
 void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

Une fois le code téléversé sur la carte Arduino et le montage réalisé selon l'illustration, il convient d'ouvrir la fenêtre du moniteur série de l'Arduino IDE. En appuyant sur les boutons de la télécommande, le moniteur affiche les codes des signaux émis par l'émetteur.

Le résultat :

STOP	**80BF5BA4**
AVANT	**80BF7B84**
ARRIERE	**80BF6B94**
A DROITE	**80BF1BE4**
A GAUCHE	**80BFDB24**
VITESSE +	**80BF718E**
VITESSE -	**80BF619E**
MODE SUIVEUR DE LIGNE	**80BFF10E**
MODE DETECTEUR D'OBSTACLES	**80BF49B6**
MODE IR	**80BFE11E**
MODE BLUETOOTH	**80BFC936**
SWITCH DE MODE	**80BF6996**
SERVO TO THE RIGHT	**80BF23DC**
SERVO TO THE LEFT	**80BFA35C**

TABLE 5.3 – Les codes hexadécimaux des boutons

Organigramme de mode manuelle IR :

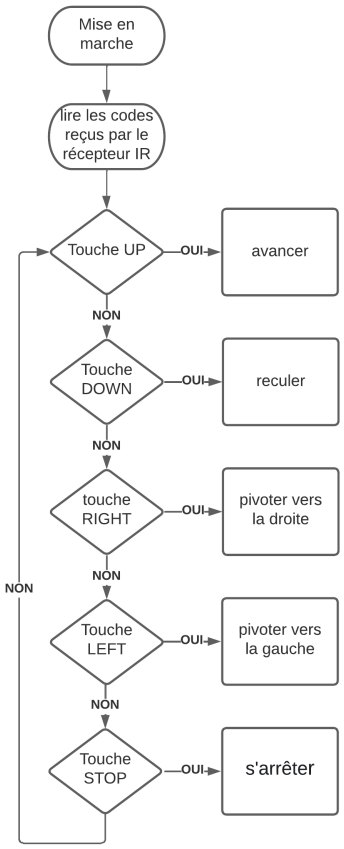

FIGURE 5.20 – Organigramme du mode manuelle IR

Le schéma de câblage :

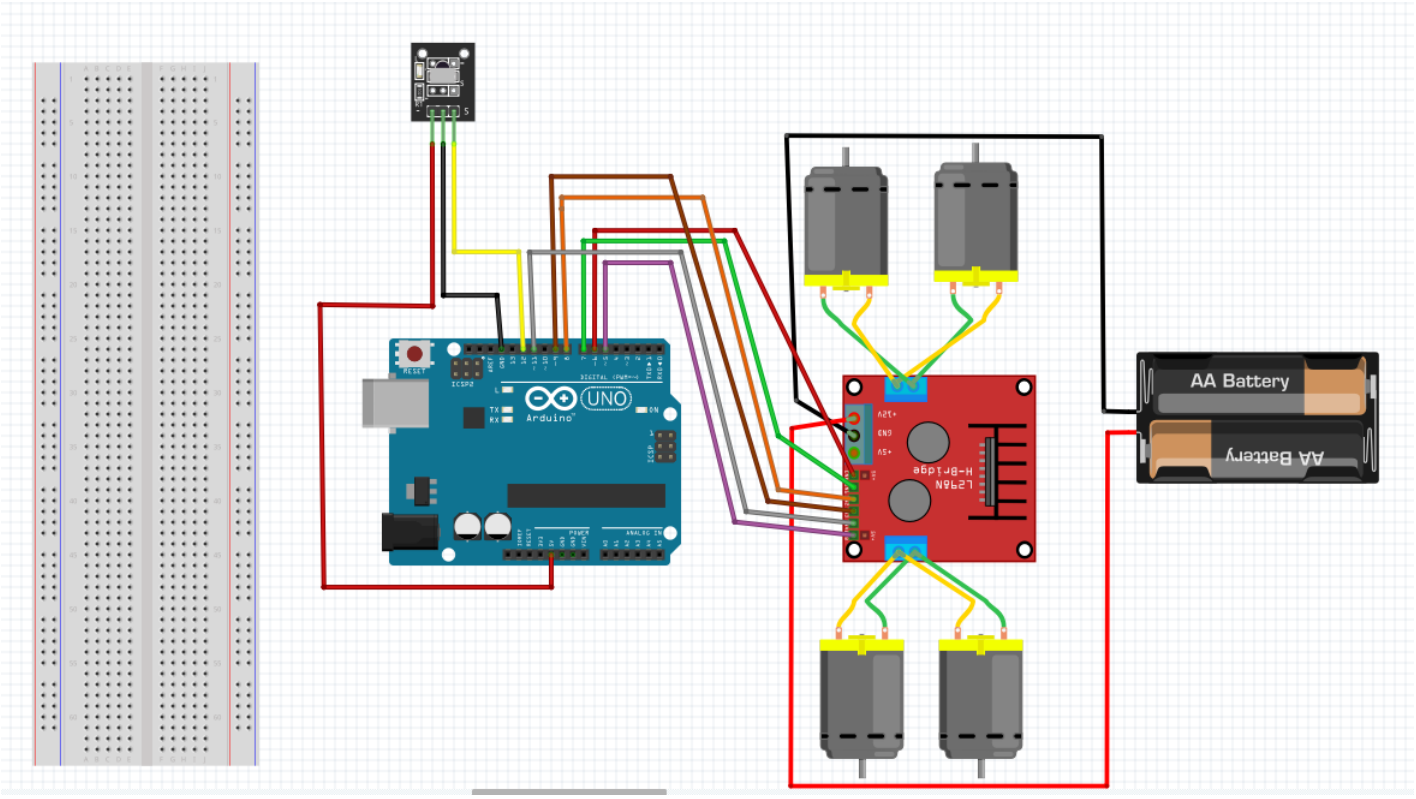

FIGURE 5.21 – Schéma de câblage du mode IR par frizting

Le code :

```
#include <IRremote.h>
int RECV_PIN = 12;
IRrecv irrecv(RECV_PIN);
decode_results results;
#define IR_Go       0x80BF7B84
#define IR_Back     0x80BF6B94
#define IR_Left     0x80BFDB24
#define IR_Right    0x80BF1BE4
#define IR_Stop     0x80BF5BA4
#define Lpwm_pin  5     //adjusting speed
#define Rpwm_pin  6    //adjusting speed //
int pinLB=2;             // defining pin2 left rear
int pinLF=4;             // defining pin4 left front
int pinRB=7;             // defining pin7 right rear
int pinRF=8;             // defining pin8 right front
unsigned char Lpwm_val =150;
unsigned char Rpwm_val = 150;
int Car_state=0;
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT); // pin2
  pinMode(pinLF,OUTPUT); // pin4
  pinMode(pinRB,OUTPUT); // pin7
```

```
  pinMode(pinRF,OUTPUT); // pin8
  pinMode(Lpwm_pin,OUTPUT); // pin5 (PWM)
  pinMode(Rpwm_pin,OUTPUT); // pin6 (PWM)
}
void Set_Speed(unsigned char Left,unsigned char Right)
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}

void advance()      // going forward
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 1;
    }

void turnR()
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW );
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
     Car_state = 4;
    }
void turnL()
    {
     digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 3;
    }
void stopp()          //stop
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,HIGH);
     Car_state = 5;
    }
void back()           //back up
    {
     digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
```

```
      Car_state = 2;
     }

void IR_Control(void)
{
   unsigned long Key;
   if(irrecv.decode(&results)) //judging if serial port receives data
 {
     Key = results.value;
     switch(Key)
      {
        case IR_Go:advance();    //UP
        break;
        case IR_Back: back();    //back
        break;
        case IR_Left:turnL();    //Left
        break;
        case IR_Right:turnR(); //Righ
        break;
        case IR_Stop:stopp();    //stop
        break;
        default:
        break;
      }
      irrecv.resume(); // Receive the next value
     }

}
void setup()
{
   M_Control_IO_config();
   Set_Speed(Lpwm_val,Rpwm_val);
   irrecv.enableIRIn(); // Start the receiver
   Serial.begin(9600);
   stopp();
}
void loop()
{

   IR_Control();


}
```

5.5.2 Mode manuel Bluetooth

Pour établir une communication avec notre carte Arduino, nous utiliserons une liaison Bluetooth en utilisant un module HC-06 en mode esclave. Cela permettra de connecter la carte Arduino à d'autres systèmes tels que des smartphones, des ordinateurs ou d'autres microcontrôleurs, afin d'échanger des données dans les deux sens.
On configure le module Bluetooth Avec le robot selon le montage suivant :(voir figure 5.22)

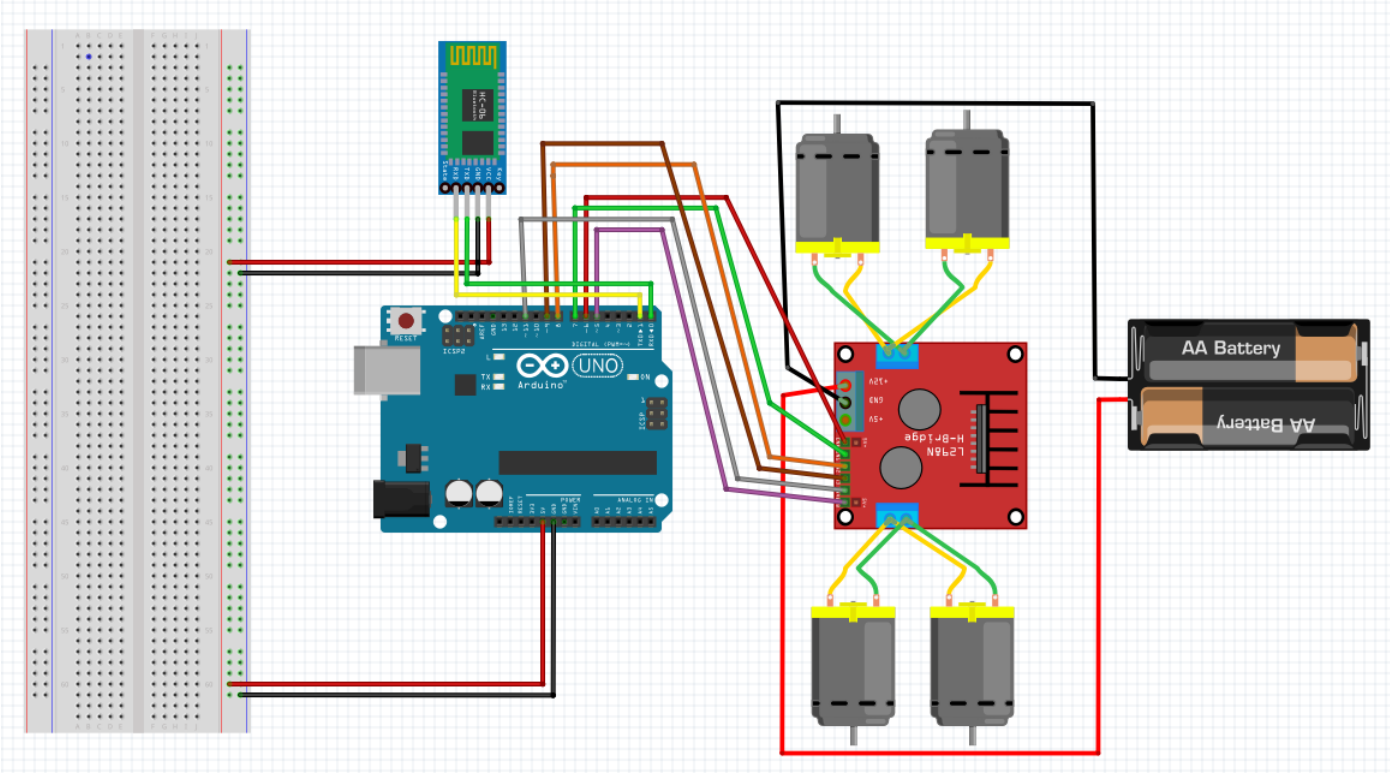

FIGURE 5.22 – Schéma de câblage du mode BLUETOOTH par frizting

Principe de fonctionnement :

- Le module HC-06 est connecté à l'Arduino, établissant ainsi une communication sans fil via Bluetooth.
- Une application installée sur un smartphone est utilisée pour envoyer des commandes de contrôle au robot.
- L'Arduino reçoit les commandes via la connexion Bluetooth et les interprète
- En fonction des commandes reçues, l'Arduino active les moteurs appropriés pour déplacer le robot selon les instructions

Organigramme mode bluetooth

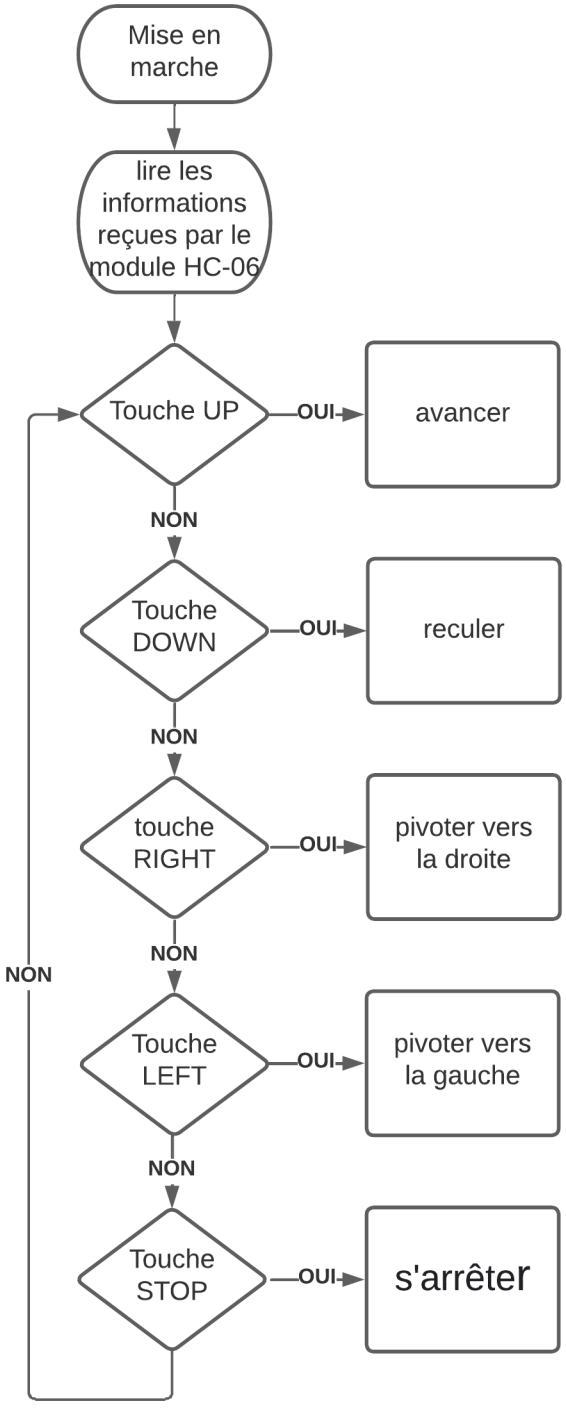

FIGURE 5.23 – Organigramme mode bluetooth

Le code :

```
unsigned char Bluetooth_val; //defining variable val
#define Lpwm_pin  5
#define Rpwm_pin  6
int pinLB=2;
int pinLF=4;
int pinRB=7;
int pinRF=8;
unsigned char Lpwm_val = 180;
unsigned char Rpwm_val = 180;
int Car_state=0;
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT); // pin 2
  pinMode(pinLF,OUTPUT); // pin 4
  pinMode(pinRB,OUTPUT); // pin 7
  pinMode(pinRF,OUTPUT); // pin 8
  pinMode(Lpwm_pin,OUTPUT); // pin 5 (PWM)
  pinMode(Rpwm_pin,OUTPUT); // pin 6 (PWM)
}
void Set_Speed(unsigned char Left,unsigned char Right)
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance()      //  going forward
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 1;
    }
void turnR()        //turning right(dual wheel)
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW );
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
     Car_state = 4;
    }
void turnL()        //turning left(dual wheel)
    {
      digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 3;
    }
```

```
void stopp()          //stop
    {

     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,HIGH);
     Car_state = 5;
    }
void back()           //back up
    {
     digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
     Car_state = 2;
    }

void setup()
{
   M_Control_IO_config();
   Set_Speed(Lpwm_val,Rpwm_val);
   Serial.begin(9600);    //initialized serial port
   stopp();
}
void loop()
{
   if(Serial.available()) //to judge whether the serial port receives the data.
    {
     Bluetooth_val=Serial.read();  //reading (Bluetooth) data of serial port
    switch(Bluetooth_val)
     {
       case 'U':advance(); //UP
       break;
       case 'D': back();    //back
       break;
       case 'L':turnL();    //Left
       break;
       case 'R':turnR();   //Right
       break;
       case 'S':stopp();     //stop
       break;
     }
    }
}
```

5.5.3 Mode autonome

Mode suiveur de ligne

Le fonctionnement du robot suiveur de ligne repose sur la détection et le suivi d'une ligne tracée sur une surface. Pour cela, des capteurs KY-033 sont utilisés pour détecter la présence de la ligne. Dans notre cas, nous avons utilisé trois capteurs positionnés sous le châssis du robot. Chaque capteur fournit une valeur binaire (0 ou 1) pour indiquer si la ligne est détectée ou non.
Le processus de fonctionnement du robot suiveur de ligne peut être résumé comme suit :
Lecture des capteurs de ligne : Les capteurs sont lus en continu à intervalles réguliers. Chaque capteur fournit une valeur binaire pour indiquer la détection de la ligne.
Traitement des données des capteurs : Les valeurs des capteurs sont analysées pour déterminer la position du robot par rapport à la ligne. En fonction de la combinaison des valeurs des capteurs, différentes actions sont déclenchées pour maintenir le robot sur la ligne.
Contrôle des moteurs : Les actions générées par l'analyse des données des capteurs sont utilisées pour contrôler les moteurs des roues du robot. En ajustant les vitesses et les directions des moteurs, le robot est capable de suivre précisément et en douceur la ligne.

Analyse fonctionnelle

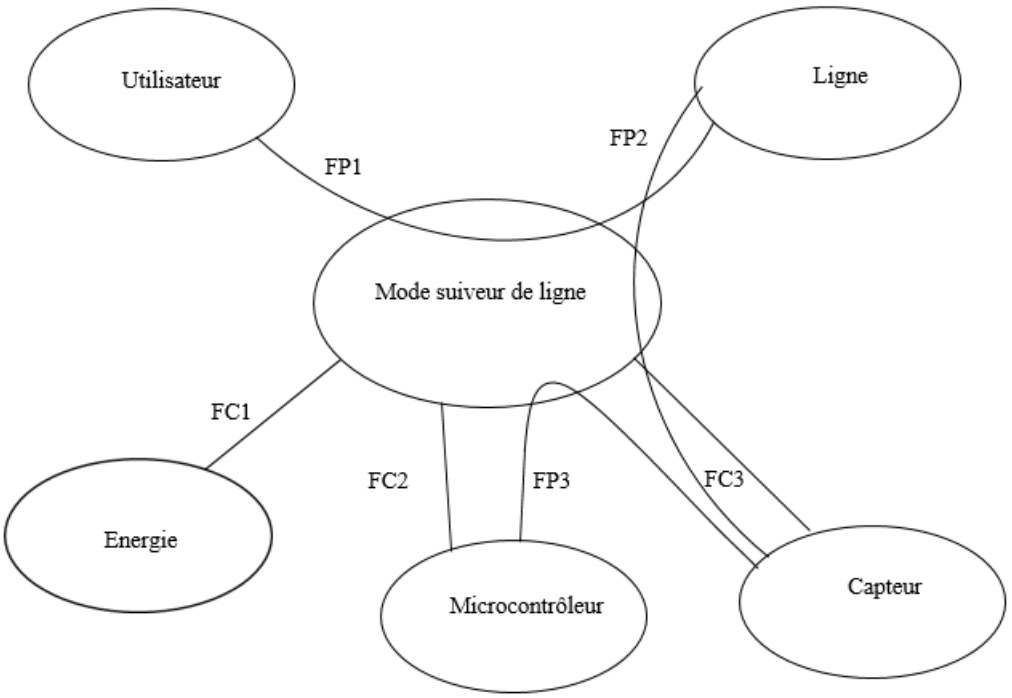

FIGURE 5.24 – Graphe des intéracteurs du mode suiveur de ligne

Fp : fonction principale
Fc : fonction complémentaire ou contrainte

Fonction	Description
FP1	Suivre la trajectoire pour servir l'utilisateur
FP2	Détecter la ligne
FP3	Analyser les données du capteur
FC1	Utiliser une alimentation comme source d'énergie
FC2	Commander le déplacement du robot
FC3	Indiquer l'existence de la ligne

TABLE 5.4 – Description des fonctions

Organigramme du mode suiveur de ligne

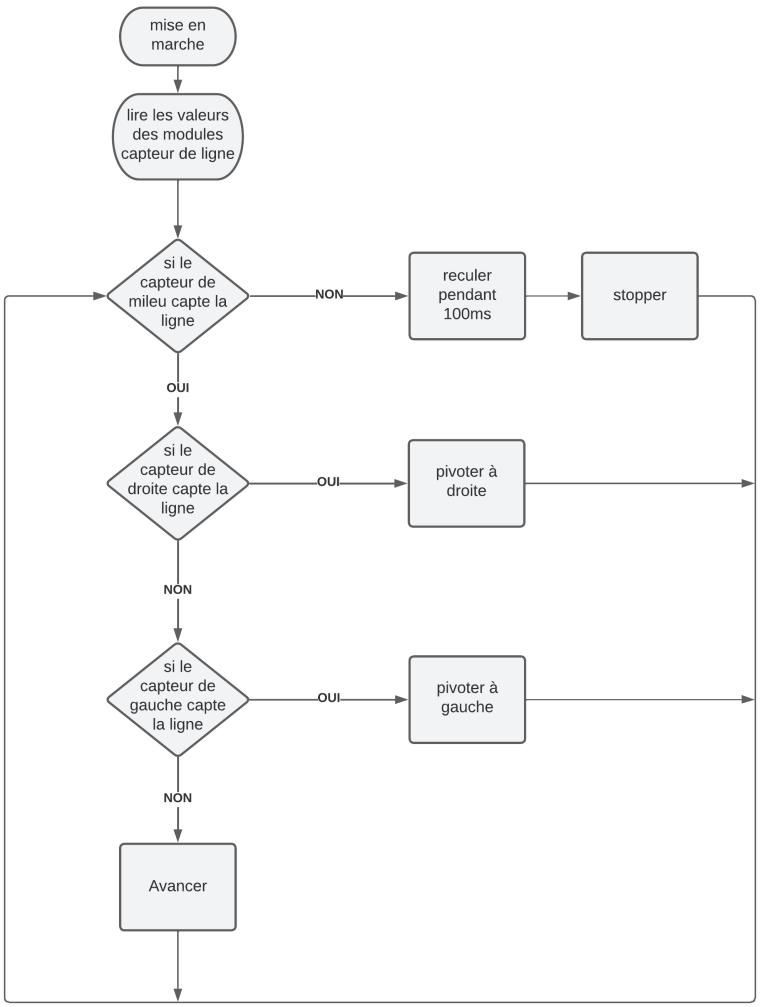

FIGURE 5.25 – Organigramme de mode suiveur de ligne

Le schéma de câblage :

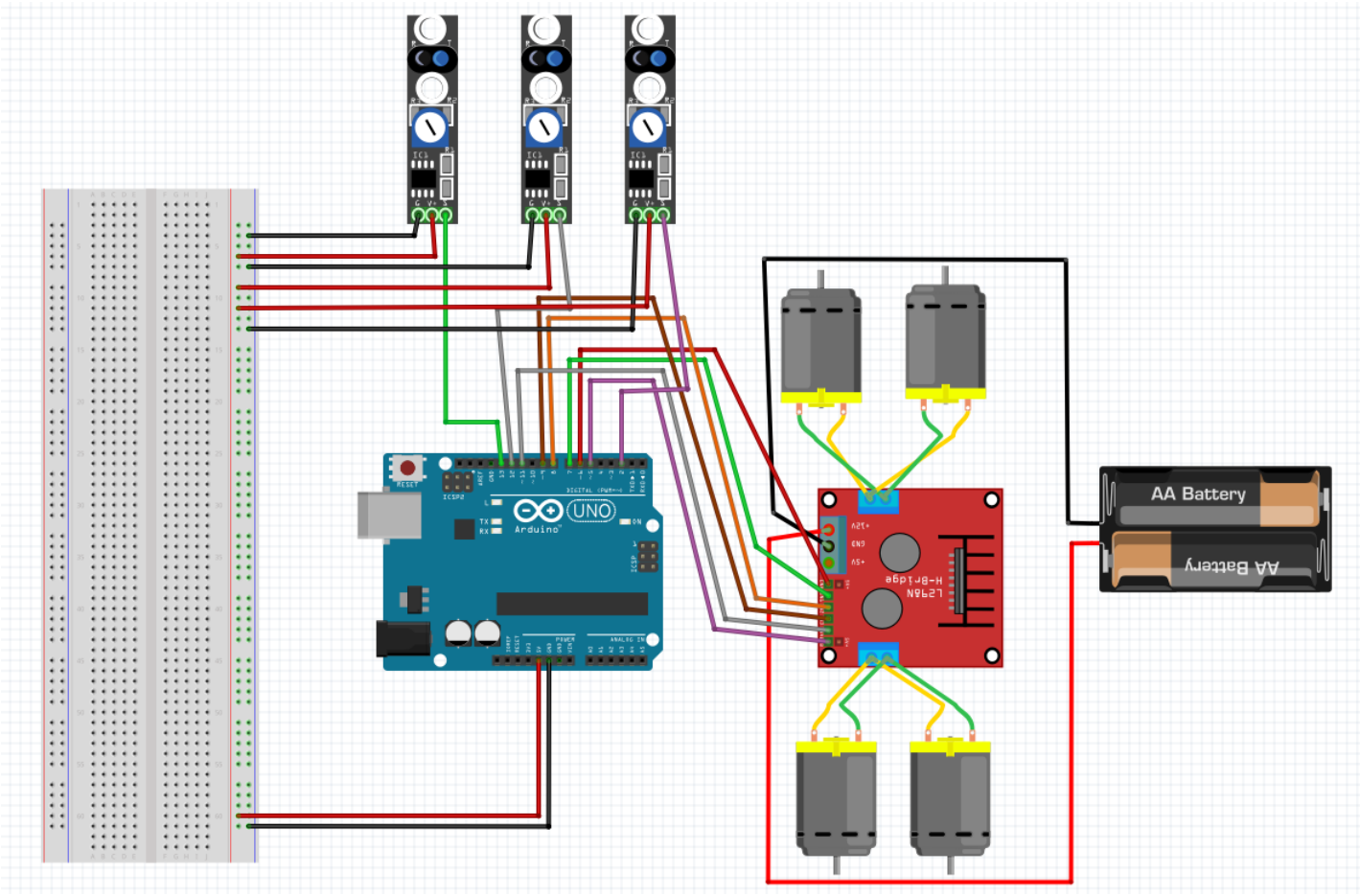

FIGURE 5.26 – Schéma de câblage du mode suiveur de ligne

Le code :

```
#define SensorLeft     9    //input pin of left sensor
#define SensorMiddle   10    //input pin of middle sensor
#define SensorRight    11    //input pin of right sensor
unsigned char SL;           //state of left sensor
unsigned char SM;           //state of middle sensor
unsigned char SR;           //state of right sensor
#define Lpwm_pin  5
#define Rpwm_pin  6
int pinLB=2;
int pinLF=4;
int pinRB=7;
int pinRF=8;
unsigned char Lpwm_val =120;
unsigned char Rpwm_val = 120;
int Car_state=0;                //state of car moving

void Sensor_IO_Config()
```

```
{
  pinMode(SensorLeft,INPUT);
  pinMode(SensorMiddle,INPUT);
  pinMode(SensorRight,INPUT);
}
void Sensor_Scan(void)
{
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
}
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT); // pin 2---IN1 of motor driver board
  pinMode(pinLF,OUTPUT); // pin 4---IN2 of motor driver board
  pinMode(pinRB,OUTPUT); // pin 7---IN3 of motor driver board
  pinMode(pinRF,OUTPUT); // pin 8---IN4 of motor driver board
  pinMode(Lpwm_pin,OUTPUT); // pin 5  (PWM) ---ENA of motor driver board
  pinMode(Rpwm_pin,OUTPUT); // pin 6 (PWM) ---ENB of motor driver
}
void Set_Speed(unsigned char Left,unsigned char Right)
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance()      // going forwards
    {
      digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 1;
    }
void turnR() //turning on the right(dual wheels)
    {
    digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW );
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
     Car_state = 4;
    }
void turnL()//turning on the left(dual wheels)
    {
    digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 3;
    }
```

```
void stopp()          //stop
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,HIGH);
     Car_state = 5;
    }
void back()           //back
    {
     digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
     Car_state = 2;
    }

void setup()
{
   Sensor_IO_Config();
   M_Control_IO_config();
   Set_Speed(Lpwm_val,Rpwm_val);
   stopp();
}
unsigned char old_SL,old_SM,old_SR;
void loop()
{
  Sensor_Scan();
 if (SM == HIGH)
{
if (SL == LOW & SR == HIGH)
{
turnR();
}
else if (SR == LOW & SL == HIGH)
{
turnL();
}
else // white on both sides, going forward
{
advance();
}
}
else // middle sensor on white area
{
if (SL== LOW & SR == HIGH)
{
turnR();
}
```

```
else if (SR == LOW & SL == HIGH)
{
turnL();
}
else // all white, stop
{
back();
delay(100);
stopp() ;
}
}
}
```

Mode détecteur d'obstacle

Le robot détecteur d'obstacles fonctionne en détectant et en réagissant aux objets présents dans son environnement. Pour cela, un capteur ultrason SR-04 est utilisé pour mesurer la distance entre le robot et les objets environnants. Un mécanisme de rotation, contrôlé par un servo-moteur, permet au capteur de balayer une zone à 180 degrés.
Le processus de fonctionnement du robot détecteur d'obstacles peut être décrit comme suit :

- Mesure de la distance : Le capteur ultrason SR-04 émet des signaux ultrasoniques et mesure le temps nécessaire pour que ces signaux rebondissent sur les objets et reviennent au capteur. La distance entre le robot et l'obstacle est calculée en fonction de ce temps de vol
- Réaction aux obstacles : En fonction de la distance mesurée, le robot prend des décisions pour éviter les obstacles. Par exemple, si la distance est inférieure à une valeur prédéfinie, le robot effectue une manœuvre d'évitement en tournant dans une direction spécifique afin d'éviter la collision.

- Contrôle du servo-moteur : Le servo-moteur est utilisé pour faire pivoter le capteur ultrasonique sur une plage de 180 degrés. Cela permet au robot de balayer son environnement et de détecter les obstacles dans différentes directions.

Analyse fonctionnelle

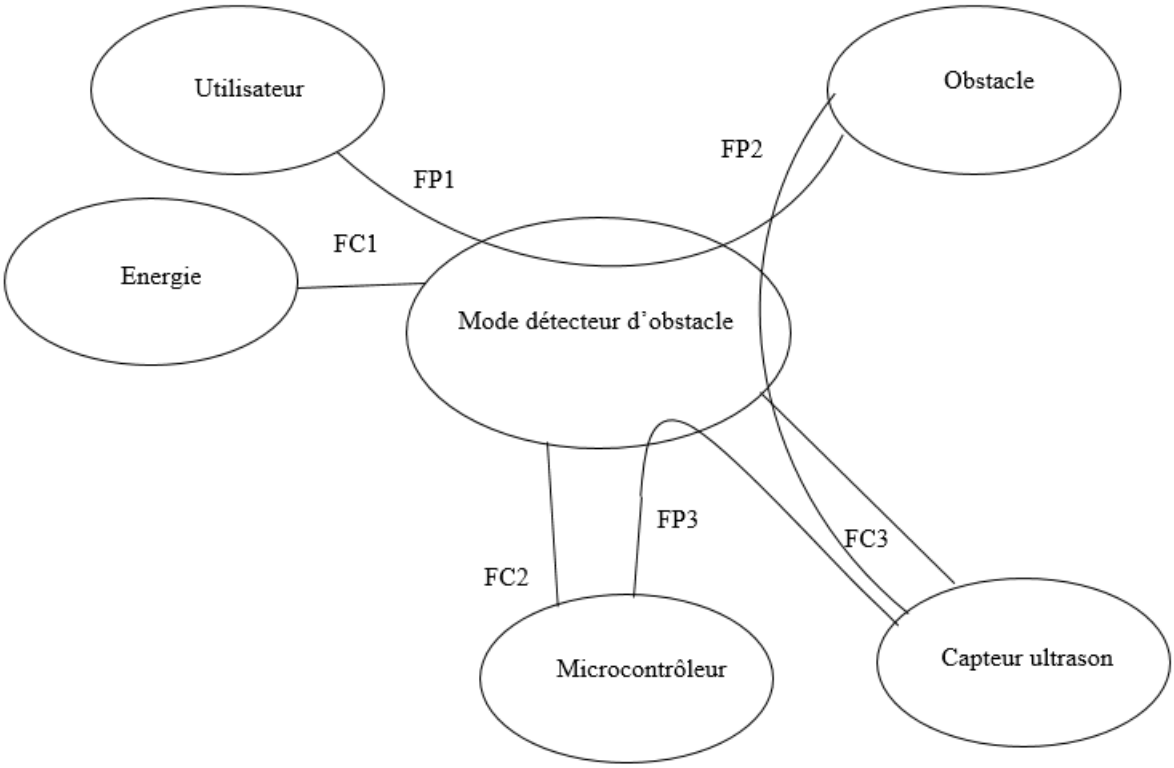

FIGURE 5.27 – Graphe des intéracteurs du mode détecteur d'obstacle

Fp : fonction principale
Fc : fonction complémentaire ou contrainte

Fonction	Description
FP1	Détecter les obstacles et les éviter afin de servir l'utilisateur
FP2	Détecter l'obstacle
FP3	Analyser les données du capteur
FC1	Utiliser une alimentation comme source d'énergie
FC2	Commander le déplacement du robot
FC3	Indiquer l'existence d'un obstacle

TABLE 5.5 – Description des fonctions

Organigramme du mode détecteur d'obstacle

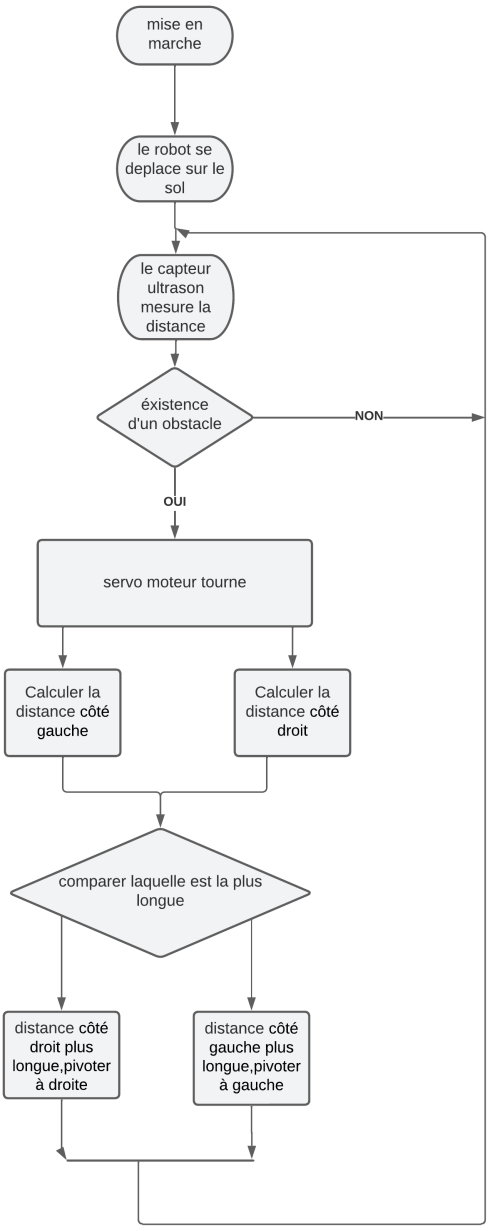

FIGURE 5.28 – Organigramme du mode détecteur d'obstacle

Le schéma de câblage :

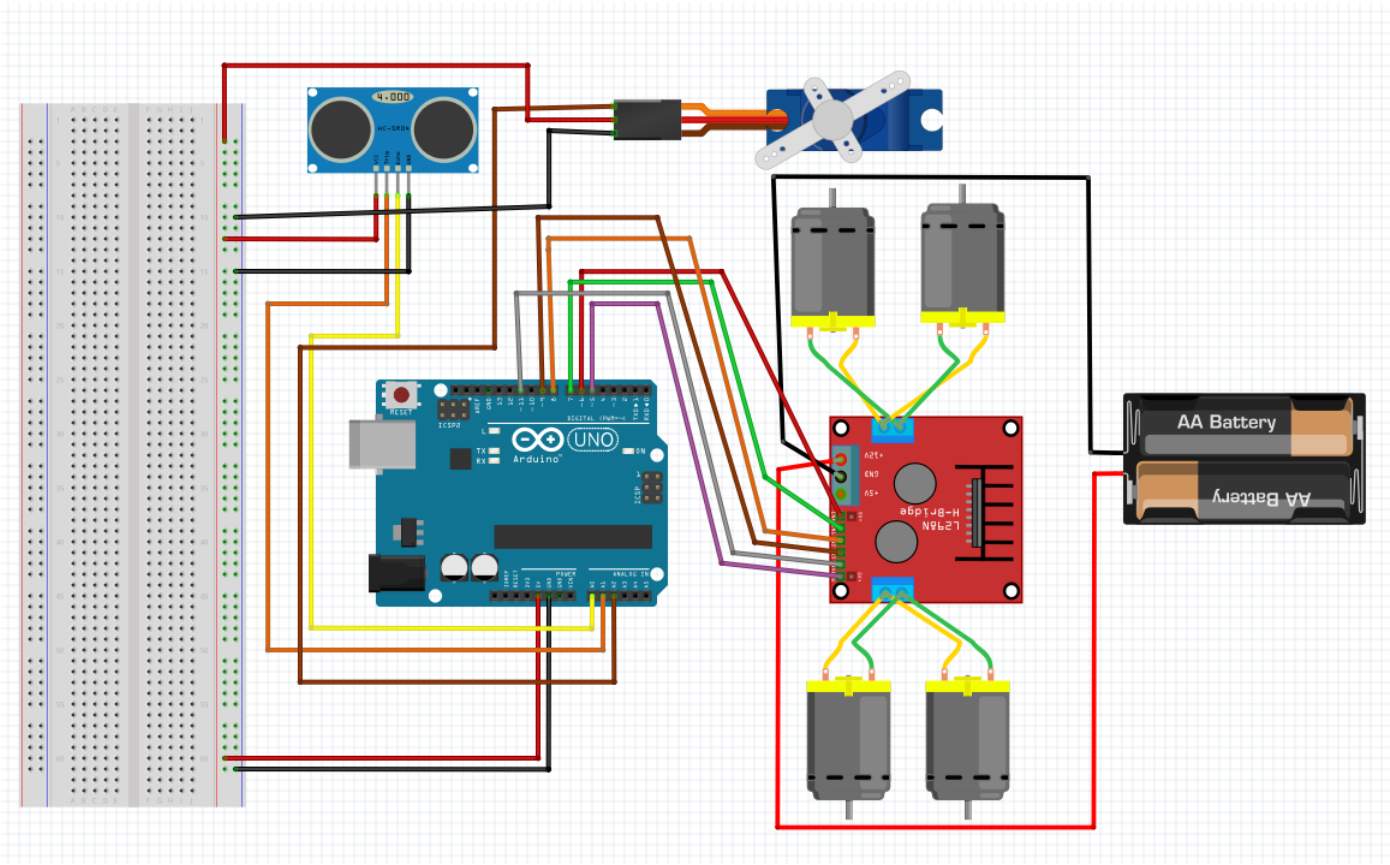

FIGURE 5.29 – Schéma de câblage du mode détecteur d'obstacle

Le code :

```
#include <Servo.h>
Servo myservo;
int inputPin=A0;   // ultrasonic module   ECHO to A0
int outputPin=A1;   // ultrasonic module   TRIG to A1
#define Lpwm_pin  5
#define Rpwm_pin  6
int pinLB=2;
int pinLF=4;
int pinRB=7;
int pinRF=8;
unsigned char Lpwm_val = 120;
unsigned char Rpwm_val = 120;
int Car_state=0; //the working state of car
int myangle; //defining variable of angle
int pulsewidth; //defining variable of pulse width
unsigned char DuoJiao=90;
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT); // /pin 2
```

```
  pinMode(pinLF,OUTPUT); // pin 4
  pinMode(pinRB,OUTPUT); // pin 7
  pinMode(pinRF,OUTPUT);  // pin 8
  pinMode(Lpwm_pin,OUTPUT);  // pin 5 (PWM)
  pinMode(Rpwm_pin,OUTPUT);  // pin 6(PWM)
}
void Set_Speed(unsigned char Left,unsigned char Right)
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance()
    {
       digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 1;
    }
void turnR()
    {
    digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW );
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
     Car_state = 4;
    }
void turnL()
    {
     digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);
     Car_state = 3;
    }
void stopp()
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,HIGH);
     Car_state = 5;
    }
void back()
    {
      digitalWrite(pinRB,LOW);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,LOW);
     digitalWrite(pinLF,HIGH);
```

```
      Car_state = 2;
     }



void Self_Control(void)//self-going
{
    int H;
    myservo.write(DuoJiao);
    H = Ultrasonic_Ranging(1);
    delay(300);
    if(Ultrasonic_Ranging(1) < 15)
    {
        stopp();
        delay(100);
        back();
        delay(50);
     }

   if(Ultrasonic_Ranging(1) < 30)
       {
         stopp();
         delay(100);
         myservo.write(0);
         int L = ask_pin_L(2);
         delay(300);
          myservo.write(180);;
         int R = ask_pin_R(3);
         delay(300);

         if(ask_pin_L(2) > ask_pin_R(3))
         {
          back();
         delay(100);
         turnL();
        delay(400);
        stopp();
        delay(50);
        myservo.write(DuoJiao);
        H = Ultrasonic_Ranging(1);
        delay(500);
         }

       if(ask_pin_L(2)  <= ask_pin_R(3))
       {
        back();
        delay(100);
        turnR();
        delay(400);
```

```
        stopp();
        delay(50);
        myservo.write(DuoJiao);
        H = Ultrasonic_Ranging(1);
        delay(300);
         }
         if (ask_pin_L(2)  < 35 && ask_pin_R(3)< 35)
         {
        stopp();
        delay(50);
        back();
        delay(50);
         }
       }
       else
       {
       advance();
       }
}
int Ultrasonic_Ranging(unsigned char Mode)
{
  int old_distance;
  digitalWrite(outputPin, LOW);
  delayMicroseconds(2);
  digitalWrite(outputPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(outputPin, LOW);
  int distance = pulseIn(inputPin, HIGH);
  distance= distance/58;
  if(Mode==1){
         Serial.print("\n H = ");
         Serial.print(distance,DEC);
         return distance;
  }
   else  return distance;
}
int ask_pin_L(unsigned char Mode)
  {
  int old_Ldistance;
  digitalWrite(outputPin, LOW);
  delayMicroseconds(2);
  digitalWrite(outputPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(outputPin, LOW);
  int Ldistance = pulseIn(inputPin, HIGH);
  Ldistance= Ldistance/58;
  if(Mode==2){
         Serial.print("\n L = ");
         Serial.print(Ldistance,DEC);
```

```
        return Ldistance;
  }
   else  return Ldistance;
}
int ask_pin_R(unsigned char Mode)
   {
  int old_Rdistance;
  digitalWrite(outputPin, LOW);
  delayMicroseconds(2);
  digitalWrite(outputPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(outputPin, LOW);
  int Rdistance = pulseIn(inputPin, HIGH);
  Rdistance= Rdistance/58;
  if(Mode==3){
        Serial.print("\n R = ");
        Serial.print(Rdistance,DEC);
        return Rdistance;
  }
   else  return Rdistance;
}

void setup()
{
   myservo.attach(A2);
   M_Control_IO_config();
   Set_Speed(Lpwm_val,Rpwm_val);
   myservo.write(DuoJiao);
   pinMode(inputPin, INPUT);
   pinMode(outputPin, OUTPUT);
   Serial.begin(9600);
   stopp();
   delay(1000);
}
void loop()
{

       Self_Control();
}
```

Mode parking autonome

Le robot va s'avancer et grâce aux capteurs ultrason, il va calculer la distance de la place et va choisir le mode de parking.
Le fonctionnement de ce projet repose sur les éléments suivants :

- Utilisation de capteurs ultrasoniques : Des capteurs ultrasoniques sont utilisés pour détecter les obstacles autour du véhicule. Ils sont placés à l'avant et à l'arrière de la voiture afin de mesurer la distance entre le véhicule et les obstacles environnants.
- Détection d'une place de stationnement : Le véhicule est équipé de capteurs ultrasoniques à l'avant qui sont utilisés pour détecter une place de stationnement disponible sur le côté de la route. Le programme Arduino analyse les mesures de distance obtenues par les capteurs afin d'identifier une place suffisamment grande pour garer le véhicule.
- Algorithme de stationnement : Une fois qu'une place de stationnement adéquate est détectée, l'algorithme de stationnement est activé. Le véhicule effectue une série de mouvements en avant et en arrière tout en ajustant son angle pour s'aligner correctement avec la place de stationnement
- Contrôle des moteurs : Le mouvement du véhicule est contrôlé à l'aide de moteurs à courant continu connectés à la carte Arduino. Le programme Arduino envoie les commandes appropriées aux moteurs pour effectuer les mouvements nécessaires pendant le processus de stationnement.
- Arrêt automatique : Une fois que le véhicule est correctement garé en créneau, le processus de stationnement s'arrête automatiquement.

Analyse Fonctionnelle

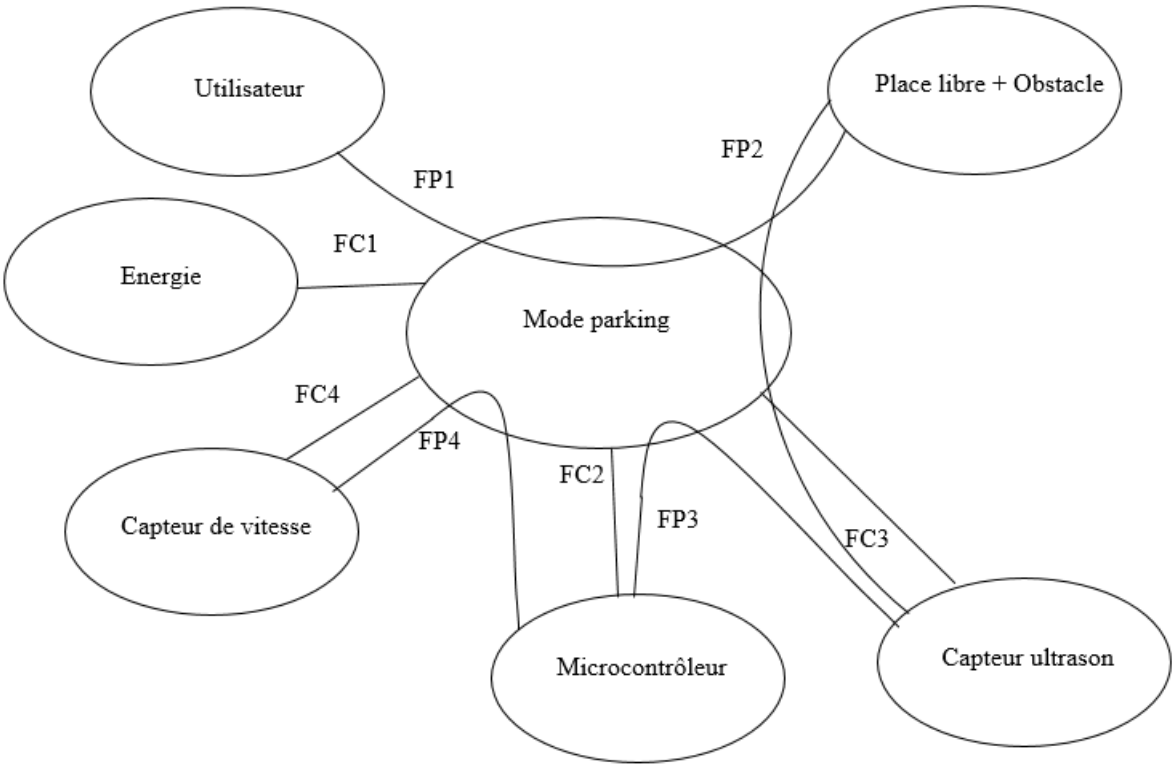

FIGURE 5.30 – Graphe des intéracteurs du mode parking autonome

Fp : fonction principales
Fc : fonction complémentaire ou contrainte

Fonction	Description
FP1	Trouver une place libre, mesurer sa distance et effectuer un type de stationnement selon cette distance afin de servir l'utilisateur
FP2	Détecter les obstacles et les places libres
FP3	Analyser les données du capteur ultrason
FP4	Analyser les données du capteur de vitesse
FC1	Utiliser une alimentation comme source d'énergie
FC2	Commander le déplacement du robot
FC3	Indiquer l'existence des places libres et des obstacles
FC4	Indiquer le changement d'état du capteur de vitesse

TABLE 5.6 – Description des fonctions

Organigramme de sélection

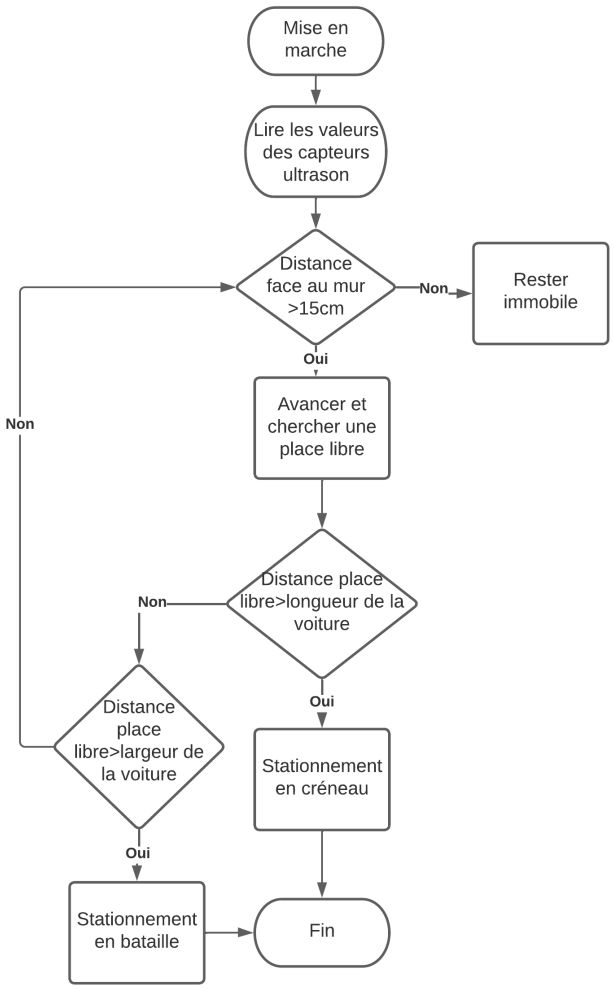

FIGURE 5.31 – Organigramme de sélection

Organigramme de stationnement en créneau

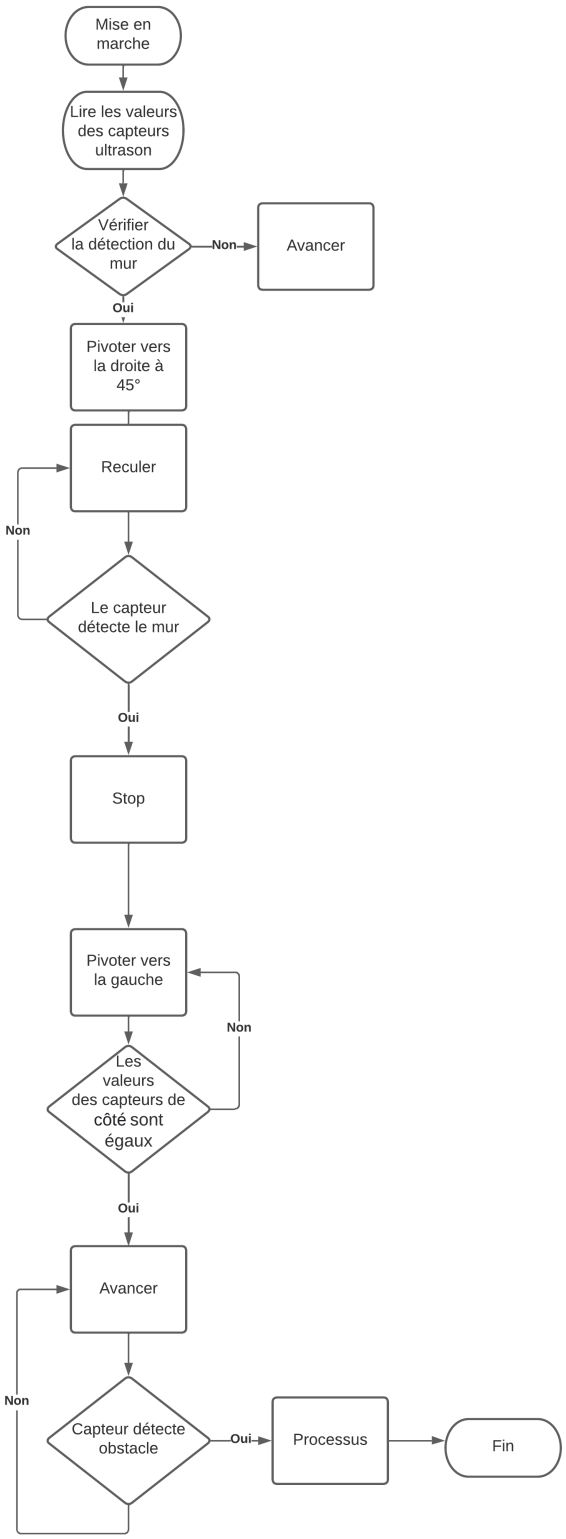

FIGURE 5.32 – Organigramme de stationnement en créneau

Organigramme de stationnement en bataille

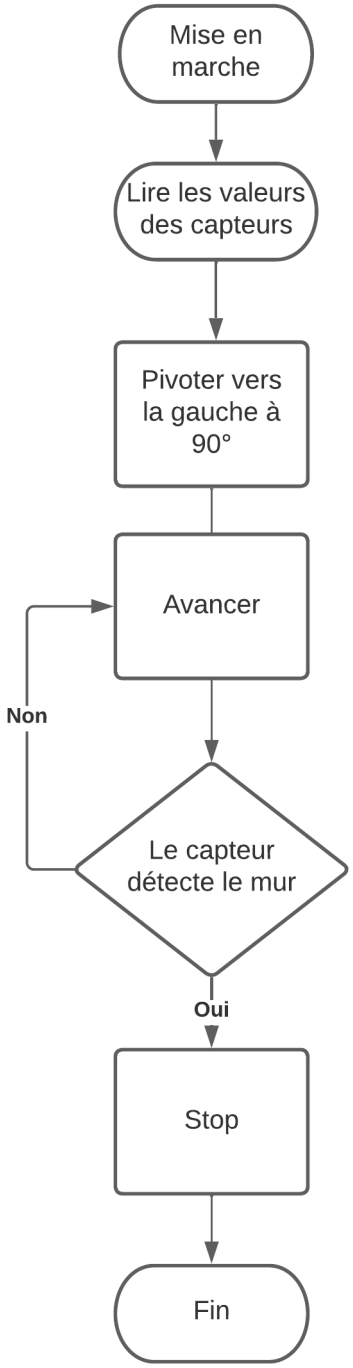

FIGURE 5.33 – Organigramme de stationnement en bataille

Le schéma de câblage :

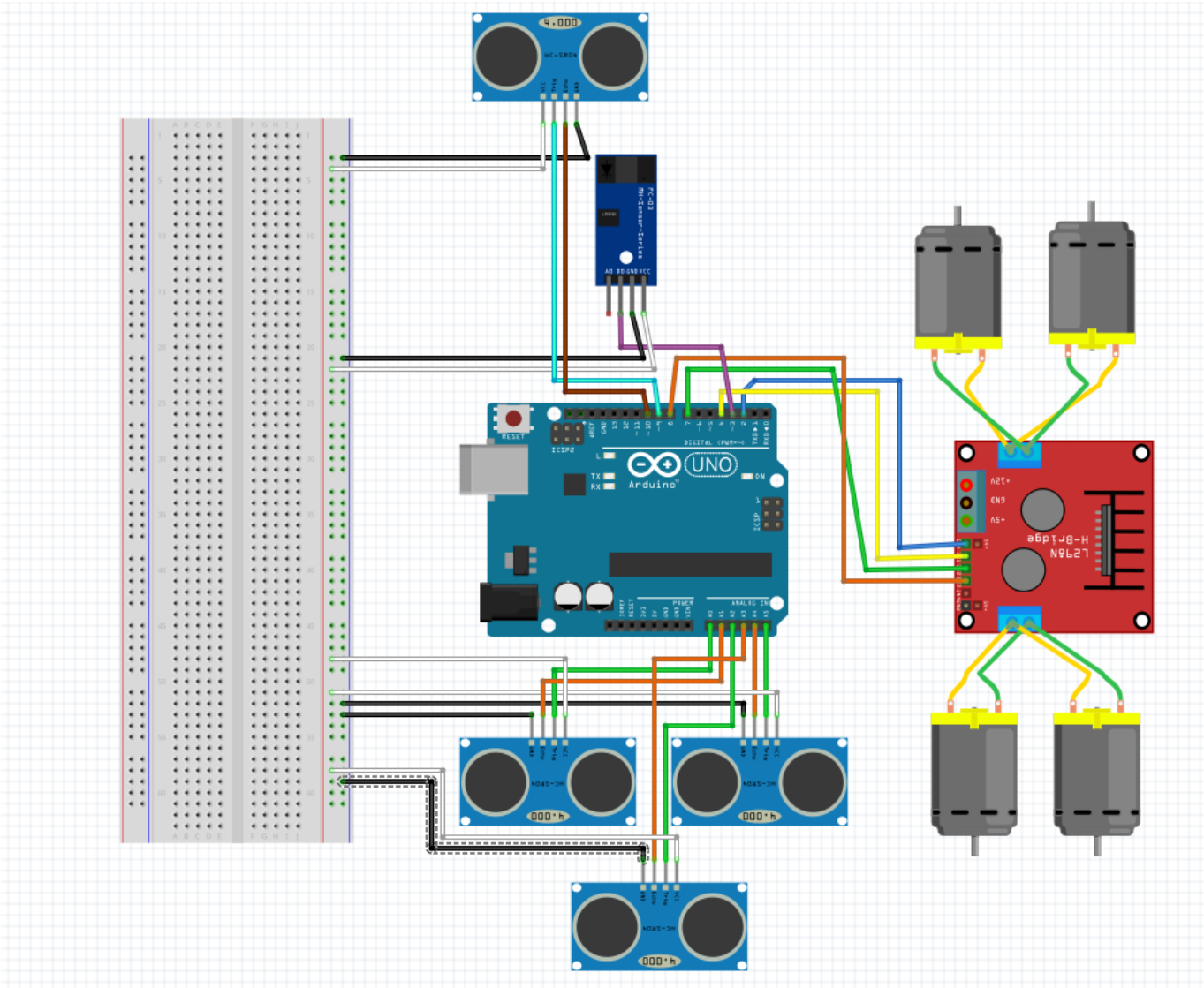

FIGURE 5.34 – Schéma de câblage du mode PARKING

Le code :

```
#include <Ultrasonic.h>
int pinLB=2;
int pinLF=4;
int pinRB=7;
int pinRF=8;
int ENA=5;
int ENB=6;


Ultrasonic ultrasonic_back(9,10),ultrasonic_left_back(A5,A4),
ultrasonic_left_on(A0,A1),ultrasonic_on(A2,A3);
// definitions of ultrasomic sensors
```

```
#define Left 0 // left direction command
#define Right 1 // right direction command
#define Forward 2 //forward direction command
#define Back 3 //reverse direction command
#define minimum_limit 15 // Width of the car (cm)
#define minimum_limit1 28 //length of the car (cm)
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT); // /pin 2 IN1
  pinMode(pinLF,OUTPUT); // pin 4 IN2
  pinMode(pinRB,OUTPUT); // pin 7 IN3
  pinMode(pinRF,OUTPUT);  // pin 8 IN4
  pinMode(ENA,OUTPUT);  // pin 5 (PWM)
  pinMode(ENB,OUTPUT);  // pin 6(PWM)
}

byte park_status = 0;
int signalalpin = 3;
volatile int val;

int counter = 0;
int current_status = 0;
int previous_status = 0;

void count(int countdir)
{
for (int i = 0 ; i <= countdir; i+1)
{
val = digitalRead(signalalpin);
if (val == LOW) {

current_status = 0;
}
else {

current_status = 1;
}

if(current_status != previous_status)
{
if(current_status == 1)
{
counter = counter + 1;
Serial.println(counter);
i = i+1;
}
else
{
i = i ;
```

```
}

previous_status = current_status;

}
if (i == countdir)
{

 digitalWrite(pinRB,HIGH);
 digitalWrite(pinRF,HIGH);
 digitalWrite(pinLB,HIGH);
 digitalWrite(pinLF,HIGH);

}

}

}

void motor_pinSetup()
{

 digitalWrite(pinRB,HIGH);
 digitalWrite(pinRF,HIGH);
 digitalWrite(pinLB,HIGH);
 digitalWrite(pinLF,HIGH);
}

// Motion functions
void Robot_Move(byte motor, byte spd)
{
if (motor == Forward)
{
 digitalWrite(pinRB,HIGH);
      digitalWrite(pinRF,LOW);
      digitalWrite(pinLB,HIGH);
      digitalWrite(pinLF,LOW);
      analogWrite(ENA,spd);
      analogWrite(ENB,spd);

}
if (motor == Back)
{
digitalWrite(pinRB,LOW);
      digitalWrite(pinRF,HIGH);
      digitalWrite(pinLB,LOW);
      digitalWrite(pinLF,HIGH);
      analogWrite(ENA,spd);
      analogWrite(ENB,spd);
```

```
}
if (motor == Left)
{
  digitalWrite(pinRB,LOW);
      digitalWrite(pinRF,HIGH);
      digitalWrite(pinLB,HIGH);
      digitalWrite(pinLF,LOW);
        analogWrite(ENA,spd);
      analogWrite(ENB,spd);
}

if (motor == Right)
{
digitalWrite(pinRB,HIGH);
      digitalWrite(pinRF,LOW );
      digitalWrite(pinLB,LOW);
      digitalWrite(pinLF,HIGH);
      analogWrite(ENA,spd);
      analogWrite(ENB,spd);

}

}

void Robot_Stop()
{
digitalWrite(pinRB,HIGH);
      digitalWrite(pinRF,HIGH);
      digitalWrite(pinLB,HIGH);
      digitalWrite(pinLF,HIGH);

}

// Parking lot search
bool Park_Place_Control()
{

long on_Sensor = ultrasonic_on.Ranging(CM);
long right_Sensor = ultrasonic_left_on.Ranging(CM);
long right_back_Sensor = ultrasonic_left_back.Ranging(CM);

if( (right_Sensor <= minimum_limit)
&&(right_back_Sensor <= minimum_limit)&&(park_status == 0))
{
Robot_Move(Forward, 100);
park_status = 1; Serial.println(park_status);
}
```

```
if((right_Sensor > minimum_limit)
&&(right_Sensor < minimum_limit1)
&&(right_back_Sensor > minimum_limit)
&&(right_back_Sensor < minimum_limit1)
&&(park_status == 1))
{
Robot_Move(Forward, 100);
park_status = 2;Serial.println(park_status);
}

if((right_Sensor >= minimum_limit1)
&&(right_back_Sensor >= minimum_limit1)
&&(park_status == 1))
{
/* Upright Parking Decision */
Robot_Stop() ;
delay(500);
park_status = 10;Serial.println(park_status);
}

if((right_Sensor <= minimum_limit)
&&(right_back_Sensor <= minimum_limit)
&&(park_status == 2))
{
/* Parallel Parking Decision */
park_status = 3; Serial.println(park_status);
}

return park_status;
}

void Park_find()
{
Park_Place_Control();
if(park_status == 3 )
{
Robot_Stop();Serial.println(park_status);
delay(400);
park_status = 4;
}
if(park_status == 4 )
{

Robot_Move(Back,120);
count(18);
Robot_Stop();Serial.println(park_status);
delay(500);
Robot_Move(Right,150);
```

```
count(9);
Robot_Stop();
delay(500);
park_status = 5;
}
if(park_status == 5)
{

Robot_Move(Back,120);
count(18);
long back_Sensor =
ultrasonic_back.Ranging(CM);Serial.println(back_Sensor);

if(back_Sensor>0 && back_Sensor <= 13)
{
Robot_Stop();
delay(400);
park_status = 6;
}
return back_Sensor;
}

if(park_status == 6)
{
Robot_Move(Left,140);

long right_Sensor =
ultrasonic_left_on.Ranging(CM);
Serial.println(right_Sensor);
long right_back_Sensor =
ultrasonic_left_back.Ranging(CM); Serial.println(right_back_Sensor);

if(right_Sensor == right_back_Sensor)
{
Robot_Stop();
park_status = 7;
}

return right_Sensor,right_back_Sensor;
}
if(park_status == 7)
{
long on_Sensor = ultrasonic_on.Ranging(CM);

if(on_Sensor<=6)
{
Robot_Stop();
park_status = 8;
}
```

```
else
{
Robot_Move(Forward,100);
}
return on_Sensor;
}
if (park_status ==10)
{

Robot_Move(Left,180);
count(14);
Robot_Stop();
delay(500);
park_status = 7;

}

}

void setup()
{
Serial.begin(9600);
attachInterrupt(5, count, CHANGE);
pinMode (signalalpin, INPUT) ;

motor_pinSetup();
}

void loop()
{
Park_find();
}
```

5.5.4 Aperçu du robot

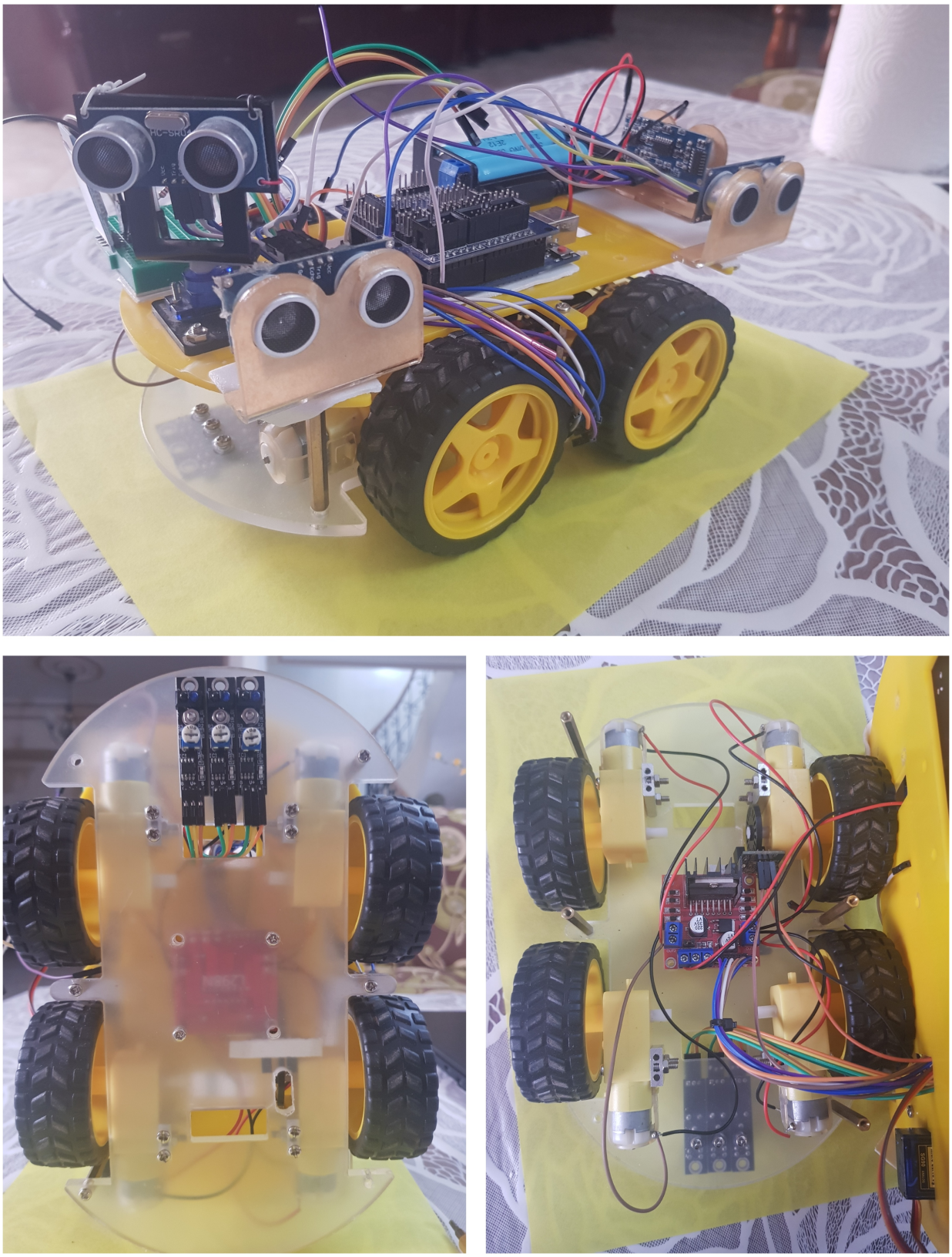

FIGURE 5.35 – Photos du robot sous différents angles

5.5.5 Exemples en images du robot mobile en mode Parking

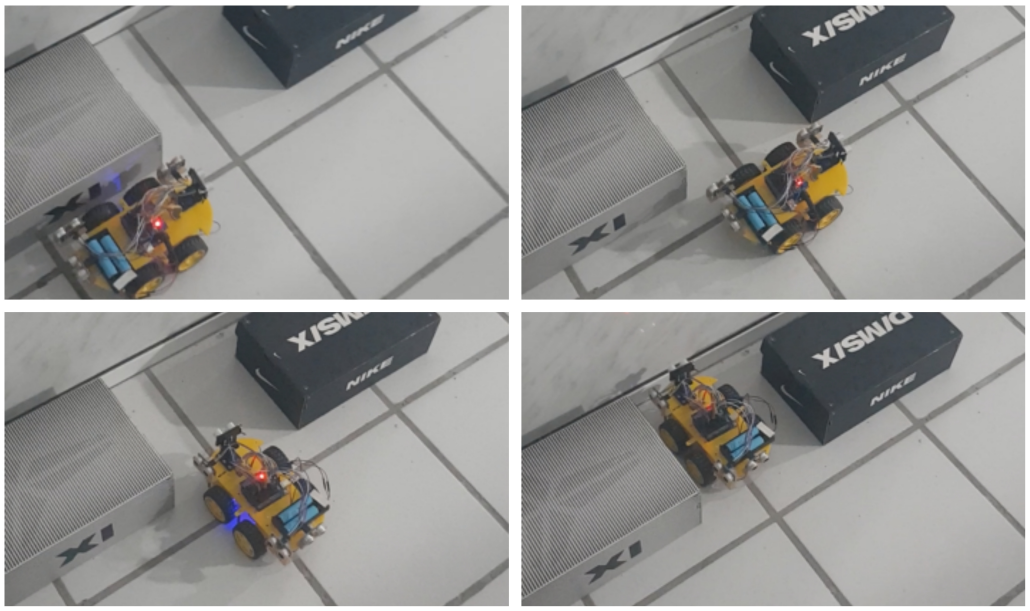

FIGURE 5.36 – Robot mobile en mode bataille

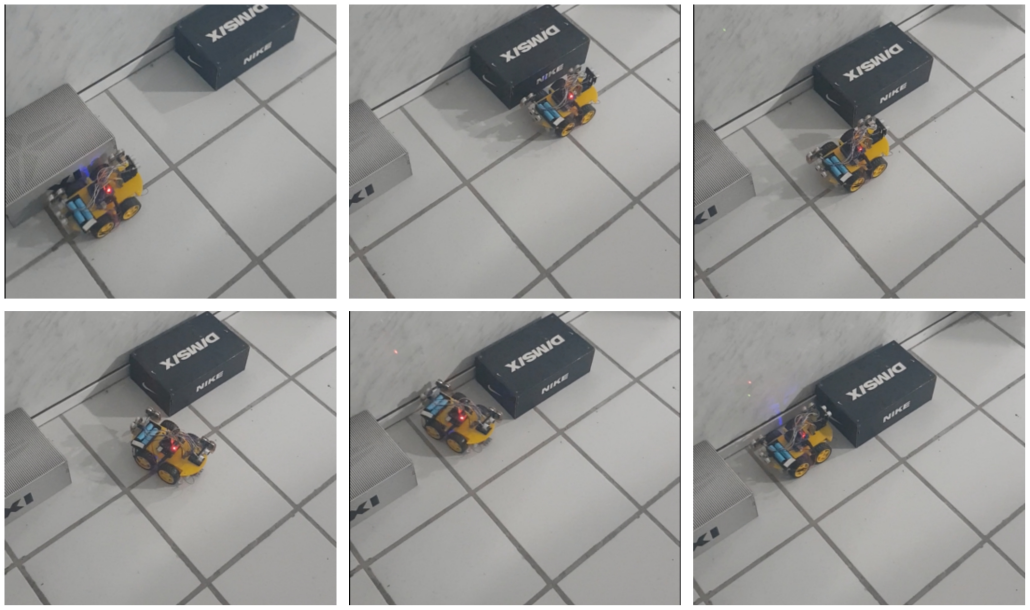

FIGURE 5.37 – Robot mobile en mode créneau

5.6 Conclusion

Ce chapitre final représente une étape cruciale de notre projet, où nous avons exposé en détail les éléments essentiels tels que le cahier des charges, le matériel et le logiciel utilisés. Nous avons également procédé à des simulations pour confirmer le bon fonctionnement de certains composants. Nous avons examiné en détail les différents modes de fonctionnement du robot, en expliquant leur principe de fonctionnement, leur structure et leur code respectif. Ces modes incluent le mode manuel, le mode suiveur de ligne, le mode détecteur d'obstacles et le mode de parking autonome. Chacun de ces modes a été développé en tenant compte des besoins matériels, des principes de fonctionnement, des algorithmes et des codes spécifiques.
Grâce à notre recherche approfondie et à nos échanges, nous avons réussi à concevoir un robot mobile polyvalent qui répond aux différentes exigences et demandes spécifiques du projet. Notre expertise en automatisation, en robotique mobile et en programmation Arduino nous a permis de créer un robot autonome performant et adaptable.

Conclusion générale

Dans ce projet de mémoire, nous avons entrepris l'étude et le développement d'un robot mobile polyvalent doté de plusieurs modes de fonctionnement. Grâce à des recherches approfondies et à des discussions approfondies, nous avons réussi à concevoir un robot capable de réaliser diverses tâches. Nous avons commencé par examiner les principes fondamentaux de la robotique mobile, en comprenant les différents éléments et systèmes nécessaires à la construction d'un robot fonctionnel. Nous avons accordé une attention particulière à l'importance des microcontrôleurs, notamment l'Arduino, dans le contrôle et la coordination des mouvements du robot.
Par la suite, nous avons abordé les différents modes de fonctionnement de notre robot, à savoir le mode manuel, le mode suiveur de ligne, le mode détecteur d'obstacles et le mode de parking autonome. Chaque mode a été minutieusement étudié et implémenté en utilisant des composants spécifiques et des algorithmes appropriés.
Dans le mode manuel, nous avons mis en place une connexion Bluetooth avec une application mobile et une communication infrarouge entre le robot et une télécommande grâce a un récepteur. Cela permet à l'utilisateur de contrôler le robot en envoyant des commandes via la télécommande.
En ce qui concerne le mode suiveur de ligne, nous avons utilisé des capteurs de suivi de ligne afin de permettre au robot de suivre une trajectoire prédéfinie avec précision. Les données collectées par ces capteurs ont été traitées à l'aide d'algorithmes pour ajuster et maintenir la trajectoire du robot.
Le mode détecteur d'obstacles a donné au robot la capacité de détecter et d'éviter les obstacles grâce à des capteurs d'ultrasons ou des capteurs infrarouges. Des algorithmes de détection d'obstacles ont été mis en place pour prendre des décisions en temps réel et permettre au robot de contourner les obstacles de manière autonome
Enfin, nous avons implémenté le mode de parking autonome en combinant des capteurs d'ultrasons avec des algorithmes spécifiques de stationnement. Le robot était ainsi en mesure de rechercher une place de stationnement approprié et de s'y garer de manière autonome. Le parking a été effectué sans erreurs grâce au capteur de vitesse Grâce à ces différents modes de fonctionnement, notre robot mobile polyvalent a démontré une grande adaptabilité à diverses situations, que ce soit pour des applications de divertissement, d'exploration ou même de tâches pratiques. Il offre une solution complète et autonome.
Ce projet de mémoire met en évidence l'importance cruciale de la conception et de la programmation dans le domaine de la robotique. En explorant diverses fonctionnalités et en développant des compétences techniques, nous avons réussi à concevoir un robot mobile polyvalent capable de répondre à différentes exigences et de fonctionner de manière autonome.
En conclusion, ce projet de robot mobile multitâche constitue une contribution significative au domaine de la robotique. Il démontre le potentiel des systèmes autonomes en offrant des fonctionnalités polyvalentes pour des utilisations pratiques et ludiques. Il ouvre également la voie à de futurs développements dans le domaine de la robotique et de l'intelligence artificielle, avec des possibilités d'amélioration et d'expansion.

Bibliographie

[1] boutique semageek motoreducteur a courant continu - "tt motor" - 200rpm - 3 a 6vdc. lien :www.boutique.semageek.com/fr/1281-motoreducteur-a-courant-continu-tt-motor-200rpm-3-a-6vdc-3006152778672.html.

[2] data sheet servo motor sg90 .lien :www.itechnofrance.wordpress.com/2013/05/05/utilisation-du-servomoteur-sg90-avec-larduino/.

[3] electroschematics motor speed sensor module circuit .lien :www.electroschematics.com/motor-speed-sensor-module-circuit/.

[4] Etude électronique de la carte arduino. https ://www.framboiseetcompagnie.fr/etude-electronique-de-la-carte-arduino/. Accessed :24 February 2021.

[5] Fritzing : le logiciel pour les makers et l'électronique (et les alternatives). https ://www.hwlibre.com/fr/fritzing/.

[6] pecquery wixsite le module bluetooth hc-06 .lien :www.pecquery.wixsite.com/arduino-passion/copie-de-le-detecteur-a-ultrasons-h-1.

[7] robot-maker capteur à ultrasons hc-sr04. lien :www.robot-maker.com/shop/capteurs/13-telemetre-a-ultrasons-hc-sr04-13.html.

[8] sensorkit ky-033 module suiveur de ligne. lien :www.sensorkit.joy-it.net/fr/sensors/ky-033.

[9] tiptopboards alimentation 9v pour carte arduino (transformateur) .lien :www.tiptopboards.com/185-alimentation-9v-pour-carte-arduino-transformateur.html.

[10] Rui Abreu, Peter Zoeteweij, and Arjan JC Van Gemund. On the accuracy of spectrum-based fault localization. In *Testing : Academic and industrial conference practice and research techniques-MUTATION (TAICPART-MUTATION 2007)*, pages 89–98. IEEE, 2007.

[11] Auday Al-Mayyahi, Weiji Wang, and Phil Birch. Design of fractional-order controller for trajectory tracking control of a non-holonomic autonomous ground vehicle. *Journal of Control, Automation and Electrical Systems*, 27 :29–42, 2016.

[12] Auday Al-Mayyahi, William Wang, and Phil Birch. Adaptive neuro-fuzzy technique for autonomous ground vehicle navigation. *Robotics*, 3(4) :349–370, 2014.

[13] Held At, Janos Sztipanovits, John A Stankovic, and David E Corman. Industry–academy collaboration in cyber physical systems (cps) research. 2009.

[14] Bernard Bayle. Robotique mobile. *Ecole Nationale Superieure de Physique de StrasbourgUniversite Louis Pasteur*, 2007, 2008.

[15] CHERIGUI Nesrine ET Dich Bochra. Mémoire de fin d'étude modélisation et simulation de la commande d'un moteur à courant continu. 2019.

[16] Jean-Louis Boimond. Cours de robotique. *ISTIA, Universite Angers*, 14 :15–16.

[17] Johann Borenstein and Liqiang Feng. Gyrodometry : A new method for combining data from gyros and odometry in mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 423–428. IEEE, 1996.

[18] Narimane Bouaziz. “classes de 2nde si-cit et de première si.”. 2019.

[19] Cyril Drocourt. Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels. *PhD diss., ANRT [diff.]*, 2002.

[20] Rafael Fierro and Frank L Lewis. Control of a nonholomic mobile robot : Backstepping kinematics into dynamics. *Journal of robotic systems*, 14(3) :149–163, 1997.

[21] Rafael Fierro and Frank L Lewis. Control of a nonholonomic mobile robot using neural networks. *IEEE transactions on neural networks*, 9(4) :589–600, 1998.

[22] David Filliat. *Robotique mobile*. PhD thesis, EDX, 2011.

[23] J. Gangloff. “robotique”.

[24] Pierrick Grandjean and A Robert de Saint Vincent. 3-d modeling of indoor scenes by fusion of noisy range and stereo data. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 681–687. IEEE, 1989.

[25] Belkhadria Khemisti. commande d'un robot mobile par réseaux de neurones artificiels. *Mémoire en vue de l'obtention du diplôme de magister en électronique. Option : Robotique*, 2014.

[26] M. LAKHDARI.F. Polycopie des travaux pratiques : Introduction à la simulation et routage des circuits avec le logiciel proteus v7 et v8. 2016.

[27] Jean-Paul Lallemand and Saïd Zeghloul. *Robotique : aspects fondamentaux : modélisation mécanique, CAO robotique, commande : avec exercices corrigés*. Masson, 1994.

[28] Stéphane Lens. Locomotion d'un robot mobile. 2008.

[29] Gilles Mauris. *Capteurs ultrasonores intelligents : application à la représentation symbolique de mesures de distance par codage flou*. PhD thesis, Chambéry, 1992.

[30] R Mermond. Planification de chemins pour un robot non-holonome sous des contraintes d'incertitudes g eom etriques. *DEA, Institut National Polytechnique de Grenoble, France*, 1996.

[31] SLIMANE Noureddine. Thèse pour l'obtention du grade de doctorat batna. 2005.

[32] Tariq Samad and AM Annaswamy. The impact of control technology. *IEEE Control Systems Society*, 1 :246, 2011.

[33] Christian Tavernier. *Arduino : Applications avancées : Claviers tactiles, télécommande par Internet, géolocalisation, applications sans fil...* Dunod, 2012.

[34] Benoît Thuilot. *Contribution à la modélisation et à la commande de robots mobiles à roues*. PhD thesis, École Nationale Supérieure des Mines de Paris, 1995.

[35] Hans-Joachim von der Hardt, Didier Wolf, and René Husson. Localization of a wheeled mobile robot using incremental odometry, a gyroscope and a magnetic compass. In *The Fourth International Conference on Control, Automation, Robotics and Vision (ICARCV'96)*, pages 1116–1120.