

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

Présenté par

BOUCHIBA Mohamed El Amine

&

MISSOUM Mustapha

Étude et réalisation d'une voiture autonome avec Arduino

Proposé par : Mme KAOULA Ikram

Année Universitaire 2021-2022

Remerciements

Nous tenons à remercier énormément le bon dieu de nous avoir donné la force et le courage d'accomplir ce travail.

On remercie également beaucoup notre promotrice Mme KAOUILA.1 pour son encadrement, sa compréhension et sa gentillesse durant toute la période de la préparation de notre mémoire.

Nous tenons aussi à remercier tous les enseignants et aussi les membres de l'administration de département d'électronique.

On remercie également les membres du jury d'avoir accepté de juger ce modeste travail.

Enfin, nous remercions tous les membres de la section << Instrumentation >> ainsi que tous nos amis de toute la faculté.

Je dédie ce modeste travail à :

Ma chère Mère,

Mon cher Père,

Mes chères Sœurs,

Mes chers frères,

Ceux qui m'aiment,

Ceux que j'aime,

Mais également :

À nos pères,

À nos mères,

Les plus exquis et les plus ingénieuses des mères,

À nos bienveillants frères et sœurs,

Nous tenons à exprimer toute notre

reconnaissance à tous nos amis,

Bouchiba Mohamed El Amine

Missoum Mustapha

الهدف من هذا المشروع هو تصميم وبناء سيارة ذاتية القيادة يمكنها التحرك دون تدخل بشري، وهي مزودة بأجهزة استشعار للتعرف على البيئة المحيطة، ومنفذات (محركات ذات التيار المستمر) لضمان الحركة في اتجاهات مختلفة.

كما أنها مجهزة بالوسائل التي تسمح لها بالتحرك بذكاء، وفقا لدرجة استقلاليتها، لتحقيق هذه الغايات، استخدمنا لوحة أردوينو ميغا، ووحدة التحكم مع المحركات، ومستشعر الموجات فوق الصوتية، ونظام تحديد المواقع العالمي لتحديد مكان الروبوت.

كلمات المفاتيح: سيارة ؛ ذاتية القيادة ؛ علم الروبوتات

Résumé :

Le but de ce projet est de faire la conception et la réalisation d'une voiture autonome, pouvant se déplacer sans intervention humaine, elle est munie de capteurs pour reconnaître l'environnement autour, et d'actionneurs (moteurs à courant continu) pour assurer le déplacement dans les différentes directions.

Elle est aussi dotée de moyens qui lui permettent de se déplacer intelligemment, suivant son degré d'autonomie, pour arriver à ces fins, on a utilisé la carte Arduino Mega, moteur driver avec les moteurs, capteur à ultrasons, et un GPS pour déterminer la localisation du robot.

Mots clés : voiture ; autonome ; robotique

Abstract :

The goal of this project is to design and build an autonomous car, able to move without human intervention, it is equipped with sensors to recognize the surrounding environment, and actuators (direct current motors) to ensure movement in different directions.

It is also equipped with means that allow it to move intelligently, according to its degree of autonomy, to achieve these ends, we used the Arduino Mega card, motor driver with the motors, ultrasonic sensor, and a GPS to determine the location of the robot.

Keywords : car ; autonomous ; robotics

Listes des acronymes et abréviations

Chapitre 1

RC : Radio Controlled

R&D : Research and Development

TMS : Troubles Musculo-Squelettiques

AI ou IA : Artificial Intelligence / Intelligence Artificielle

Chapitre 2

IDE : Integrated Development Environment

SRAM : Static Random Access Memory

EEPROM : Electrically Erasable Programmable Read Only Memory

MLI : Modulation de la Largeur d'Impulsion, aussi appelée :

PWM : Pulse Width Modulation

I²C : Inter Integrated Circuit

USART : Universal Synchronous & Asynchronous Receiver Transmitter

IoT : Internet of Things

GPS : Global Positioning System

RADAR : Radio Detection and Ranging

LIDAR : Light Detection and Ranging

Chapitre 3

NMEA : National Marine Electronics Association

IR : Infra Rouge

Table des matières

INTRODUCTION GÉNÉRALE.....	1
CHAPITRE 1 INTRODUCTION À LA ROBOTIQUE.....	3
1.1 INTRODUCTION.....	3
1.2 DEFINITION.....	3
1.2.1 Robotique.....	3
1.2.2 Robot.....	3
1.2.3 Voiture Autonome.....	4
1.3 VOITURE RADIOCOMMANDEE.....	4
1.4 RECHERCHE ET DEVELOPPEMNT DANS LA ROBOTIQUE.....	5
1.4.1 La téléopération.....	5
1.4.2 La cobotique.....	5
1.4.3 La manipulation autonome.....	6
1.5 LES TROIS LOIS DE LA ROBOTIQUE.....	6
1.6 INTERET DE LA ROBOTIQUE.....	7
1.6.1 Historique.....	7
1.6.2 Avantages de la robotisation.....	7
1.7 DOMAINES D'APPLICATION DE LA ROBOTIQUE.....	8
1.8 DEGRES D'AUTONOMIE D'UN VEHICULE.....	8
1.9 L'INTELLIGENCE ARTIFICIELLE.....	13
CHAPITRE 2 L'ENVIRONNEMENT DE DÉVELOPPEMENT.....	14
2.1 INTRODUCTION.....	14
2.2 TYPES DE CARTES ARDUINO.....	15
2.3 APPLICATIONS.....	16

2.4	AVANTAGES.....	17
2.5	PARTIE MATERIELLE (HARDWARE)	18
2.5.1	Caractéristiques techniques de la carte « Arduino MEGA »	18
2.5.2	Moteur driver shield I293d	20
2.5.3	Moteurs à courant continu	21
2.5.4	Réducteur	23
2.5.5	Modulation de la largeur d'impulsion (PWM)	23
2.5.6	Capteurs à Ultrasons	25
2.5.7	GPS	26
2.5.8	Localisation.....	27
2.5.9	Radar.....	28
2.5.10	Lidar.....	29
2.5.11	Fusion de capteurs	30
2.6	PARTIE LOGICIELLE (SOFTWARE)	31
2.6.1	Arduino IDE	31
2.6.2	Proteus	35

CHAPITRE 3 RÉALISATION ET SIMULATION DE LA VOITURE AUTONOME 37

3.1	PRESENTATION	37
3.1.1	Introduction.....	37
3.1.2	GPS.....	39
3.1.3	Batterie lithium ion	40
3.1.4	Servomoteur	41
3.1.5	Module Bluetooth HC-05	42
3.2	SIMULATION.....	43
3.2.1	Simulation du capteur à ultrasons	43

3.2.2	Simulation du GPS	44
3.2.3	Simulation du capteur infrarouge	46
3.2.4	Simulation des moteurs	47
3.3	REALISATION	49
3.3.1	Châssis de la voiture	49
3.3.2	Test du GPS	50
3.3.3	Test du capteur à ultrasons	52
3.3.4	Test du module Bluetooth avec l'application.....	55
3.3.5	Test de la voiture	56
3.4	ORGANIGRAMMES ET SCHEMAS	57
3.4.1	Organigramme général d'évitement d'obstacles.....	57
3.4.2	Lecture de distance pour actionner les moteurs et les LEDs.....	58
CONCLUSION GENERALE		59
ANNEXES.....		60
BIBLIOGRAPHIE		63

Liste des figures

Chapitre 1

Figure 1.1 : Voiture Radiocommandée (RC).....	4
Figure 1.2 : Chaîne de téléopération.....	5
Figure 1.3 : Application de la cobotique.....	5
Figure 1.4 : Manipulation autonome.....	6
Figure 1.5 : Niveaux d'autonomie d'un véhicule.....	12

Chapitre 2

Figure 2.1 : Les différentes cartes Arduino.....	15
Figure 2.2 : Caractéristiques techniques des différents cartes Arduino.....	15
Figure 2.3 : Schéma la carte Arduino MEGA 2560.....	19
Figure 2.4 : Pinout ATmega2560.....	19
Figure 2.5 : Moteur Shield L293d.....	20
Figure 2.6 : Sens du courant dans un pont en H.....	21
Figure 2.7 : Moteur à courant continu en détail.....	22
Figure 2.8 : Moteur à courant continu avec réducteur.....	22
Figure 2.9 : Schéma d'un réducteur.....	23
Figure 2.10 : Exemple de signaux PWM.....	24
Figure 2.11 : Capteur à Ultrasons.....	25
Figure 2.12 : Nuage de points de l'environnement.....	29
Figure 2.13 : Estimation de position avec un Lidar.....	29
Figure 2.14 : Comparatif capteurs.....	30

Figure 2.15 : Logo Arduino IDE.....	31
Figure 2.16 : Interface Arduino IDE.....	32
Figure 2.17 : Interface Arduino IDE expliquée.....	33
Figure 2.18 : Logo Proteus.....	35
Figure 2.19 : Interface Proteus.....	36

Chapitre 3

Figure 3.1 : Composants du circuit.....	38
Figure 3.2 : Schéma Proteus du circuit.....	38
Figure 3.3 : GPS NEO-6M.....	39
Figure 3.4 : Schéma synoptique du GPS NEO-6M.....	40
Figure 3.5 : Batterie lithium ion rechargeable.....	40
Figure 3.6 : Principe de fonctionnement d'un servomoteur.....	41
Figure 3.7 : Module Bluetooth HC-05.....	42
Figure 3.8 : Simulation du capteur à ultrasons.....	44
Figure 3.9 : Simulation du GPS sans Arduino.....	45
Figure 3.10 : Simulation du GPS avec Arduino.....	46
Figure 3.11 : Simulation du capteur infrarouge.....	47
Figure 3.12 : Simulation des moteurs à courant continu sans driver	47
Figure 3.13 : Simulation des moteurs avec le driver L298N.....	48
Figure 3.14 : Châssis de la voiture avant le montage.....	49
Figure 3.15 : GPS avec Arduino.....	50
Figure 3.16 : Résultat du GPS.....	50
Figure 3.17 : Résultat du GPS en cas d'échec de localisation.....	52

Figure 3.18 : Test du capteur à ultrasons.....	52
Figure 3.19 : Présence d'un obstacle.....	53
Figure 3.20 : Sans présence d'un obstacle.....	53
Figure 3.21 : Distance de l'obstacle sur le moniteur série.....	54
Figure 3.22 : Distance de l'obstacle sur matlab.....	54
Figure 3.23 : Icône de l'Application Bluetooth.....	55
Figure 3.24 : Réception des données GPS.....	55
Figure 3.25 : Test de la voiture.....	56
Figure 3.26 : Organigramme d'évitement d'obstacles.....	57
Figure 3.27 : Schéma d'instructions général.....	58
Figure 3.28 : Schéma des entrées/sorties.....	58

Introduction générale

Le monde de l'automobile actuel, connaît une révolution, qui est le passage de l'énergie thermique des véhicules, vers l'énergie électrique, cette transition donnera naissance par la suite aux voitures électriques, celles-ci sont de plus en plus intelligentes, voire même autonomes.

La robotique est la clé de la conception des machines robotisées, ces dernières sont munies de capteurs, prélevant l'information, et d'actionneurs, pouvant réaliser des tâches spécifiques, selon le domaine appliqué, il existe deux types de robotique : mobile et industrielle, ce développement est dû à l'évolution continue dans les différents domaines de la robotique : électronique, informatique, mécanique, ou autre.

Ce projet interprète la conception et la réalisation d'une voiture autonome, qui sera capable de se déplacer toute seule en toute sécurité, selon des niveaux d'autonomie bien définis, cela est rendu possible, grâce aux fonctionnalités intéressantes que propose l'univers de prototypage et de cartes programmables Arduino.

L'utilisation d'Arduino dans ce projet offre plusieurs avantages, notamment sa flexibilité et sa facilité d'utilisation, ainsi que son coût abordable. Avec Arduino, nous pouvons facilement intégrer différents capteurs et actionneurs, ainsi que programmer des algorithmes de contrôle pour réaliser des tâches spécifiques. En outre, la communauté Arduino est très active et fournit un support et des ressources en ligne pour aider les développeurs à résoudre les problèmes et à améliorer leurs projets. Tout au long de ce mémoire, nous allons détailler les étapes de conception et de réalisation de notre voiture autonome, ainsi que les défis rencontrés et les résultats obtenus.

En fin de compte, ce mémoire vise à montrer comment cette carte programmable Arduino peut contribuer à la conception et à la réalisation de systèmes autonomes. En particulier, nous explorons l'application de cette technologie dans l'industrie automobile et comment elle peut être utilisée pour rendre les voitures plus intelligentes et plus sûres. Nous espérons que ce travail sera utile aux ingénieurs, aux développeurs et aux chercheurs qui s'intéressent à la robotique et aux systèmes autonomes, ainsi qu'à toute personne intéressée par les technologies de pointe qui façonnent notre avenir.

Ce mémoire se compose de 3 chapitres,

Le premier chapitre introduit la robotique, sa définition, ses domaines, son intérêt, ainsi que des exemples d'application.

Le deuxième chapitre présente la notion de la carte programmable Arduino, ainsi que ses différents modules hardware et software.

Le troisième chapitre détaille la simulation, les étapes de réalisation du projet, ainsi que sa programmation.

Et pour terminer, on finit avec une conclusion générale, qui résume les résultats.

Chapitre 1 Introduction à la robotique

1.1 Introduction

La robotique combine à elle seule plusieurs sciences, et sa pratique a réuni des savoir-faire techniques et des connaissances scientifiques des domaines de l'électronique, de l'informatique et de la mécanique, voire plus.

Le développement des logiciels de programmation, ainsi que l'intelligence artificielle qui ne cesse de progresser, permettent déjà de nous remplacer sur de nombreuses tâches de plus en plus complexes, comme par exemple la conduite avec les voitures autonomes, ou encore des tâches trop dangereuses, comme l'exploration d'un milieu hostile ou d'un site radioactif [1].

1.2 Définitions

1.2.1 Robotique

La robotique est l'ensemble des études et des techniques autour des robots. La robotique englobe la fabrication, la conception, la réalisation des robots, mais aussi leur utilisation pour une tâche déterminée dans une situation précise ou non [2].

1.2.2 Robot

Le robot est une machine construite, non biologique, qui reproduit dans une certaine mesure, les gestes, les tâches et parfois le comportement d'un humain ou d'un animal.

1.2.3 Voiture autonome

Ce que l'on appelle une voiture autonome est un véhicule capable de rouler sans l'intervention d'un être humain, grâce à de nombreux capteurs et à un logiciel spécifique, elle est capable de se déplacer dans le trafic et de prendre des décisions toute seule, sans l'apport d'un conducteur.

Les premiers pas de la voiture autonome ont été réalisés avec le stationnement semi-automatique il y a quelques années, c'est le même principe mais appliqué à la conduite toute entière, de nombreux constructeurs (Tesla, Google et d'autres) travaillent aujourd'hui sur le développement de leur propre système autonome qui est perçu comme l'avenir de l'automobile [3].

1.3 Voiture Radiocommandée

Ce sont des voitures Radiocommandées (RC), qu'on peut contrôler à distance (par radio, infrarouge ou par connexion câblée), grâce à une télécommande ou un autre dispositif, ces dernières sont des voitures ayant une petite taille (miniature), qui sont des modèles réduits de véhicules à grande taille.

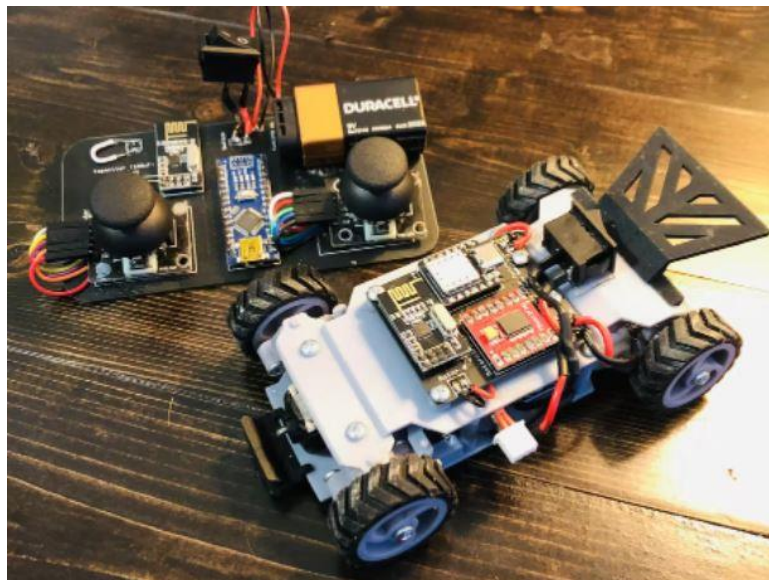


Figure 1.1. Exemple de voiture Radiocommandée (RC).

1.4 Recherche et développement dans la robotique

1.4.1 La téléopération

Aussi appelée « opération à distance », l'activité dans le domaine de la téléopération est orientée vers le contrôle supervisé et l'assistance à l'opérateur pour augmenter l'efficacité des tâches opérées à distance.

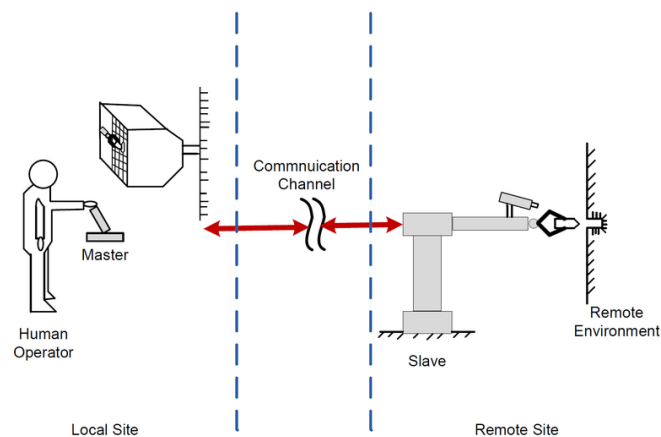


Figure 1.2. Chaîne de téléopération.

1.4.2 La cobotique

La Cobotique, ou robotique collaborative, vise à développer des technologies robotiques en interaction continue avec l'homme, en coopération. La recherche porte sur la sécurité et l'efficacité de l'interaction « homme-robot ».

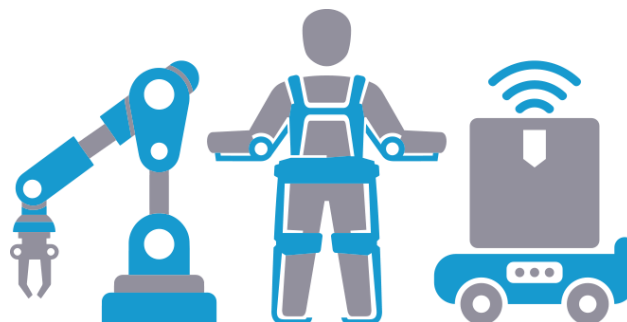


Figure 1.3. Application de la cobotique.

1.4.3 La manipulation autonome

L'objectif de cette discipline est de développer les technologies pour la robotique mobile et la manipulation intelligente, par exemple des objets, pour le transport ou déplacement d'objets dans le milieu industriel [4].

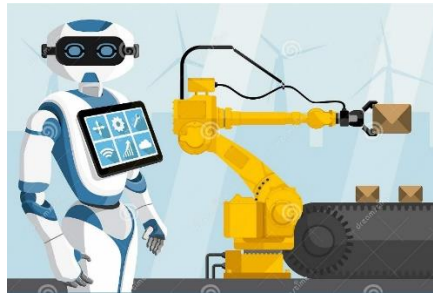


Figure 1.4. Manipulation autonome.

1.5 Les trois lois de la robotique

Pour limiter les comportements ou effets néfastes que pourraient provoquer des robots pourvus d'autonomie, trois lois de la robotique ont été édictées :

- Un robot ne peut porter atteinte à un être humain ni, restant passif, permettre qu'un être humain soit exposé au danger.
- Un robot doit obéir aux ordres que lui donne un être humain, à part si de tels ordres peuvent entrer en conflit avec la première loi.
- Un robot doit protéger son existence tant que cette protection n'entre pas en conflit avec la première ou la deuxième loi [5].

1.6 Intérêt de la robotique

1.6.1 Historique

Pendant plus de deux siècles, l'usage des robots a été presque exclusivement réservé aux tâches répétitives ou dangereuses dans les industries, avec quelques exceptions dans des secteurs peu accessibles au grand public comme le domaine militaire et l'aérospatial.

De nos jours, la robotique est omniprésente, pour usage domestique ou professionnel, public ou privé, les robots sont à peu près partout.

1.6.2 Avantages de la robotisation

- **Plus de productions avec moins de coûts**

Programmé par l'opérateur, le robot peut ensuite fonctionner seul, 24h/24 et 7 jours/7, offrant plus de flexibilité dans la gestion de production et vous permettant d'augmenter la production.

- **Plus de flexibilité**

Au contraire des machines spécialisées pour une seule et unique tâche, un robot est flexible, il s'adapte à différentes opérations, il suffit de le reprogrammer, voire de changer ses équipements périphériques, pour qu'il s'adapte aux évolutions du produit.

- **Meilleure qualité des produits**

Le robot répète une opération toujours de la même manière, avec le même rendement, même performance et une qualité constante, assurant plus de fiabilité, ainsi la qualité des produits s'améliore, et le processus est stable et maîtrisé.

- **Améliorer les conditions de travail**

Le robot aide les opérateurs sur les postes de travail présentant un risque pour leur santé et générateur de maladies professionnelles et TMS, il automatise les tâches pénibles, dangereuses ou répétitives : manutention de charges lourdes, environnement bruyant, émanations, risque de brûlures [6].

1.7 Domaines d'application de la robotique

-En médecine dans les hôpitaux et autres établissements de santé pour réaliser des examens, des traitements (radiothérapie) ou des interventions chirurgicales dans des zones difficiles d'accès qui requièrent une grande précision (ablations).

-Dans le domaine spatial pour explorer l'espace, la lune ou des planètes.

-Dans les logements des particuliers comme les tondeuses et les aspirateurs.

-Dans la construction automobile, surtout les véhicules à moteur électrique.

-Dans le transport pour le déplacement des personnes ou des marchandises [7].

1.8 Degrés d'autonomie d'un véhicule

Il existe au total 6 niveaux d'autonomie d'un véhicule, allant du plus simple et basique, au plus avancé et complexe :

- **Niveau 0**

A ce niveau, il n'existe aucune autonomisation, tout se fait manuellement, cependant on peut rajouter un petit plus pour améliorer certaines fonctions basiques, rendant la navigation plus simple.

Exemples :

-Radar de recul : c'est un système de sécurité efficace qui permet aux conducteurs de reculer en toute confiance et de réduire les risques de collisions.

En utilisant des ondes sonores, le radar mesure la distance entre le véhicule et les objets, fournissant des informations précieuses via un écran ou des signaux sonores. Il est devenu un équipement standard sur de nombreuses voitures modernes, offrant une sécurité accrue et évitant les dommages à la carrosserie.

- **Niveau 1**

Le système est capable de prendre en charge soit le contrôle longitudinal (vitesse et distance avec les véhicules qui précèdent), soit le contrôle latéral du véhicule (suivi des lignes blanches) mais pas les deux en même temps.

Exemples :

-Régulateur de vitesse : c'est un système destiné à stabiliser automatiquement la vitesse des véhicules automobiles. Le conducteur fixe la vitesse à laquelle il veut rouler puis l'automatisme prend le relais et maintient la vitesse définie.

-Freinage automatique d'urgence : c'est un dispositif qui permet à la voiture ou au camion de freiner automatiquement lorsqu'il détecte une collision imminente avec un véhicule, un piéton ou un autre obstacle détecté.

-Avertisseur de collision : est un système de sécurité active conçu pour réduire la gravité des accidents des véhicules motorisés.

-Régulateur adaptatif : est une version évoluée du régulateur de vitesse. Il ne permet pas uniquement de maintenir une vitesse de croisière : il est également capable de calculer et de conserver une distance de sécurité avec le véhicule qui le précède sur la même voie.

- **Niveau 2**

Dans certaines situations, le conducteur peut déléguer au système le contrôle longitudinal et latéral du véhicule, mais il reste responsable de la supervision : il doit garder un œil sur le système à tout moment, surveiller l'environnement du véhicule et reprendre le contrôle total si nécessaire.

Exemples :

-Assistant de conduite basse vitesse dans les embouteillages : La fonction Assistant dans les embouteillages peut maintenir la voiture immobile pendant 4 minutes au maximum. Le frein de stationnement est ensuite activé et le régulateur adaptatif de vitesse est alors désactivé.

-Assistant de parking et stationnement : L'aide au stationnement désigne tout dispositif à même d'assister le conducteur lorsqu'il veut stationner son véhicule.

- **Niveau 3**

La surveillance de l'environnement du véhicule incombe au système du véhicule, cela implique des fonctions avancées de détection d'objets (routes, lignes, véhicules) et d'événements, la conduite est autonome mais seulement dans des conditions prédéfinies, sur autoroute, par exemple.

Exemple :

-Le pilote d'autoroute : c'est un système de conduite semi-autonome qui permet au conducteur de se reposer pendant les longs trajets sur autoroute. Le système utilise des capteurs pour maintenir la voiture dans sa voie, régler la vitesse et maintenir une distance de sécurité avec les autres véhicules.

-Le parking automatique est un système de stationnement assisté qui permet aux conducteurs de garer leur voiture sans avoir à manœuvrer. Le système utilise des capteurs pour détecter les places de stationnement et la voiture se gare automatiquement. Cela permet de réduire le stress et le temps passé à chercher une place de stationnement [8].

- **Niveau 4**

L'autonomie totale se limite à une zone géographique spécifique, par exemple une autoroute ou un parking avec lequel le véhicule est compatible. Elle se fait également dans des conditions météorologiques et de visibilité précises.

Hors de cette zone, la conduite redevient manuelle, mais en cas d'absence d'instruction, le véhicule doit être en mesure de réagir tout seul.

Exemple :

-Valet Parking : pour aller se garer ou venir chercher le chauffeur, Le valet parking avec véhicule autonome est un système de stationnement qui utilise des voitures autonomes pour récupérer les voitures des clients et les garer automatiquement.

- **Niveau 5**

Les systèmes de niveau 5 n'ont jamais besoin de l'aide du conducteur, car la voiture conduit réellement toute seule dans toutes les conditions et est capable de réagir comme le ferait un conducteur humain (voire mieux). Les éléments de contrôle comme le volant ou les pédales ne sont plus nécessaires.

Exemple : La Google Car de niveau 5 autonomie est un véhicule autonome entièrement automatisé qui n'a pas besoin de conducteur humain. Il s'agit d'un véhicule électrique qui utilise un ensemble de capteurs, de caméras et de logiciels sophistiqués pour détecter et interpréter l'environnement routier en temps réel.

Il est capable de prendre des décisions de conduite de manière autonome, comme changer de voie, s'arrêter aux feux de signalisation, éviter les obstacles et les autres véhicules, et naviguer dans des conditions de conduite difficiles, comme la pluie et la neige. Le véhicule est également équipé d'un système de communication qui lui permet de communiquer avec les autres véhicules et les infrastructures routières [9].



















	Accélération freinage & volant	Surveillance de la route	Contrôle en cas de problème	Quelles conditions
Niveau 0				Toutes les routes
Niveau 1				Certaines routes
Niveau 2				Certaines routes
Niveau 3				Certaines routes
Niveau 4				Certaines routes
Niveau 5				Toutes les routes

Figure 1.5. Niveaux d'autonomie d'un véhicule

1.9 L'intelligence artificielle

L'intelligence artificielle est une branche de l'informatique qui permet aux systèmes d'apprendre et d'exécuter des tâches normalement associées à l'intelligence humaine, telles que la reconnaissance vocale, la prise de décisions ou la perception visuelle, l'intelligence artificielle est partout à bord des voitures autonomes, et en particulier dans la perception de l'environnement [10].

L'application de l'intelligence artificielle dans un véhicule autonome a révolutionné l'industrie automobile. Grâce à des systèmes avancés de perception, de prise de décision et de contrôle, ces véhicules peuvent désormais naviguer en toute sécurité sans l'intervention d'un conducteur humain. L'IA analyse en temps réel les données provenant des capteurs embarqués tels que les caméras, les lidars et les radars, pour détecter les obstacles, les piétons, les panneaux de signalisation et les autres véhicules sur la route. Elle utilise ensuite ces informations pour prendre des décisions intelligentes en matière de conduite, telles que l'accélération, le freinage, le changement de voie et même la gestion des situations d'urgence.

L'IA dans les véhicules autonomes promet d'améliorer la sécurité routière, de réduire les accidents et de faciliter la mobilité en offrant une expérience de conduite plus confortable et efficace [11].

Chapitre 2 L'environnement de développement

2.1 Introduction

Arduino est une plate-forme électronique open source basée sur du matériel et des logiciels faciles à utiliser, il est destiné à tous ceux qui réalisent des projets interactifs, cela permet d'innover et de concevoir différents projets électroniques, pour tout type d'usage.

Cette plateforme est considérée comme libre de droit, car les schémas de ces cartes sont sous une licence open-source (libre), par contre le microcontrôleur, lui ne l'est pas.

Il existe différentes cartes électroniques produites par Arduino pour différents usages, on peut citer : Arduino Nano / UNO Rev3 / Mega / Due, etc.

La plus utilisée reste l'Arduino UNO Rev3 grâce à son coût relativement bas et ses possibilités offertes pour réaliser différents projets divers et variés.

Le langage utilisé pour programmer le microcontrôleur est le C/C++, Arduino a créé sa propre bibliothèque de fonctions pour faciliter la programmation des microcontrôleurs pour les débutants et les non-spécialistes en électronique. Cette bibliothèque de fonctions est conçue pour masquer certains détails techniques et simplifier la programmation de la carte Arduino, le programme peut être ensuite téléversé à travers les ports USB d'un ordinateur en utilisant le logiciel dédié pour qui s'appelle « Arduino IDE » : c'est une application créée avec le langage JAVA, la mise en place de ce langage standard rend aisé le développement des programmes sur les plates-formes Arduino à toute personne maîtrisant le C/C++.

2.2 Types de cartes Arduino

Il existe différentes cartes Arduino, disponibles sur le marché, allant de la plus petite, à la plus grande, avec plus d'entrées/sorties, plus de mémoire et un meilleur microcontrôleur, ou d'autres améliorations.

Plusieurs cartes, pour différentes tailles et caractéristiques, comme on peut le voir sur ces images :



Figure 2.1. Les différentes cartes Arduino [12].

Boards	Microcontroller	Operating Voltage/s (V)	Digital I/O Pins	PWM Enabled Pins	Analog I/O Pins	DC per I/O (mA)	Flash Memory (KB)	SRAM (KB)	EEPROM (KB)	Clock (MHz)	Length (mm)	Width (mm)	Cable	Native Network Support
Uno	ATmega328	5	14	6	6	20	32	2	1	16	68.6	53.4	USB A-B	None
Leonardo	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	micro-USB	None
Micro	ATmega32u4	5	20	7	12	40	32	2.5	1	16	48	18	micro-USB	None
Nano	ATmega328	5	22	6	8	40	32	2	0.51	16	45	18	mini-B USB	None
Mini	ATmega328	5	14		6	20	32	2	1	16	30	18	USB-Serial	None
Due	Atmel SAM3X8E ARM Cortex-M3 CPU	3.3	54	12	12	800	512	96	X	84	102	53.3	micro-USB	None
Mega	ATmega2560	5	54	15	16	20	256	8	4	16	102	53.3	USB A-B	None
M0	Atmel SAMD21	3.3	20	12	6	7	256	32	X	48	68.6	53.3	micro-USB	None
Yun Mini	ATmega32u4	3.3	20	7	12	40	32	2.5	1	400	71.1	23	micro-USB	Ethernet/Wifi
Uno Ethernet	ATmega328p	5	20	4	6	20	32	2	1	16	68.6	53.4	Ethernet	Ethernet
Tian	Atmel SAMD21	5	20	12	0	7	16000	64000	X	560	68.5	53	micro-USB	Ethernet/Wifi
Mega ADK	ATmega2560	5	54	15	16	40	256	8	4	16	102	53.3	USB A-B	None
M0 Pro	Atmel SAMD21	3.3	20	12	6	7	256	32	X	48	68.6	53.3	micro-USB	None
Industrial 101	ATmega32u4	5	7	2	4	40	16000	64000	1	400	51	42	micro-USB	Ethernet/Wifi
Uno Wifi	ATmega328	5	20	6	6	20	32	2	1	16	68.6	53.4	USB A-B	Wifi
Leonardo Ethernet	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	USB A-B	Ethernet
MKRI000	Atmel SAMD21	3.3	8	12	7	7	256	32	X	48	64.6	25	micro-USB	Wifi

Figure 2.2. Caractéristiques techniques des différentes cartes Arduino.

2.3 Applications

Les cartes Arduino sont capables de lire les entrées : par exemple la lumière sur un capteur (photorésistance LDR), ou bien un doigt sur un bouton poussoir, ou un message sur une boîte mail, et de les transformer en une sortie : en activant un moteur, en allumant une LED, ou en publiant quelque chose en ligne par exemple.

On peut programmer la carte pour réaliser différentes fonctions, cela se fait en envoyant un ensemble d'instructions au microcontrôleur sur la carte, pour le faire, on utilise le matériel (hardware) Arduino (basé sur le câblage) et le logiciel (software) Arduino (IDE), basé sur le traitement.

Les applications de Arduino sont très diversifiées et peuvent toucher plusieurs domaines tels que l'automatique, la robotique, la mécanique, la domotique et ainsi, cela étant possible grâce aux différents modules qui accompagnent Arduino, on parle surtout de capteurs, circuits intégrés, et différents modules destinés à presque chaque usage que l'on souhaite.

Arduino peut aussi être utilisé pour développer des applications matérielles industrielles légères, et peut aussi recevoir en entrées une très grande variété de capteurs, ou d'actionneurs (lumières, moteurs ou toutes autres sorties matérielles).

Les projets Arduino peuvent être autonomes avec une alimentation sur batterie par exemple, ou bien communiquer avec des logiciels sur un ordinateur, ou bien une communication à distance (Wi-Fi ou Bluetooth) pour les projets IoT.

Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées préassemblées, le logiciel de développement open-source est téléchargeable gratuitement.

2.4 Avantages

L'intérêt principal des cartes Arduino est leur facilité de mise en œuvre, vu que Arduino fournit un environnement de développement qui s'appuie sur des outils open source disponibles pour tout le monde.

Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités des relations humain/machine ou environnement/machine, grâce à la popularité qu'il a gagné au fil des années, la communauté importante d'utilisateurs et de concepteurs ne cesse de grandir, ce qui permet à chacun de trouver les réponses à ses questions pour s'entraider à innover.

Ces cartes peuvent être connectées notamment à des capteurs sonores, thermiques, ou de mouvement, qui généreront ensuite des signaux électriques qui seront directement traités par le microcontrôleur, qui peut avec ces informations générer des images, actionner un bras articulé, envoyer des messages sur Internet, etc.

Parmi toutes ces fonctionnalités, on peut citer les avantages suivants :

- **Peu coûteux** : Les cartes Arduino sont relativement peu coûteuses par rapport aux autres plates-formes de microcontrôleurs, la version la moins chère du module Arduino peut être assemblée à la main pour diminuer le coût, et même les modules Arduino pré assemblés coûtent peu cher.
- **Multiplateforme** : Le logiciel Arduino (IDE) fonctionne sur la plupart des systèmes d'exploitation pour ordinateur : Windows, Macintosh OSX et Linux, contrairement à la plupart des autres systèmes de microcontrôleurs, qui sont souvent limités à Windows uniquement.
- **Environnement de programmation simple et clair** : Le logiciel Arduino (IDE) est facile à utiliser pour les débutants, mais suffisamment flexible pour des projets plus complexes, cependant, pour les enseignants, il est basé sur l'environnement de programmation de traitement, de sorte que les étudiants qui apprennent à programmer dans cet environnement seront familiarisés avec le fonctionnement de l'IDE d'Arduino.

- Logiciel open source et extensible : Le logiciel Arduino est publié sous forme d'outils open source, disponibles par extension pour les programmeurs expérimentés qui souhaitent approfondir leurs connaissances, car le langage peut être étendu via les bibliothèques C++, et ces personnes peuvent passer d'Arduino au langage de programmation AVR C sur lequel il est basé, on peut aussi ajouter du code AVR-C directement dans les programmes Arduino [13].

2.5 Partie Matérielle (Hardware)

2.5.1 Caractéristiques techniques de la carte « Arduino MEGA 2560 R3»

Sortie en 2010, la carte Arduino Mega 2560 est une carte électronique programmable a été conçue comme la remplaçante de la carte Arduino Mega, Elle est dotée d'un microcontrôleur ATmega2560 disposant d'une mémoire flash de 256 KB, dont 8 KB dévolus au Bootloader, basée sur un ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs.

Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la Uno, est constituée de 54 broches d'entrées/sorties, dont 15 sont utilisables en PWM, de 16 broches d'entrées analogiques, de 4 ports série hardware, d'une connectique USB, d'une connectique d'alimentation, d'un port ICSP et d'un bouton RESET.

L'Arduino Mega 2560 peut être alimentée à travers le port USB lorsqu'elle est branchée sur l'ordinateur, ou via la connectique d'alimentation avec le port jack ou l'entrée d'alimentation, la source d'alimentation fournissant le meilleur voltage sera sélectionnée comme source d'alimentation par la carte, comme pour l'Arduino Uno, il est conseillé d'utiliser une alimentation entre 7V et 12V de courant continu pour alimenter [14].

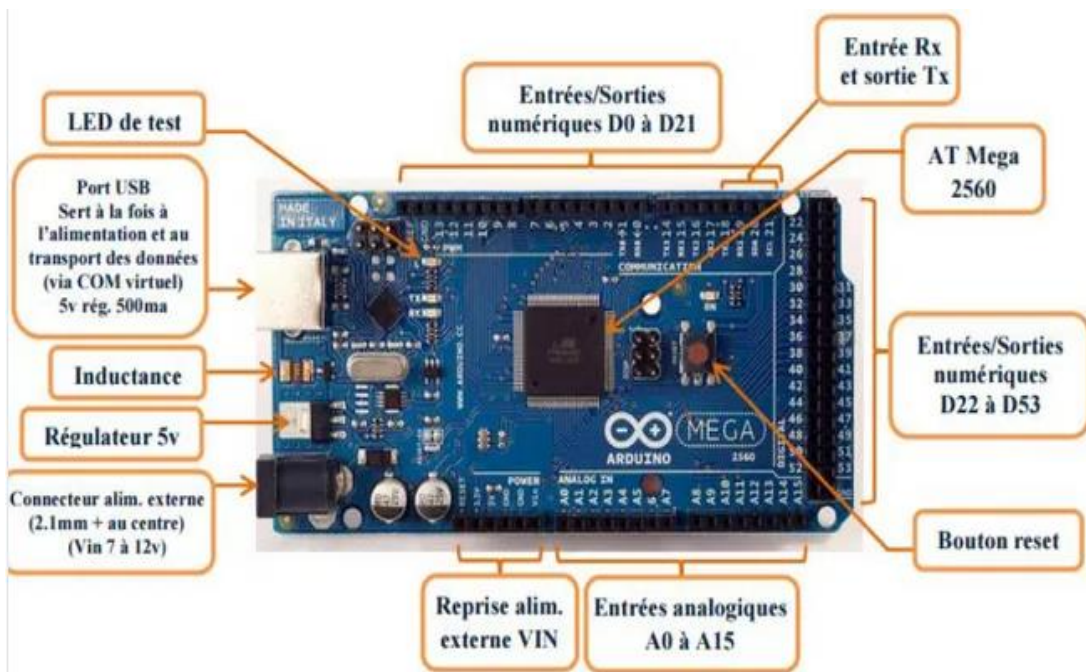


Figure 2.3. Schéma de la carte Arduino MEGA 2560.

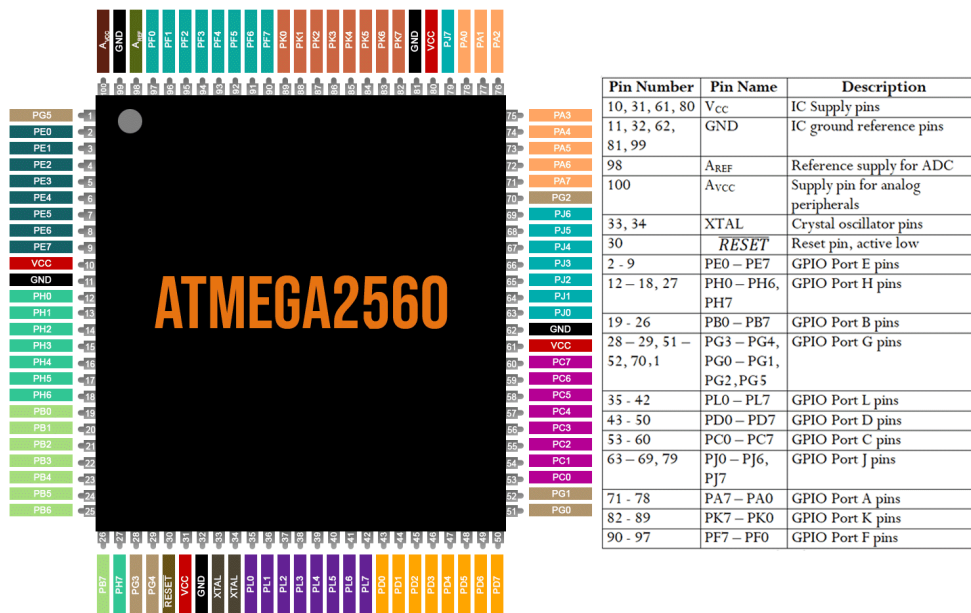


Figure 2.4. Pinout de l'ATMega2560.

2.5.2 Carte de puissance (Moteur Shield Driver L293d)

Notre robot autonome est véhiculé par deux moteurs à courant continu alimentés par une batterie et commandés chacun par un pont en H.

Le L293d se compose de 4 ponts en H : qui fournissent 0,6 A par pont (1,2 A en crête), ce qui permet de contrôler les moteurs avec différentes vitesses, et peut supporter jusqu'à 4 moteurs à courant continu, ou bien 2 moteurs pas à pas et 2 servos moteurs.

Tout cela est intégré logiciellement grâce à la bibliothèque « Adafruit Motor Shield Library », ce qui facilite grandement l'usage des moteurs avec Arduino.

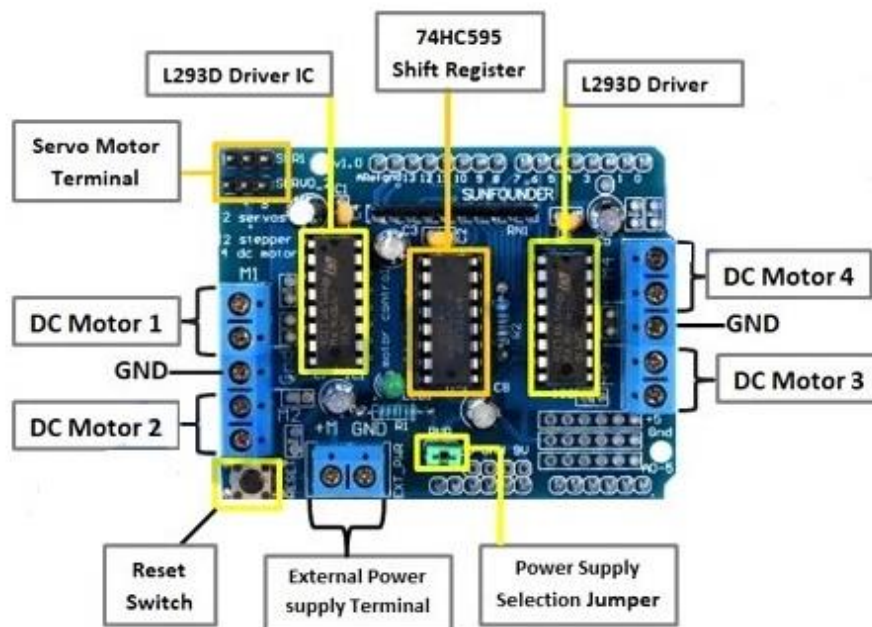


Figure 2.5. Moteur Shield l293d.

Principe de fonctionnement d'un Pont en H

Dans son principe de base, le pont en H est un assemblage de 4 transistors (T1 à T4), montés de telle façon que le courant puisse circuler dans les deux sens au travers du moteur, ce qui permettra de changer son sens de rotation en changeant la polarité à ses bornes, mais il peut aussi grâce la PWM, faire varier la vitesse du moteur en modulant la tension à ses bornes.

La structure du pont permet de modifier le sens du courant dans le moteur en gérant correctement l'ouverture et la fermeture des commutateurs :

Si T1 et T4 sont fermés alors que T2 et T3 sont ouverts, le courant passe de A vers B

Si T2 et T3 sont fermés alors que T1 et T4 sont ouverts, le courant passe de B vers A

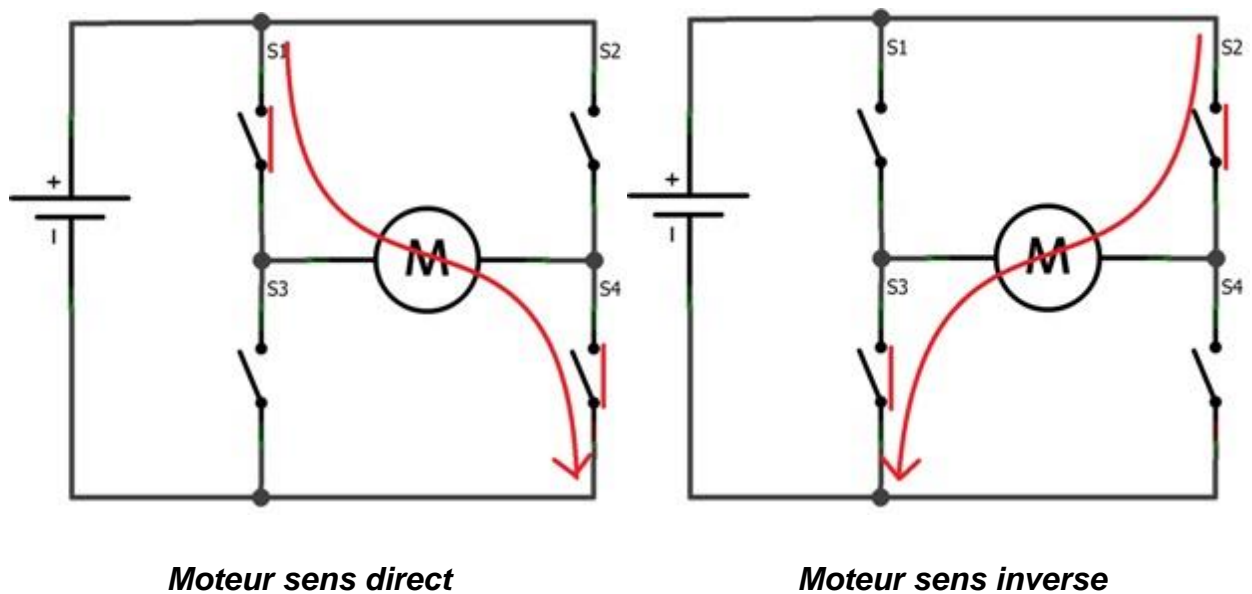


Figure 2.6. Sens du courant dans un pont en H.

2.5.3 Moteurs à courant continu

Les moteurs électriques permettent de convertir l'énergie électrique en mouvement mécanique, ils sont un des éléments majeurs de la robotique mobile puisqu'ils permettent des mouvements ou des déplacements, c'est pour cela qu'on les retrouve dans tous les robots, il existe plusieurs types de moteurs : moteurs à courant continu, servomoteurs, moteurs pas à pas et moteurs brushless, etc.

Ils sont constitués de deux parties électriques : le stator et le rotor, lorsqu'on alimente le moteur, il se crée une interaction magnétique qui met le moteur en mouvement, inverser le sens de la tension qui alimente le moteur, inversera son sens de rotation.

Les balais, aussi appelés charbons montés, « brush » en Anglais, d'où le nom moteur brushless, c'est-à-dire sans balais, sont des contacts mécaniques glissants, qui transmettent le courant électrique entre la partie tournante d'une machine et son circuit extérieur fixe, et dans le cas de machines à courant continu, assurent une commutation sans étincelle.

Le collecteur est un système de frottement, qui permet d'alimenter le rotor avec ces balais qui frottent sur les contacts en rotation.

On utilise en général des moteurs à excitation constante, l'inducteur est formé d'aimants permanents, seul le rotor est alimenté par 2 fils.

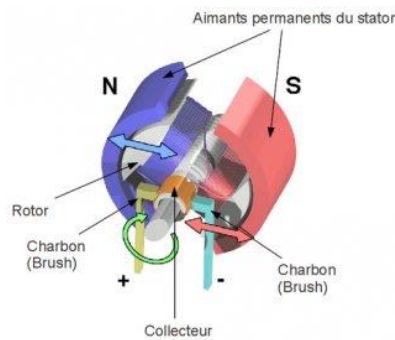


Figure 2.7. Moteur à courant continu en détail.

Le moteur à courant continu est très simple d'utilisation, pour le faire fonctionner, il suffit d'appliquer une tension électrique à ses bornes, le signe et le niveau de cette tension vont imposer le sens et la vitesse de rotation, mais pour piloter la tension à ses bornes, il est nécessaire d'utiliser un relais ou un pont en H [15].



Figure 2.8. Moteur à courant continu avec réducteur.

2.5.4 Réducteur

Le réducteur, aussi appelé moto-réducteur, qui est la combinaison entre un moteur électrique, et un réducteur, permet d'augmenter le couple d'un moteur, et baisse le nombre de tours, pour diminuer la vitesse de rotation.

Les moteurs à courant continu possèdent souvent une boîte de réduction afin d'augmenter leur couple pour un encombrement réduit.

Comme on peut le voir sur cette figure, le réducteur peut être représenté comme un système, avec des entrées et des sorties :

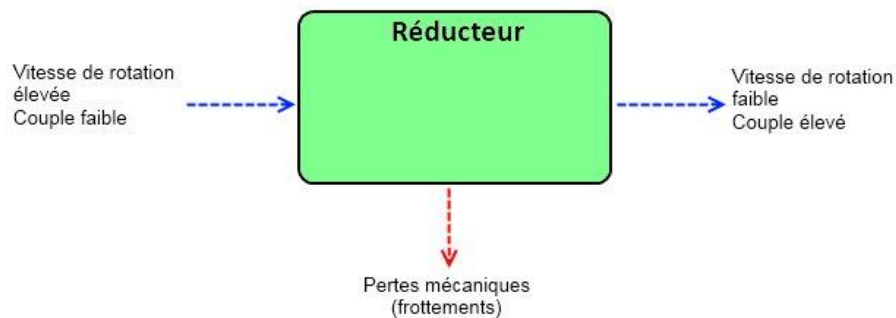


Figure 2.9. Schéma d'un Réducteur.

2.5.5 Modulation de largeur d'impulsion (PWM)

Il s'agit d'une technique de création de signaux qui permet de contrôler de l'analogique avec des sorties numériques d'un microcontrôleur, la commande en PWM consiste en une succession rapide de signaux numériques.

Ces signaux peuvent être par exemple des signaux tout ou rien, c'est-à-dire alternant sans transition entre un état bas « 0 » et un état haut « 1 ».

Si la fréquence choisie est suffisamment rapide, le résultat de cette rapide alternance d'allumage/extinction se fond dans une valeur moyenne.

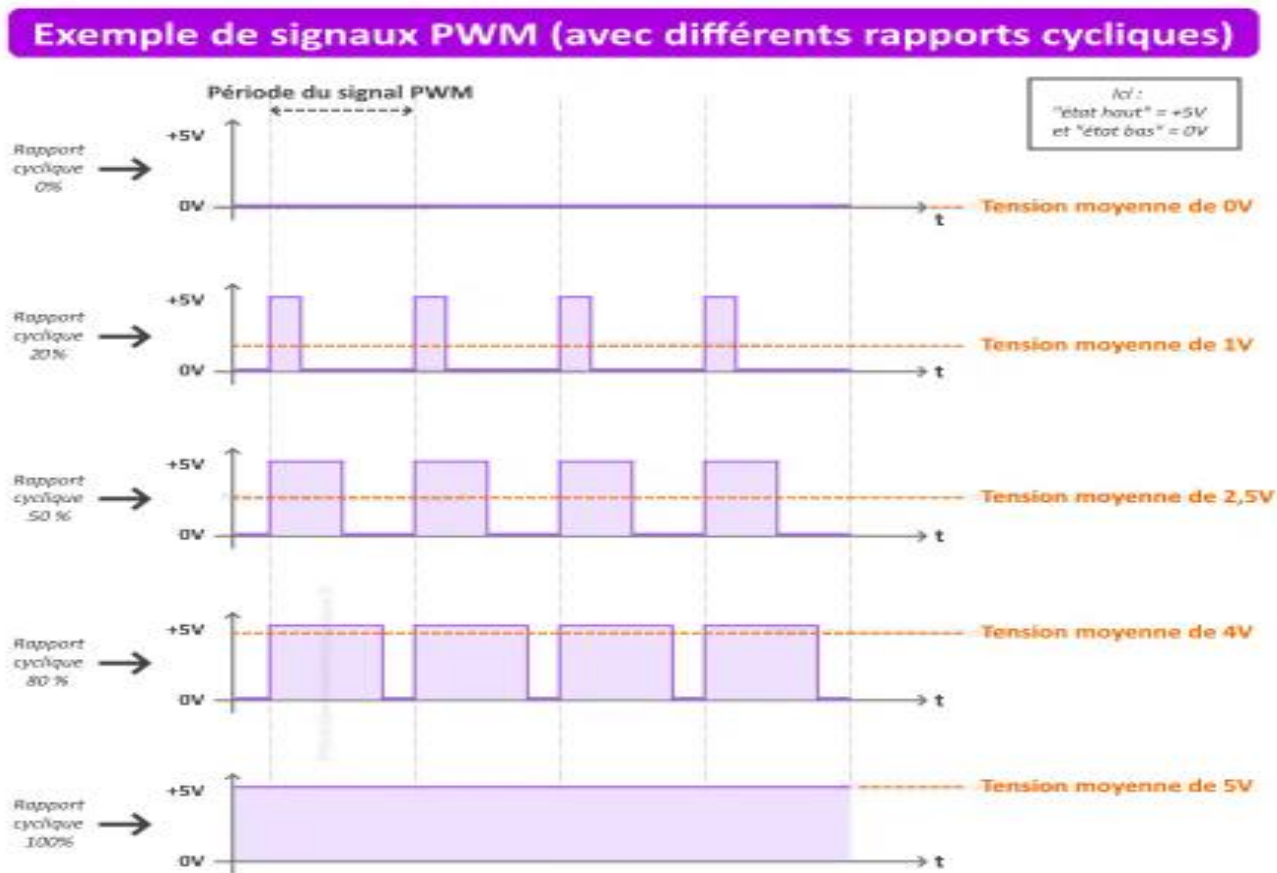


Figure 2.10. Exemple de signaux PWM.

Voici quelques exemples d'applications de la PWM :

- Gradation d'une LED.
- Fournir une sortie analogique ; si la sortie numérique est filtrée, elle fournira une tension analogique comprise entre 0% et 100%.
- Génération de signaux audio.
- Fournir un contrôle de vitesse variable pour les moteurs.
- Génération d'un signal modulé, par exemple pour piloter une LED infrarouge pour une télécommande [16].

2.5.6 Capteurs à ultrasons

Un capteur à ultrasons est un dispositif de contrôle industriel ou grand public, qui est utilisé pour mesurer et calculer la distance du capteur à un objet cible spécifié, ce dernier utilise les ondes sonores supérieures à 20 000 Hz, qui sont trop élevées pour être captées par l'oreille humaine qui entend uniquement de les sons à fréquence 20 Hz jusqu'à 20 KHz.

Le capteur possède un transducteur en céramique qui vibre quand on lui applique une énergie électrique, les vibrations compriment et dilatent les molécules d'air en vagues entre le capteur et la cible qui est en face, ce transducteur permet de transmettre et de recevoir des ondes sonores.

Le capteur à ultrasons mesure la distance en émettant une onde sonore et en écoutant pendant une durée définie pour permettre le retour de l'onde réfléchiée par la cible avant d'en émettre une nouvelle.

Un capteur à ultrasons émet à intervalles réguliers des courtes impulsions sonores à haute fréquence : ces impulsions se propagent dans l'air à la vitesse du son, et lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur : celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho [17].

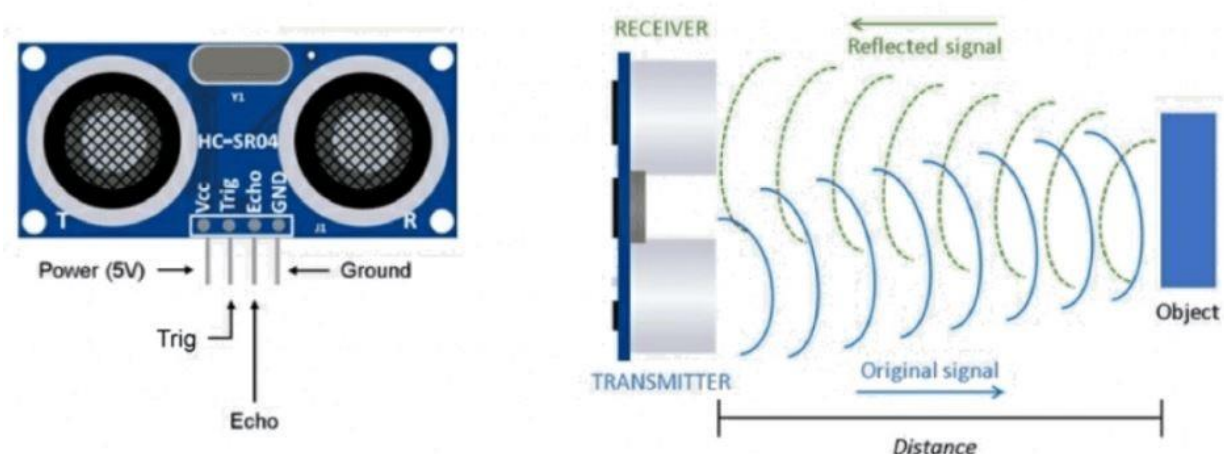


Figure 2.11. Capteur à Ultrasons.

Un parasite acoustique émis à une fréquence reconnue par le capteur à ultrasons peut interférer avec la sortie du capteur, ces interférences acoustiques peuvent également se produire si on installe deux capteurs à ultrasons de la même fréquence à proximité l'un de l'autre, cependant les parasites électriques n'affectent pas les capteurs à ultrasons.

Les fluctuations de température affectent la vitesse des ondes sonores d'un capteur à ultrasons : plus la température augmente, plus les ondes sonores se propagent rapidement entre le capteur et la cible, ceci même si la cible n'a pas bougé, elle apparaît plus proche pour le capteur, il est dès lors possible que le capteur ne puisse pas détecter l'emplacement exact de la cible.

Les meilleures cibles pour un capteur à ultrasons sont des surface planes, solides et de grande taille dans des matériaux comme le métal, la céramique, le verre ou le bois, elles doivent toujours être perpendiculaires au capteur, et non pas des cibles présentant une surface irrégulière ou molle.

Le capteur doit apprendre « l'arrière-plan » comme condition correcte : lorsque la condition apprise, et la surface en arrière-plan réfléchit les ultrasons, tout objet introduit entre le capteur et l'arrière-plan sera détecté, entraînant la commutation de la sortie [18].

2.5.7 **GPS (Global Positioning System)**

Le GPS, ou Global Positioning System, est un dispositif embarqué de navigation automobile, il utilise généralement un appareil de navigation par satellite pour obtenir ses données de position, qui sont ensuite corrélées à une position sur la route, afin de fournir la direction programmée à suivre [19].

GPS est originalement connu sous le nom de NAVSTAR GPS, c'est le système américain pour le positionnement, en Europe, on parle de Galileo, mais en Russie, on parle de GLONASS, il existe aussi le terme GNSS (Global Navigation Satellite System), qui est un système de positionnement par satellite, qui peut utiliser plusieurs de ces sous-systèmes pour augmenter la précision.

Les signaux reçus d'une antenne GPS peuvent être combinés avec des données d'autres capteurs (par exemple, des capteurs à ultrasons et/ou des caméras), afin de planifier un itinéraire et pour amener un véhicule à un emplacement cible, mais il ne faut pas oublier que les GPS sont peu efficaces en navigation intérieure à cause des interférences que peut subir le signal dans un endroit fermé.

Dans une voiture autonome, les GPS permettent par principe de situer la position du véhicule, mais dans ces mesures, il peut y avoir une erreur allant de 1 à 10 mètres, cette erreur est trop importante et peut potentiellement être fatale pour les passagers ou l'environnement du véhicule autonome, on ajoute donc une étape appelée localisation, qui est l'implémentation d'algorithmes pour estimer où est le véhicule avec une erreur de moins de 10 cm [20].

2.5.8 Localisation

La localisation est une étape implémentée dans la majorité des robots et véhicules, qui est faite pour se localiser avec une marge d'erreur faible, mais aussi car pour prendre des décisions comme dépasser un véhicule ou simplement définir un itinéraire, il faut savoir ce qu'il y a autour du véhicule en utilisant la fonction fusion de capteurs et savoir la position, ce n'est qu'avec ces informations qu'on peut définir une trajectoire.

Aujourd'hui, les recherches explorent la localisation à l'aide d'algorithmes basés sur le Deep Learning, en utilisant une caméra, qui remplace la vision du conducteur, elle peut être utilisée pour comprendre l'environnement avec de l'intelligence artificielle, cela en classifiant les routes, les piétons, les panneaux ou autre.

Si la localisation d'un véhicule est assez précise, il peut se déplacer de façon autonome, ce sujet évolue en permanence, donc les capteurs deviennent de plus en plus précis et les algorithmes de plus en plus performants, c'est ce qu'on va détailler dans les lignes qui suivent [20].

2.5.9 Le RADAR (Radio Detection and Ranging)

Les radars émettent des ondes radio pour détecter les objets dans un rayon de plusieurs mètres, ils sont présents dans nos voitures depuis des années pour détecter les véhicules situés dans les angles morts ou pour éviter les collisions, car ils ont de meilleurs résultats sur les objets en mouvement que sur les objets statiques.

Contrairement aux autres capteurs qui calculent la différence de position entre deux données, le radar utilise *l'effet Doppler* : qui correspond au phénomène physique selon lequel la fréquence d'une onde semble modifiée lorsque la source émettrice de l'onde et/ou l'observateur de l'onde sont en mouvement, cela se fait en mesurant le changement dans la fréquence d'onde suivante pour savoir si le véhicule avance vers nous ou s'éloigne [21].

Cette mesure est appelée *vélocité radiale*, qui est la vitesse d'un objet mesurée dans la direction du rayon (ou la ligne de visée) vers ou depuis le point d'observation [22].

Le radar peut estimer directement une vitesse, il a une faible résolution, mais il permet de connaître la position et la vitesse d'un objet détecté, en revanche, il ne peut pas déterminer de quel objet s'agit-il, il existe aussi les capteurs odométriques qui permettent eux aussi d'estimer la vitesse d'un véhicule, en étudiant le déplacement de ses roues pour reconstituer le mouvement global du robot et en partant d'une position initiale connue, et en intégrant les déplacements mesurés, on peut ainsi calculer à chaque instant la position courante du véhicule [20].

2.5.10 Le LIDAR (Light Detection and Ranging)

Le lidar utilise des capteurs infrarouges pour déterminer la distance avec un objet, un système rotatif lui permet d'envoyer des ondes et de mesurer le temps mis pour que cette onde lui revienne, cela lui permet de générer un nuage de points de l'environnement autour du capteur.

En général, un lidar peut générer environ 2 millions de points par seconde, ce nuage de points donnant des formes 3D différentes, permet de faire de la classification d'objets comme on peut le voir sur cette figure :

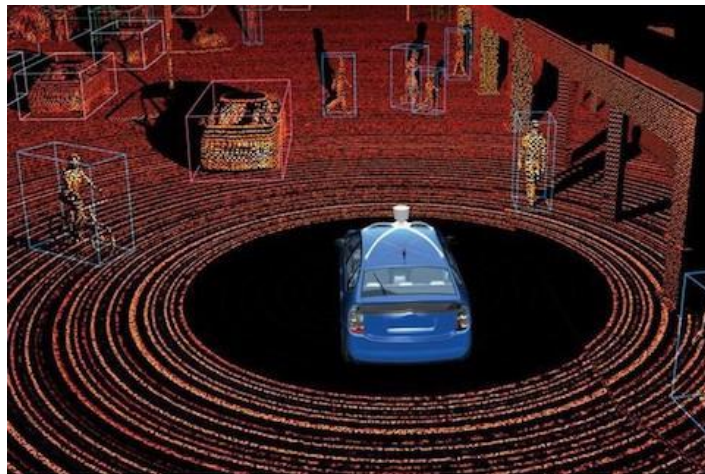


Figure 2.12. Nuage de points de l'environnement.

Avec un grand rayon de détection, il permet de scanner une grande distance (100 m à 300 m) pour estimer la position des objets autour de lui, cependant, sa taille est encombrante, puisqu'il dépasse largement du toit des véhicules [20].



Figure 2.13. Estimation de position avec un Lidar.

2.5.11 La fusion des capteurs

La fusion de capteurs est une étape cruciale pour les véhicules autonomes, elle est utilisée dans l'étape de la perception, qui sert à comprendre le monde qui entoure une voiture en combinant les capteurs présents.

Après avoir pris connaissance des principes de fonctionnement des différents capteurs équipant un robot autonome, on peut les comparer en citant leurs points positifs et négatifs, comme on peut le voir sur cette figure :



Figure 2.14. Comparatif capteurs.

Chacun de ces capteurs a des avantages et des inconvénients, la fusion de capteurs a pour objectif d'utiliser les avantages de chacun pour précisément comprendre son environnement, la caméra est un très bon outil pour détecter les routes, lire les panneaux ou reconnaître un véhicule, tandis que le lidar est plus efficace dans l'estimation des positions des objets, cependant le radar est meilleur dans l'estimation de la vitesse.

La fusion de capteurs est un des sujets les plus importants lorsqu'on parle de véhicules autonomes, car les algorithmes de fusion permettent à un véhicule de comprendre précisément combien y a-t-il d'obstacles, ou bien d'estimer où ils sont et à quelles vitesses ils se déplacent, ce qui améliore grandement la perception de l'environnement du véhicule [20].

2.6 Partie Logicielle (Software)

2.6.1 Arduino IDE

Présentation :

Arduino IDE est le logiciel qui permet de programmer les cartes Arduino, ou d'autres cartes à microcontrôleur, cet Environnement de Développement Intégré (EDI) est né en même temps que la carte Arduino, c'est d'ailleurs une évolution de Wiring et de Process (d'autres langages écrits sous licences libres avant la création de Arduino), le langage utilisé est très proche des langages C et C++ auxquels s'ajoutent les fonctions des nombreuses bibliothèques.



Figure 2.15. Logo Arduino IDE.

Il est écrit en Java (un langage de programmation multiplateforme), ce qui permet un portage facile et une interface quasiment identique, quel que soit le système d'exploitation.

L'IDE d'Arduino permet de regrouper dans le même outil les programmes nécessaires au pilotage de la carte, il comprend également un éditeur de texte, un débogueur/compilateur, une interface permettant de gérer les ports COM et le type de carte, de plus, il fournit de nombreux exemples, installe automatiquement les drivers les plus courants, et assure ensuite la communication avec l'Arduino (téléversement et moniteur série) [23].

La dernière version en date de ce logiciel est 1.8.19, sortie le 20 décembre 2021, c'est également la version utilisée pour ce projet.

Cet Arduino IDE simplifie la programmation pour les cartes d'Arduino ou autre, il ressemble à n'importe quel autre IDE de programmation, c'est-à-dire, il contient un éditeur pour coder et compiler, de plus, il inclut de différents éléments d'aide pour l'utilisateur qui vont au-delà des propres guides simples, mais aussi il a des exemples de code pour différentes fonctionnalités et appareils.

Il est pensé pour que n'importe qui puisse commencer à s'en servir avec des connaissances de base, dans tous les cas il existe de milliers de tutoriels sur internet pour apprendre programmer pour les débutants [24].

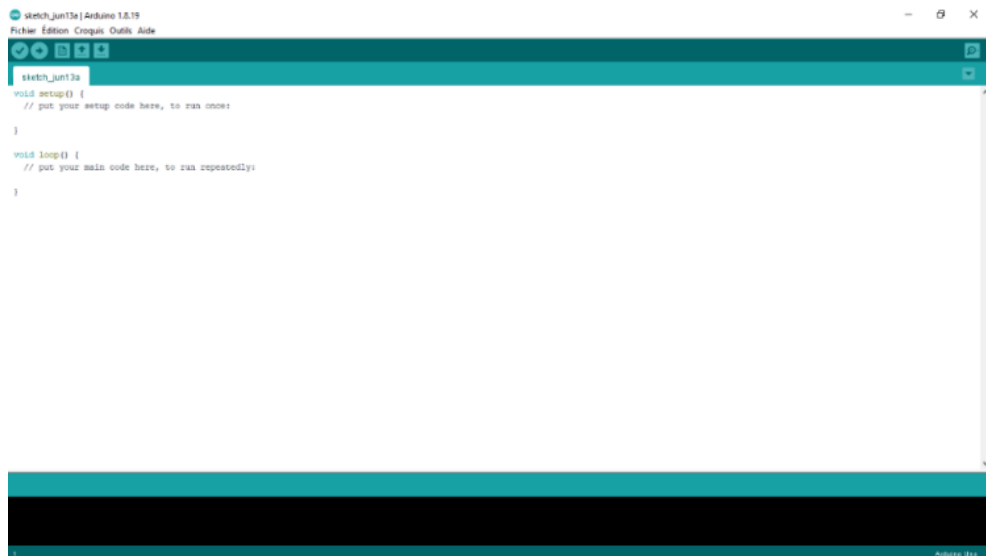


Figure 2.16. Interface Arduino IDE.

Structure d'un programme Arduino :

Un programme Arduino est constitué de trois parties (fonctions) :

Partie 1 : C'est la partie où l'on déclare les variables, on affecte les broches, et aussi on appelle les bibliothèques.

Partie 2 : C'est l'étape d'initialisation (Setup), où l'on configure les broches en entrée ou en sortie, mais aussi pour initialiser les différents objets (comme le moniteur série, ou les composants qui ont leur propre bibliothèque, cette fonction est réalisée une seule fois après la mise sous tension de la carte ou après un reset.

Partie 3 : C'est le programme principal (Loop), cette fonction est répétée indéfiniment.

Comme pour n'importe quel autre logiciel de programmation, on peut rajouter de commentaires, ce qui est vivement conseillé, deux types de commentaires sont possibles :

-Un commentaire après une instruction débute par « // ».

-Un commentaire sous forme de paragraphe commence par « /* » et se termine par « */ ».

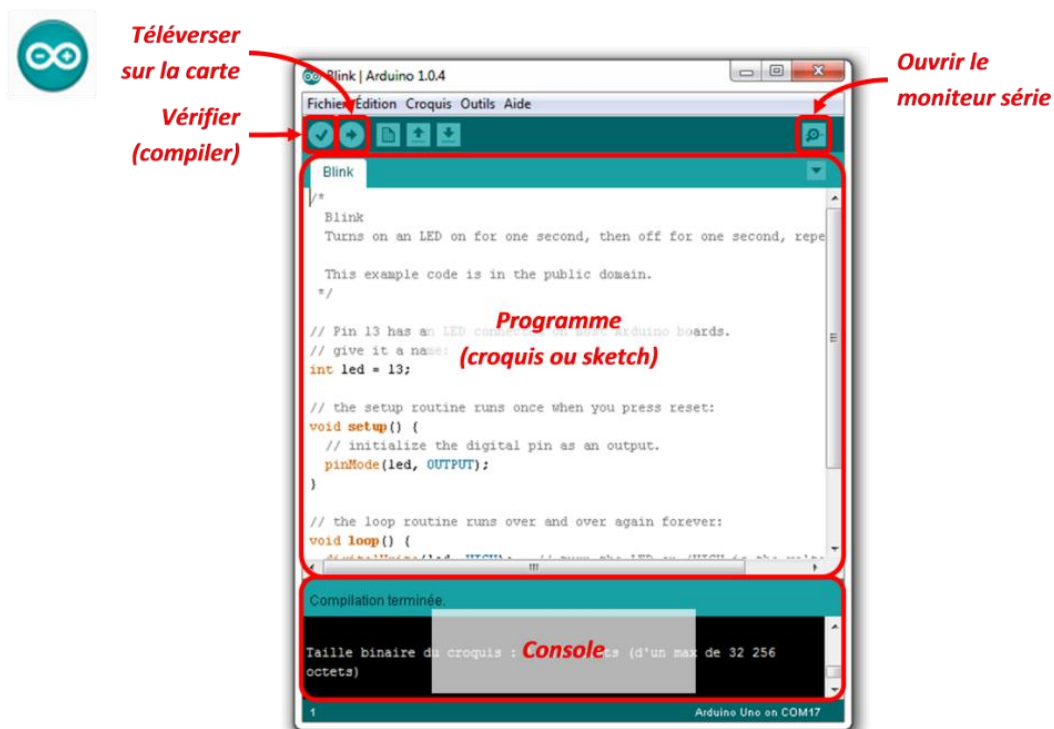


Figure 2.17. Interface Arduino IDE expliquée.

L'IDE d'Arduino possède aussi une aide en ligne où toutes les instructions, structures du langage Arduino sont décrites et accompagnées d'un exemple, pour accéder à cette aide, il faut cliquer sur Aide puis Référence [25].

L'IDE d'Arduino permet :

- D'éditer un programme : des croquis (sketch en Anglais), les programmes sont écrits en langage C.
- De compiler ce programme dans le langage « machine » de l'Arduino, la compilation est une traduction du langage C vers le langage du microcontrôleur.
- D'afficher dans la console les informations sur le déroulement de la compilation et d'afficher les messages d'erreur.
- De téléverser le programme dans la mémoire d'Arduino, le téléversement (upload) se passe via le port USB de l'ordinateur, une fois dans la mémoire de l'Arduino, le logiciel s'appelle un micrologiciel.
- De communiquer avec la carte Arduino grâce au terminal (ou moniteur série). pendant le fonctionnement du programme en mémoire sur l'Arduino, il peut communiquer avec l'ordinateur tant que la connexion est active (câble USB) [26].

Avantages et inconvénients de l'Arduino IDE :

Les plus

- Il s'agit d'un logiciel libre et aucun frais d'abonnement ne sera nécessaire.
- Des outils améliorés et intuitifs permettent aux utilisateurs d'accéder à des utilisations simples ou complexes au choix.
- Communauté active.
- Multiplateforme.
- Système piloté par la communauté.

Les moins

- Arduino IDE est destiné à ceux qui ont une expérience de programmation.
- Le téléversement prend du temps sur certaines cartes.
- Impossible de changer le style de police.
- Les boutons rapides ne peuvent pas être ajoutés.

L'un des avantages de l'EDI Arduino réside dans le fait qu'il est piloté par la communauté, Arduino qui dispose d'un forum très actif où les utilisateurs peuvent partager leurs créations avec d'autres programmeurs et obtenir des commentaires et des conseils de dépannage, les utilisateurs les plus avancés peuvent également délivrer leurs propres conseils, les utilisateurs ont aussi la possibilité de télécharger leur code directement sur les forums ce qui est extrêmement pratique.

Pour quelqu'un qui débute en programmation, il ne fait aucun doute qu'Arduino IDE est l'un des meilleurs programmes qui soit, il est facile à utiliser et contient de nombreux modèles très utiles pour les débutants, de plus, les utilisateurs peuvent profiter de la communauté des utilisateurs d'Arduino pour améliorer leurs créations [27].

2.6.2 Proteus

Proteus est une suite logicielle destinée à l'électronique, développée par la société Labcenter Electronics, les logiciels inclus dans Proteus permettent d'utiliser la CAO (Conception Assistée par Ordinateur), dans le domaine électronique.

Cette suite logicielle est très connue dans le domaine de l'électronique, de nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle, outre la popularité de l'outil, Proteus possède d'autres avantages :

- Pack contenant des logiciels faciles et rapides à comprendre et utiliser.
- Le support technique est performant.
- L'outil de création de prototype virtuel permet de réduire les coûts matériels et logiciels lors de la conception d'un projet.

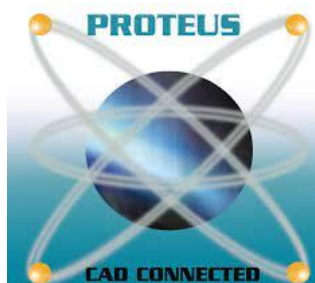


Figure 2.18. Logo Proteus.

ISIS

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques, et permet également de simuler ces schémas, ce qui permet de détecter certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

ARES

Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS, un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le circuit imprimé PCB (Printed Circuit Board), de la carte électronique.

Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement [28].

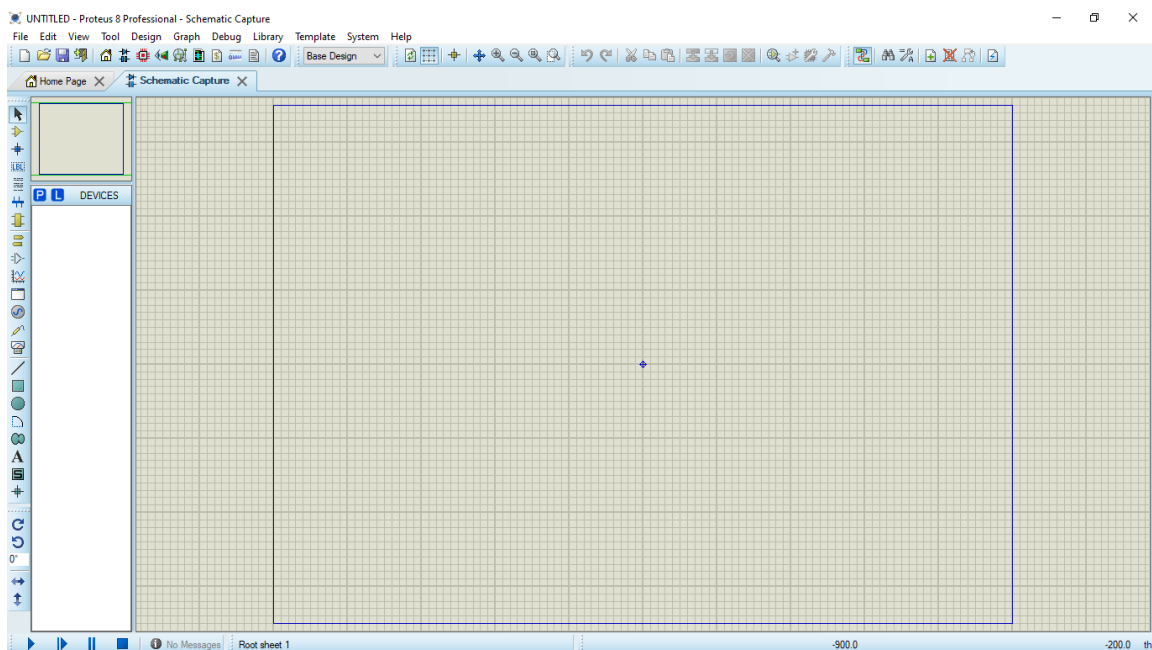


Figure 2.19. Interface Proteus.

Chapitre 3 Réalisation et Simulation de la voiture autonome

3.1 Présentation

3.1.1 Introduction

Dans ce chapitre, nous allons mettre en pratique les connaissances théoriques acquises dans les chapitres précédents pour concevoir et construire un prototype de voiture autonome à l'aide de la plateforme de développement Arduino. Nous allons décrire les différentes étapes de la conception du prototype, notamment la sélection des composants, la programmation des capteurs et des actionneurs, et les tests réalisés pour vérifier le bon fonctionnement du système. Nous allons également évaluer les performances du prototype et discuter des résultats obtenus, en les comparant aux objectifs initialement fixés.

On peut donc commencer la réalisation et la simulation des différents composants de la voiture, puis les assembler pour le montage final, les composants utilisés sont les suivants :

- Capteur à ultrasons.
- Arduino MEGA 2560 R3.
- GPS NEO6M.
- x2 Batterie Lithium 2600mAh.
- Servo moteur.

-Moteur Driver L293D.

-Bouton d'alimentation.

-Module Bluetooth HC-05.

On détaillera par la suite les ajouts sur les différents composants pour mieux voir comment ils interagissent avec la carte Arduino.

Comme on peut le voir sur cette image voici les composants utilisés :

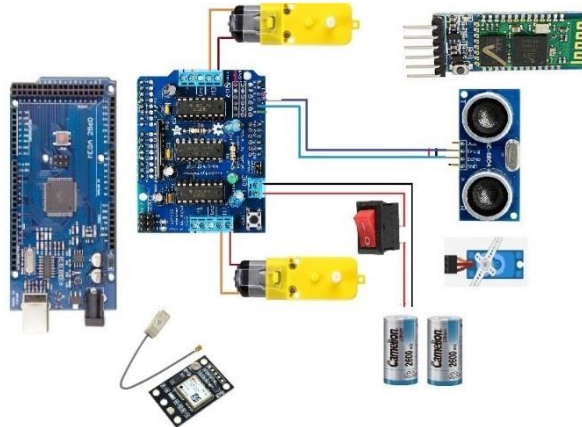


Figure 3.1. Composants du circuit.

On peut aussi tracer le schéma du circuit avec Proteus :

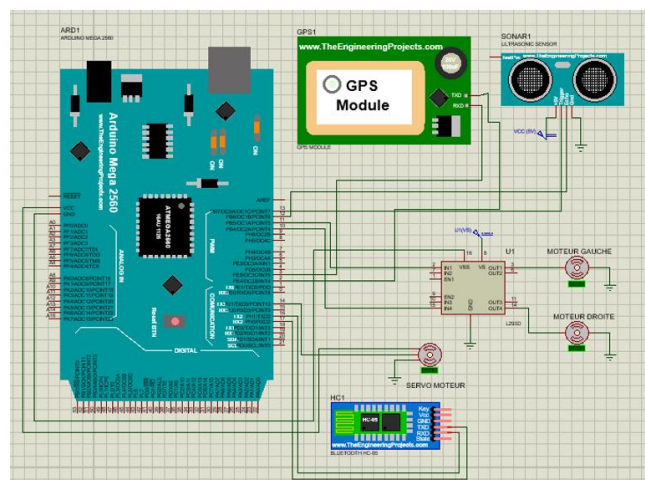


Figure 3.2. Schéma Proteus du circuit.

3.1.2 GPS

Le module GPS NEO-6M est un récepteur GPS complet très performant avec une antenne en céramique intégrée, qui offre une capacité de recherche par satellite puissante, avec les indicateurs d'alimentation et de signal, on peut surveiller l'état du module, et aussi grâce à la pile de sauvegarde des données, le module peut sauvegarder les données en cas de coupure accidentelle de l'alimentation principale.

La référence exacte GPS utilisé est le « NEO-6M-0-001 » de chez u-blox AG, sa fiche technique est détaillée en annexe.



Figure 3.3. GPS GY-NEO6M.

Cela permet également au projet Arduino d'avoir la possibilité de détecter les différents emplacements du véhicule, grâce au module GPS NEO-6M qui peut suivre 22 satellites et identifier des emplacements partout dans le monde.

Ils sont de faible puissance (adaptés aux appareils fonctionnant sur batterie, comme c'est le cas sur ce projet), abordables, faciles à interfacer, mais aussi, le chipset NEO-6M de ces modules est assez bon et précis, et il a également une sensibilité élevée pour les applications intérieures.

Le module fonctionne bien avec une tension en entrée dans la plage de 3,3 à 5 V (grâce à son régulateur de tension intégré).

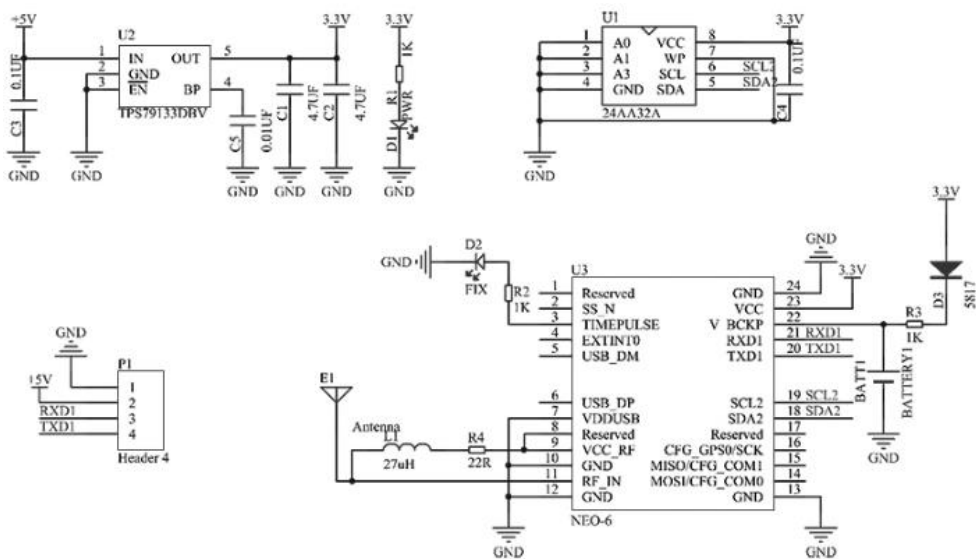


Figure 3.4. Schéma synoptique du GPS NEO6M.

3.1.3 Batterie lithium ion

Une batterie 18650 est une batterie lithium-ion qui peut servir d'alimentation pour des projets mobiles, le nom dérive des dimensions spécifiques de la batterie : 18 mm x 65 mm, pour l'échelle, c'est plus grand qu'une pile AA. La batterie 18650 a une tension de 3,7 V et a entre 2600 mAh et 3500 mAh (milli-ampères-heures), les 2 batteries utilisés dans ce projet ont une capacité de 2600mah chacune, ainsi qu'une tension de 3.7V, si on en assemble 2, on peut avoir 7.4V environ, ce qui suffisant pour alimenter les deux moteurs qui nécessitent une tension de 6V pour tourner en plein régime [29].

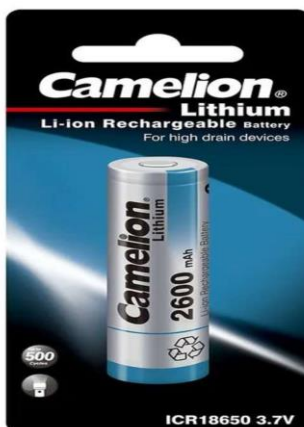


Figure 3.5. Batterie lithium-ion rechargeable.

3.1.4 Servo moteur

Un servomoteur est un type de moteur particulier, sa fonction principale consiste à assurer la production d'un mouvement afin de répondre à une commande externe.

Son signal est codé en largeur d'impulsion (PWM), cela signifie que c'est la durée des impulsions (en général entre 1 et 2 millisecondes) qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur, ce signal est répété périodiquement (en général toutes les 20 millisecondes), ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre, le rôle de l'électronique est de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue : c'est un asservissement [30].

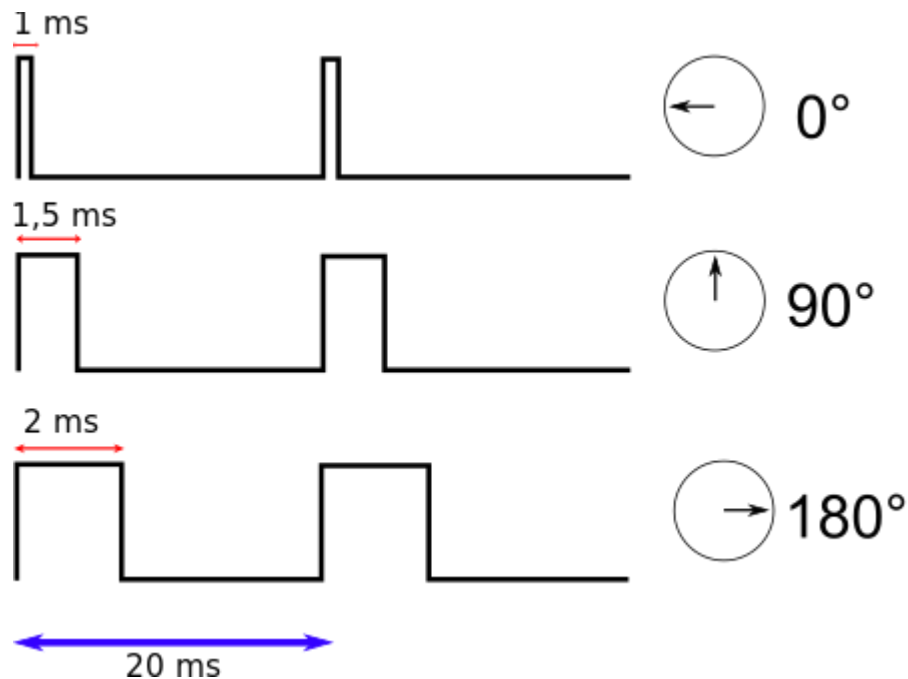


Figure 3.6. Principe de fonctionnement d'un servo moteur.

3.1.5 Module Bluetooth HC-05

Une carte Arduino peut communiquer avec d'autres appareils, par liaison Bluetooth en utilisant un module HC-05 (maître/esclave), cela permet de connecter la carte Arduino à un autre système afin d'envoyer et de recevoir des données.

La communication par Bluetooth permet, par exemple, de piloter un robot via une application Smartphone, récupérer les mesures d'un capteur sur un ordinateur ou encore permettre à deux robots de communiquer entre eux.

Le module HC-05 est très populaire en raison de sa compatibilité avec de nombreux microcontrôleurs et ordinateurs, ainsi que de sa facilité d'utilisation. Il utilise la technologie Bluetooth 2.0 et offre une portée de communication d'environ 10 mètres.

Le module est facile à connecter à un microcontrôleur Arduino via une liaison série UART (Universal Asynchronous Receiver/Transmitter) et peut être contrôlé via des commandes AT (Attention).

HC-05 Bluetooth Module

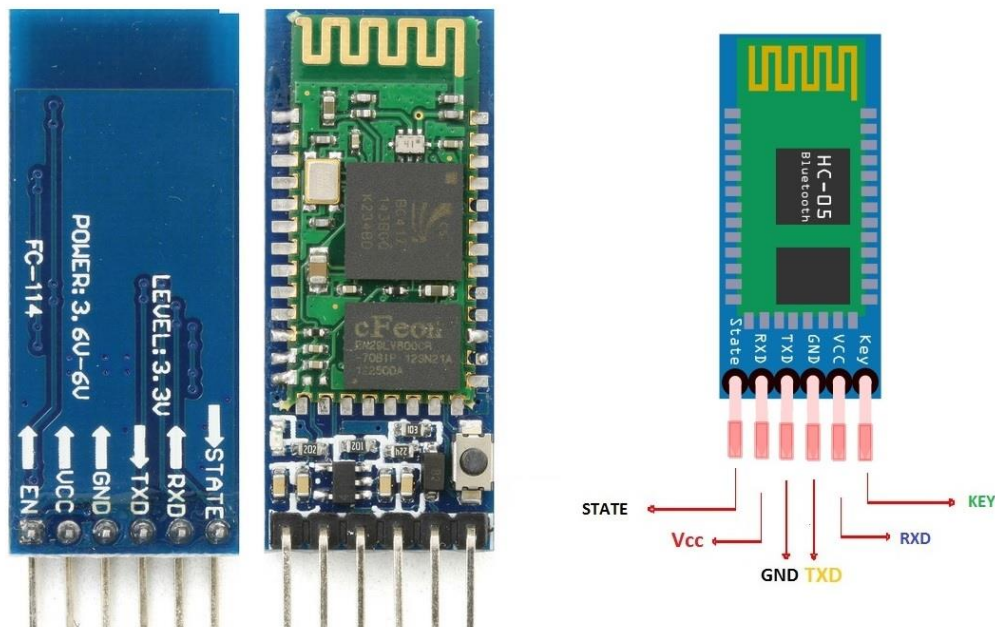


Figure 3.7. Module Bluetooth HC-05.

3.2 Simulation

Pour la simulation, on a utilisé le logiciel Proteus 8.9 présenté dans le chapitre 2, ce logiciel permet de faire la simulation des cartes Arduino, ainsi que les composants qui les accompagnent (capteurs, actionneurs...), tout cela est possible grâce aux bibliothèques téléchargeables de « TheEngineeringProjects ».

Pour simuler de l'Arduino avec Proteus il faut suivre les étapes suivantes :

- Télécharger les bibliothèques des composants.
- Faire le branchement des composants.
- Lancer Arduino IDE, faire le programme nécessaire au montage, puis le convertir en fichier .HEX.
- Insérer le fichier .HEX dans la carte Arduino, ou bien dans un composant programmable pour les simulations sans Arduino.
- Lancer la simulation.

En appliquant ces étapes, on va par la suite simuler les différents composants de notre véhicule autonome.

3.2.1 Simulation du capteur à Ultrasons

Dans cette simulation, on a besoin de : un capteur à ultrasons, une carte Arduino UNO, une LED, un voltmètre, un potentiomètre, ainsi que le « Virtual Terminal », qui permet de visualiser le résultat de la simulation.

Après avoir branché ces composants correctement, et inséré le programme dans la carte, on lance la simulation.

En réalité, le capteur convertit la distance en tension électrique, donc, dans la simulation, pour faire changer la valeur de la distance du capteur à ultrasons, il suffit d'insérer une tension variable de 0 à 5V dans le TestPin du capteur à ultrasons.

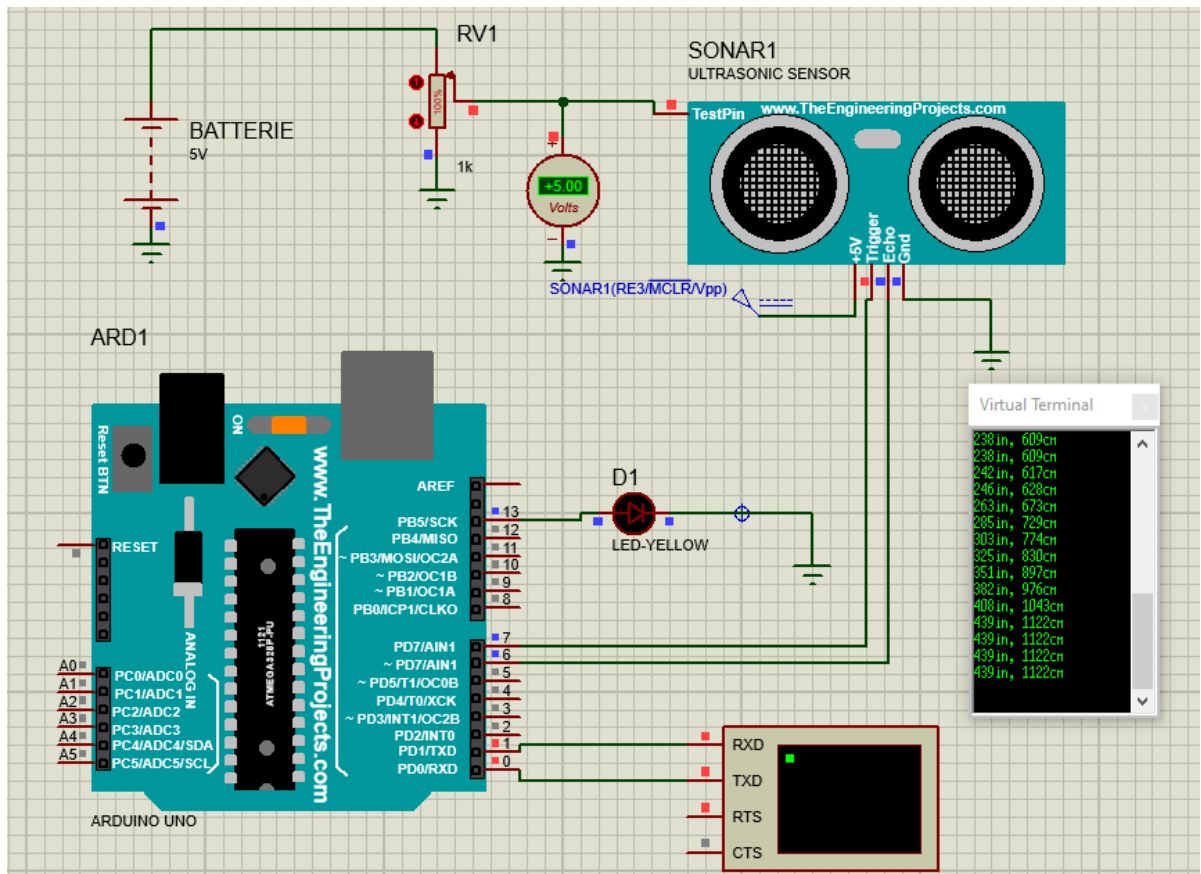


Figure 3.8. Simulation Capteur à Ultrasons.

Comme on peut le voir sur la figure, la variation de la tension de sortie du capteur à ultrasons, permet de modifier la distance affichée, exactement comme en pratique, donc la simulation a été faite avec succès.

3.2.2 Simulation du GPS

Le GPS utilisé dans ce projet est le « NEO6M », on a utilisé 2 types de simulations, une sans arduino, et une avec :

Simulation sans Arduino :

Dans cette simulation, on programme le module GPS avec un fichier .HEX, pour lui permettre de transmettre ses données de localisation, vers le Terminal Virtuel, ensuite ces données générées seront affichées dans cet espace d'affichage, comme on peut le voir sur cette figure :

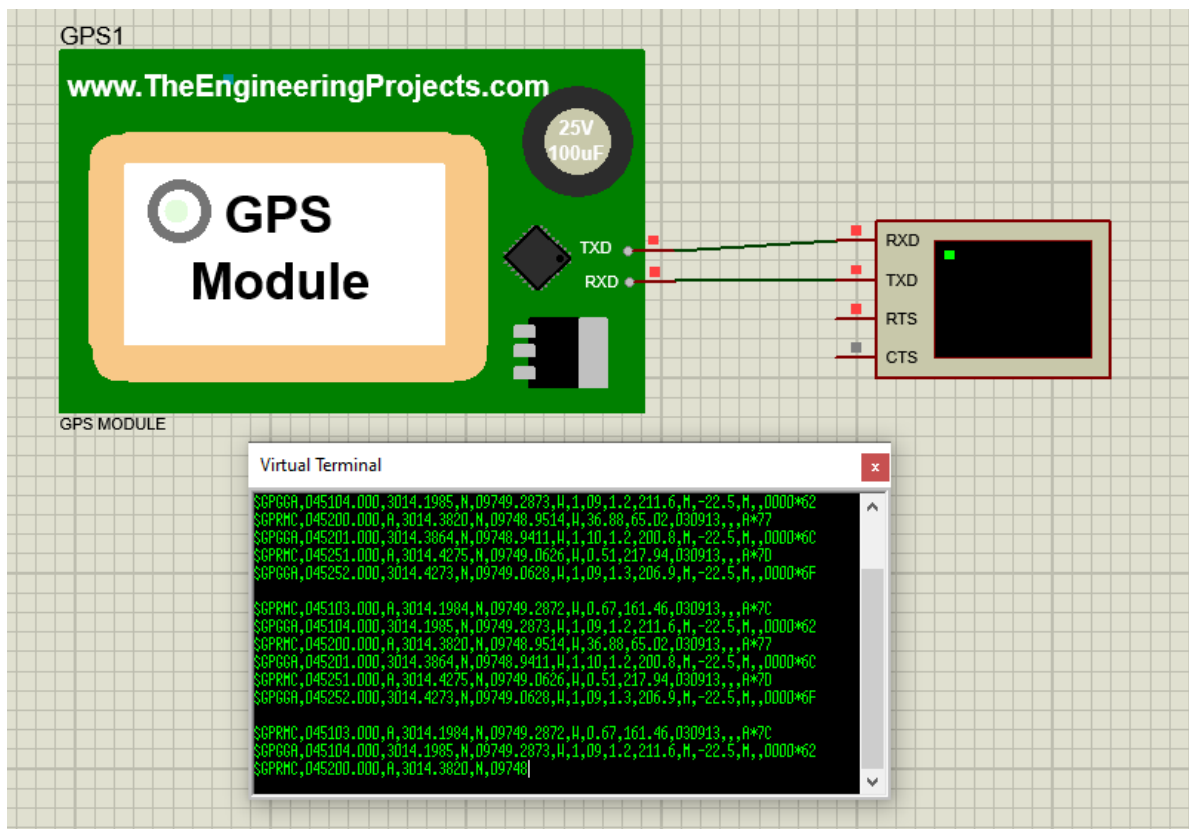


Figure 3.9. Simulation du GPS sans Arduino.

On visualise dans la simulation des codes NMEA, qui correspondent à des coordonnées GPS de localisation, c'est également des codes qui ont une norme pour pouvoir les lire et savoir la longitude et la latitude de la localisation, ce résultat est dû à l'utilisation d'un programme général et inséré comme fichier .HEX.

Simulation avec Arduino :

Dans cette simulation, on ajoute la carte Arduino UNO, on branche le TX (Transmitter) qui est l'émetteur, vers le RX (Receiver) de la carte Arduino, puis on transmet cette donnée du pin TX vers la virtual terminal, qui affichera par la suite le résultat de cette simulation, le code inséré dans l'Arduino est de la bibliothèque « TinyGPS », qui peut être téléchargé directement sur l'Arduino IDE.

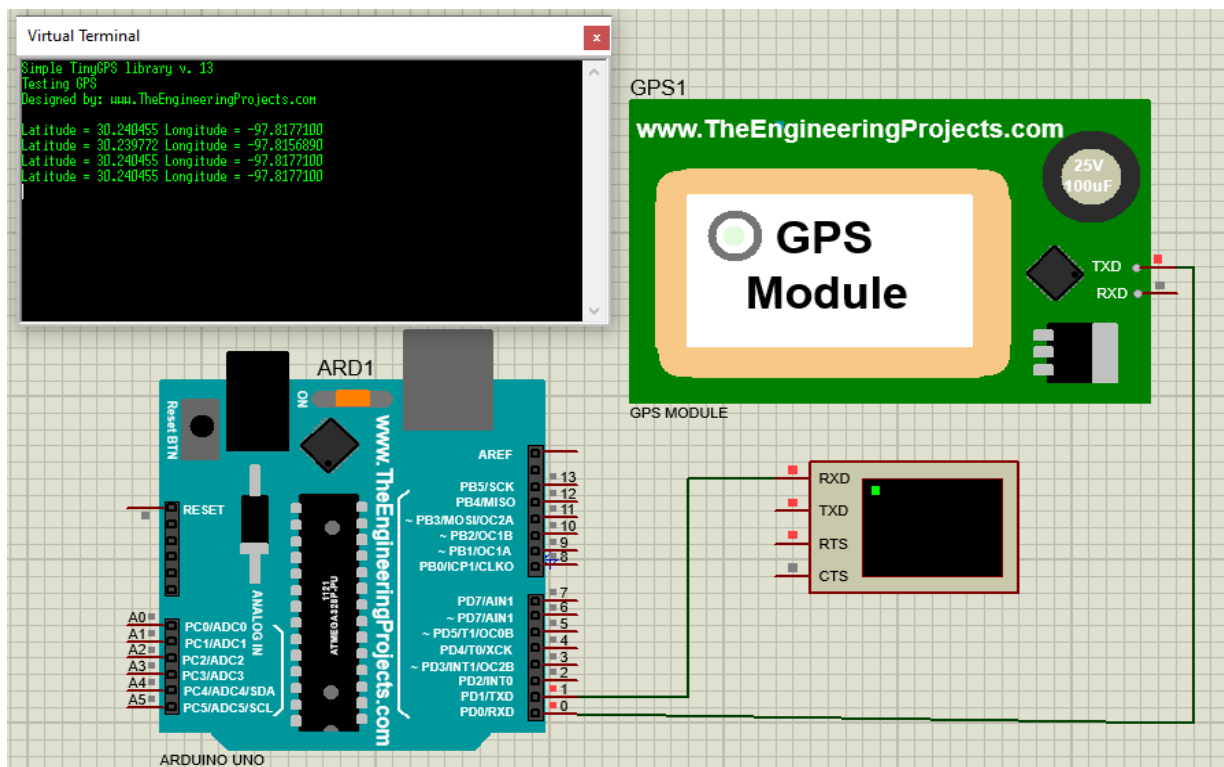


Figure 3.10. Simulation du GPS avec Arduino.

Comme on peut le voir, en changeant le programme inséré dans le module GPS et la carte Arduino, on se permet de mieux déchiffrer les localisations exactes, cela est faisable grâce à la carte Arduino qui simplifie la gestion de ces données, on aura donc la position directement sous les coordonees marines longitude et latitude, ce qui correspond à de la localisation 2D.

3.2.3 Simulation du capteur infrarouge

Le capteur IR, permet de détecter si un obstacle est à proximité, ce dernier est Tout ou Rien (TOR), 5V ou 0V, c'est-à-dire qu'il a uniquement deux états : haut et bas.

Dans cette simulation on ajoute un logic_state pour générer une impulsion à état haut ou bas pour simuler la proximité du capteur, reste plus qu'à ajouter le fichier .HEX au composant IR pour pouvoir le faire fonctionner.

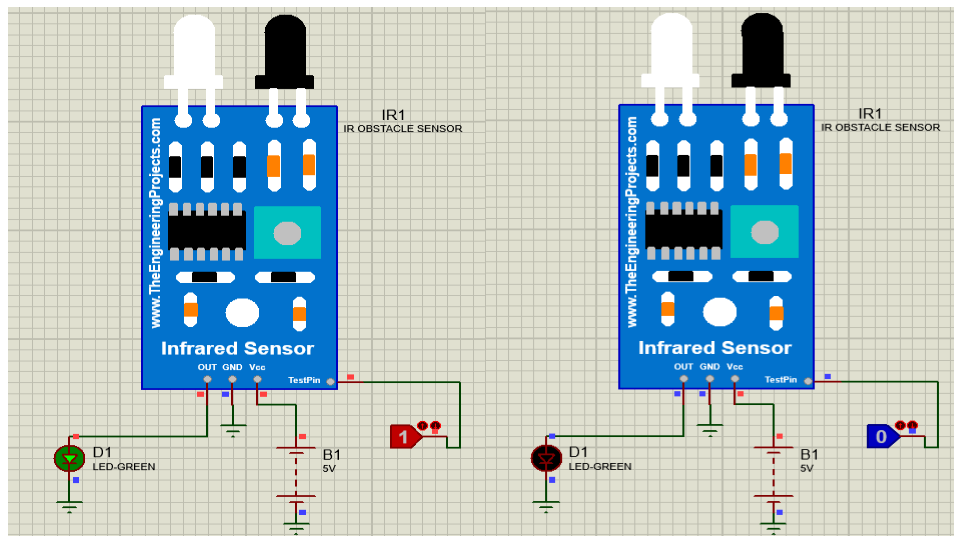


Figure 3.11. Simulation du Capteur Infrarouge.

Comme on peut le voir sur cette figure, si on injecte un '1' logique au Testpin, la sortie sera à '1', dans le cas inverse c'est '0', donc la LED s'allume ou s'éteint selon la sortie, donc on peut simuler la proximité de cette façon, comme en pratique.

3.2.4 Simulation des Moteurs

Avec Proteus, on peut simuler les moteurs à courant continu avec ou sans driver :

Sans Driver :

On ne peut pas contrôler des moteurs directement branchés à l'Arduino, car celui-ci ne fournit pas un courant suffisant pour tourner les moteurs, mais dans ce cas on peut alimenter les moteurs avec des batteries plus performantes, comme ce cas ici ou on a des logic_state qui permettent de choisir la logique (5V ici), pour alimenter les moteurs dans le sens souhaité comme on peut le voir sur cette figure :

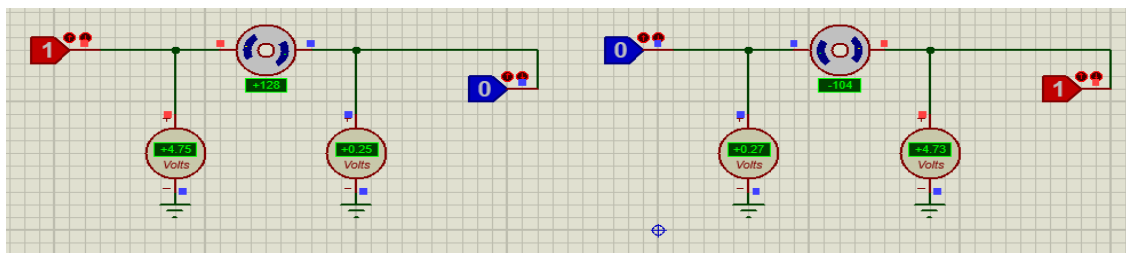


Figure 3.12. Simulation des moteurs à courant continu sans driver.

Avec Driver :

Le driver utilisé dans cette simulation est le L298N, qui permet de contrôler jusqu'à 2 moteurs, en branchant des sources au pins IN1, IN2, IN3, et IN4 pour choisir le sens de rotation, ensuite on branche les deux moteurs au pins OUT1, OUT2, OUT3 et OUT4 pour le lier, ensuite on alimente le driver avec une tension de 12V DC pour alimenter les deux moteurs correctement.

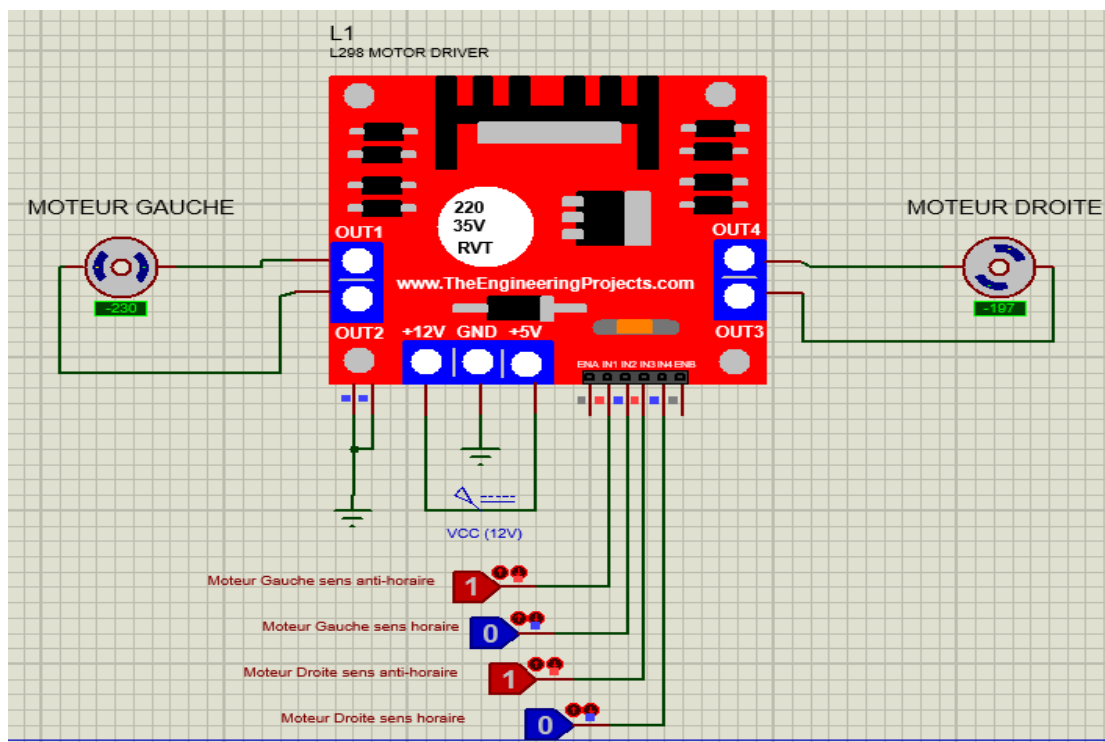


Figure 3.13. Simulation des Moteurs avec le Driver L298N.

Comme on peut le voir, pour commander le sens des moteurs avec un pont en H il suffit soit de modifier la polarisation des moteurs au driver, soit de changer l'injection des impulsions dans le driver pour choisir directement le sens dans lequel on souhaite tourner les moteurs, comme en pratique, aussi si on souhaite augmenter la vitesse des moteurs, il suffit d'augmenter la tension pour augmenter le nombre de tours / min, cela simplifie grandement l'usage des moteurs.

3.3 Réalisation

3.3.1 Châssis de la voiture

Tout d'abord, il faut assembler le châssis de la voiture, qui inclue :

- Châssis supérieur et inférieur.
- Deux moteurs à courant continu.
- 2 roues motrices.
- Une roue libre.
- Câbles.
- Vis et divers ajouts du châssis.

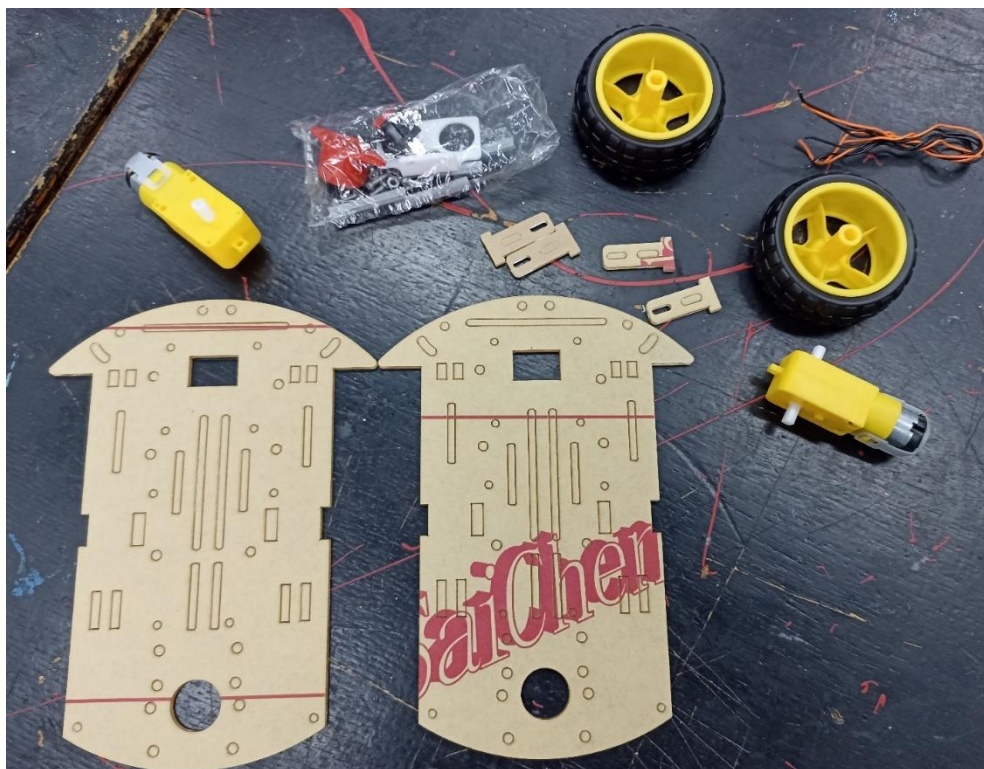


Figure 3.14. Chassis de la voiture avant le montage.

3.3.2 Test du GPS

On effectue le branchement suivant :

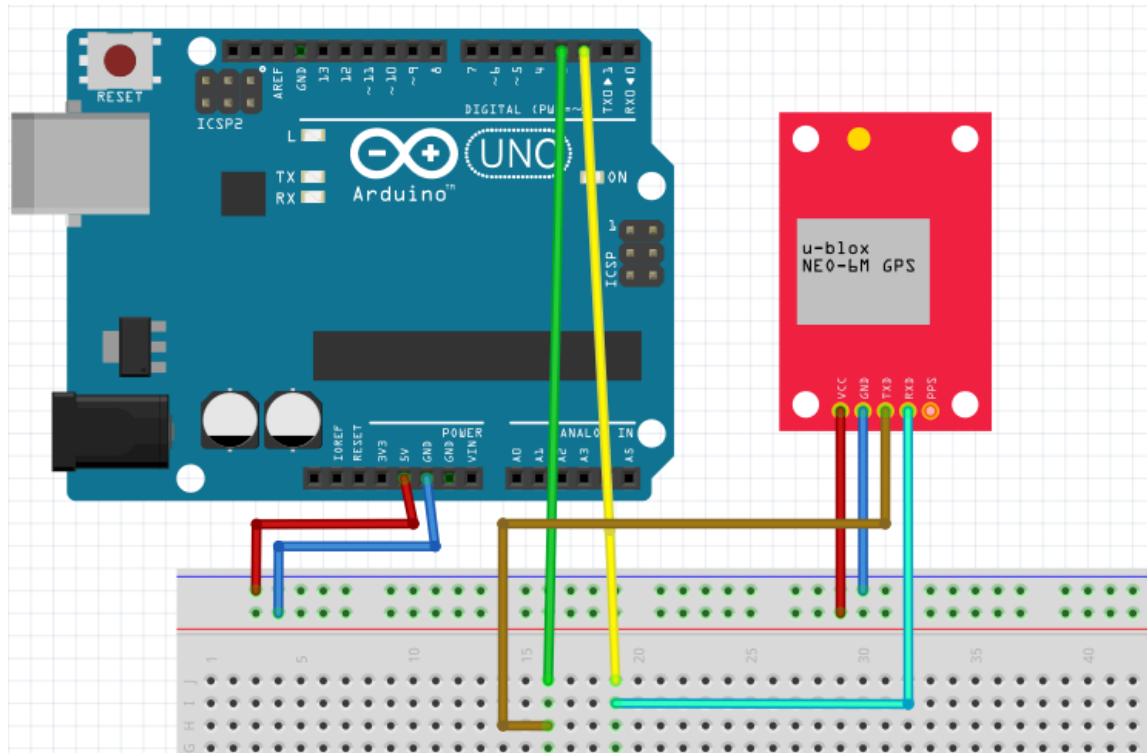


Figure 3.15. GPS avec Arduino.

On aura le résultat suivant dans le moniteur série de l'Arduino IDE :

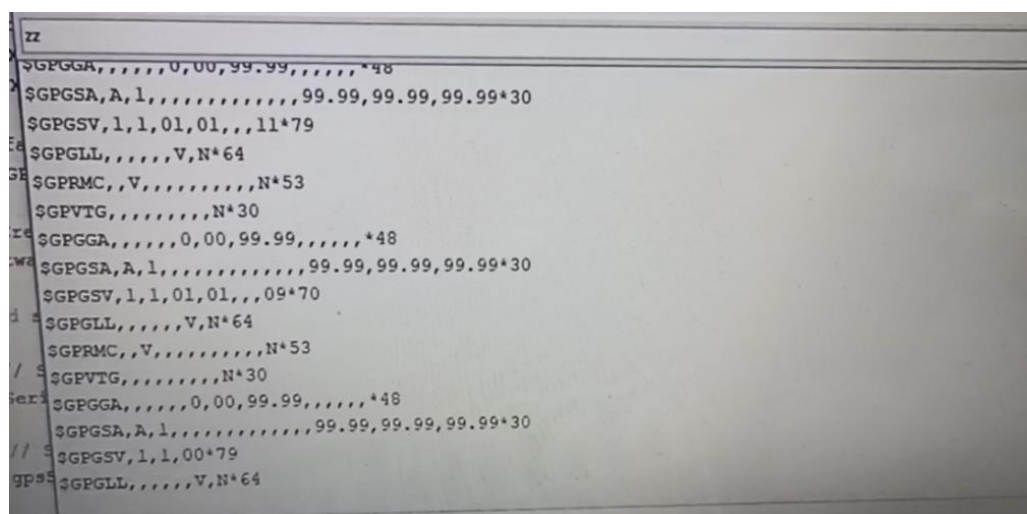


Figure 3.16. Résultat du GPS.

La phrase \$GPGGA fournit des données fixes essentielles qui fournissent les données de localisation et de précision 3D.

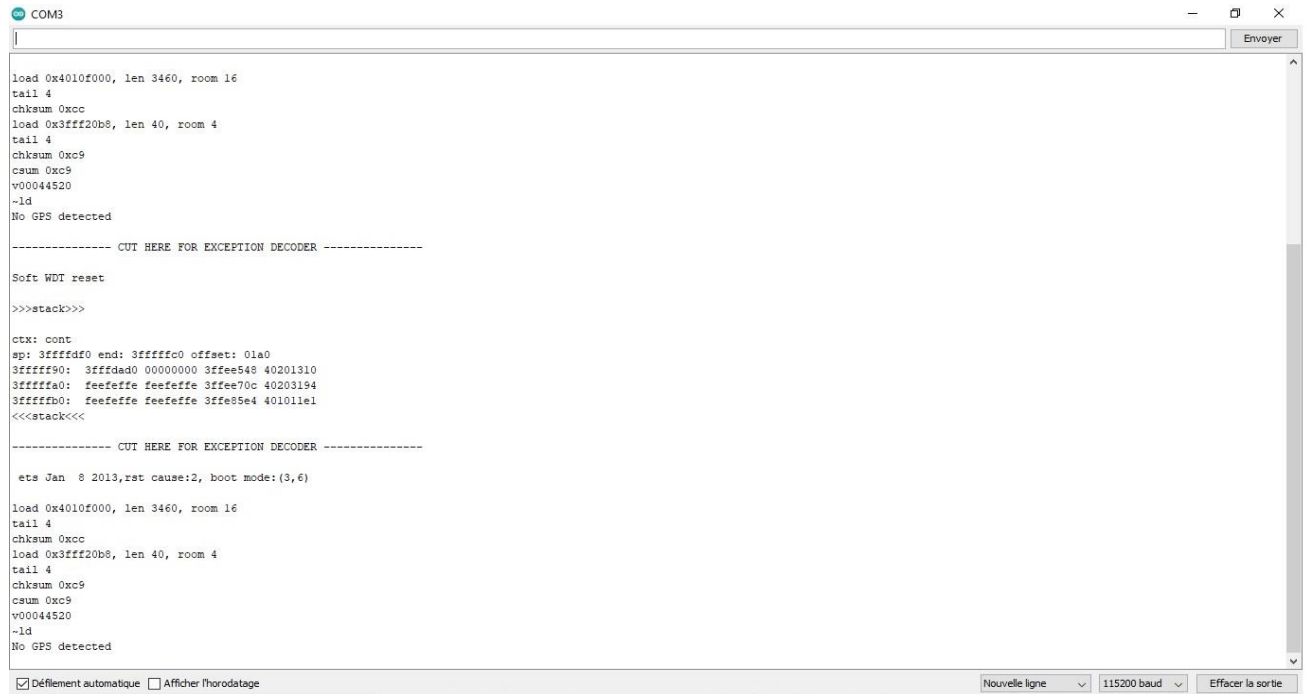
Phrase \$GPGGA NMEA

Prenons un exemple de phrase \$GPGGA NMEA :

- \$GPGGA, 123519, 4807.038, N, 01131,000, E, 1, 08, 0.9, 545.4, M, 46.9, M, *47.
- \$ Début de phrase NMEA.
- Données fixes du système de positionnement global GPGGA.
- 123519 Heure actuelle en UTC – 12:35:19.
- 4807.038 N Latitude 48° 07.038' N.
- 01131.000 E Longitude 11 deg 31.000' E.
- Repère GPS.
- Nombre de satellites suivis.
- Dilution horizontale de la position.
- 545.4 M Altitude en mètres (au-dessus du niveau moyen de la mer).
- 46.9 M Altitude du géoïde (niveau moyen de la mer).
- Temps en secondes depuis la dernière mise à jour DGPS.
- Numéro d'identification de la station DGPS.
- Les données de la somme de contrôle commencent toujours par « * ».

\$GPRMC et \$GPGGA sont des phrases GPS NMEA de base. Il existe de nombreuses phrases NMEA alternatives et complémentaires qui fournissent des informations similaires ou supplémentaires.

S'il n'y pas de signal GPS, cela surtout arrive en intérieur on aura le message suivant dans le moniteur série de l'Arduino IDE :



```
COM3
load 0x4010f000, len 3460, room 16
tail 4
chksum 0xcc
load 0x3fff20b8, len 40, room 4
tail 4
chksum 0xc9
csun 0xc9
v00044520
-ld
No GPS detected

----- CUT HERE FOR EXCEPTION DECODER -----

Soft WDT reset

>>>stack>>>

ctx: cont
sp: 3ffffd0 end: 3ffffc0 offset: 01a0
3ffff90: 3ffffd0 00000000 3ffef548 40201310
3ffffa0: feeffe feeffe 3ffef70c 40203194
3ffffb0: feeffe feeffe 3ffef85e4 401011e1
<<<stack<<<

----- CUT HERE FOR EXCEPTION DECODER -----

ets Jan 8 2013,rst cause:2, boot mode:(3,6)

load 0x4010f000, len 3460, room 16
tail 4
chksum 0xcc
load 0x3fff20b8, len 40, room 4
tail 4
chksum 0xc9
csun 0xc9
v00044520
-ld
No GPS detected

 Défilement automatique  Afficher l'historique
Nouvelle ligne 115200 baud Effacer la sortie
```

Figure 3.17. Résultat du GPS en cas d'échec de localisation.

3.3.3 Test du Capteur à ultrasons

On réalise en pratique le schéma suivant :

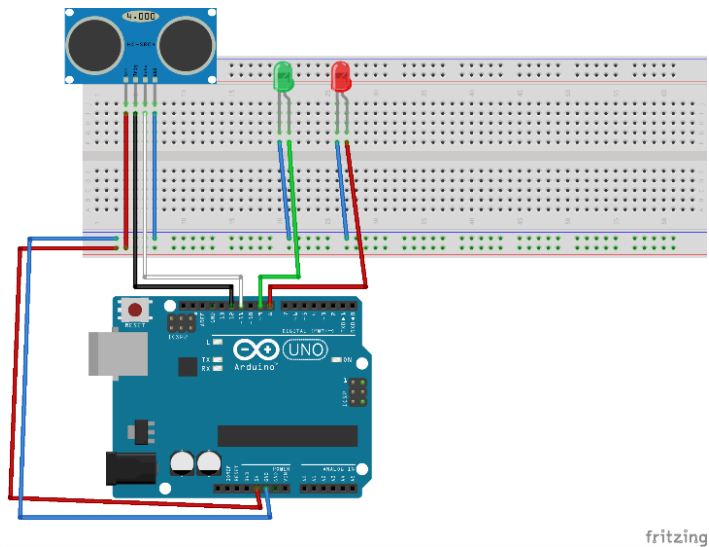


Figure 3.18. Schéma de test d'un capteur ultrasonore.

On aura le résultat :

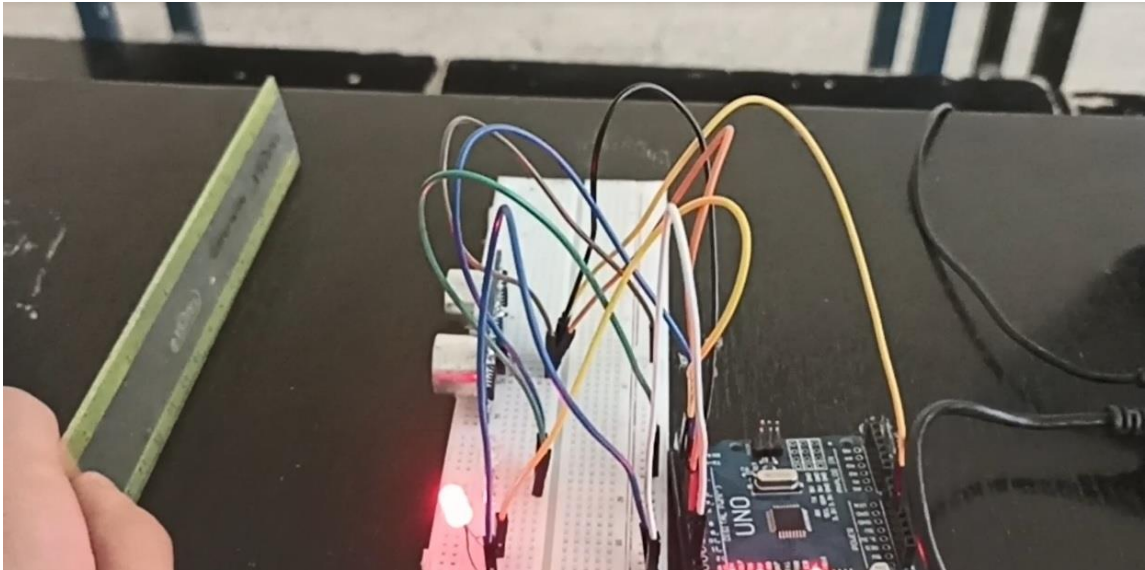


Figure 3.19. Présence d'un obstacle.

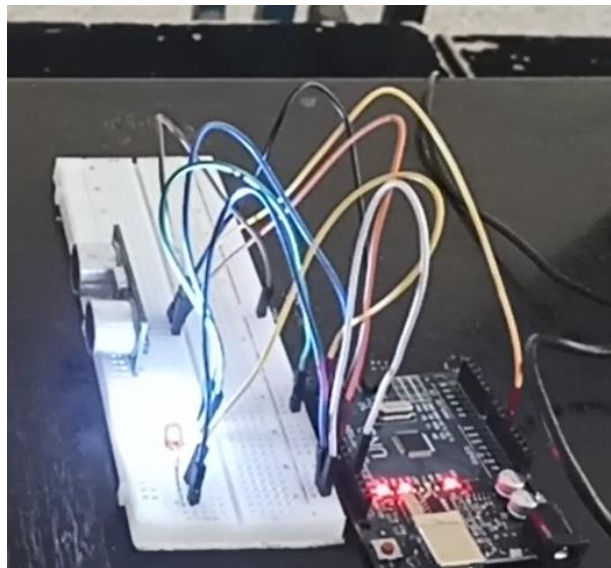
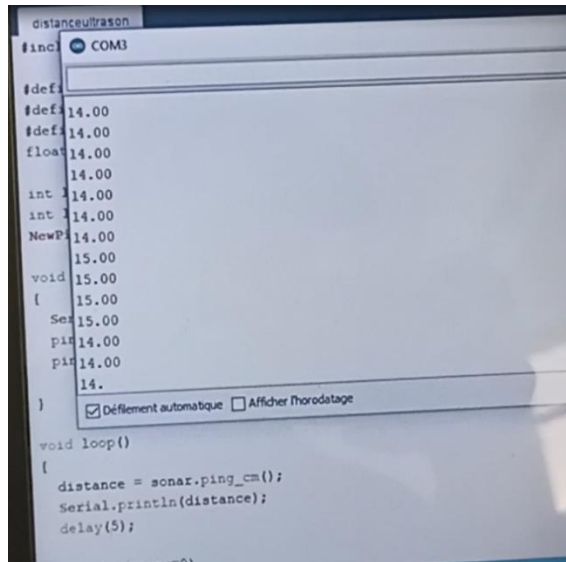


Figure 3.20. Sans présence d'obstacle.

S'il y'a un obstacle, la LED rouge s'allume, sinon c'est la bleue, c'est selon si la distance si elle est inférieure ou supérieure à 200mm.

On peut aussi visualiser la distance de l'obstacle dans l'Arduino IDE :



```
distanceultrason
#include <COM3>
#define
#define 14.00
#define 14.00
float 14.00
14.00
int 14.00
int 14.00
NewP 14.00
15.00
void 15.00
{ 15.00
  Ser 15.00
  pin 14.00
  pin 14.00
  14.
}
 Défilement automatique  Afficher l'horodatage

void loop()
{
  distance = sonar.ping_cm();
  Serial.println(distance);
  delay(5);
}
```

Figure 3.21. Distance de l'obstacle sur le moniteur série.

On peut même pour tester, visualiser cette valeur sur matlab en envoyant ces données en transmission série via le port SERIAL, et on y ajoute une petite interface graphique :

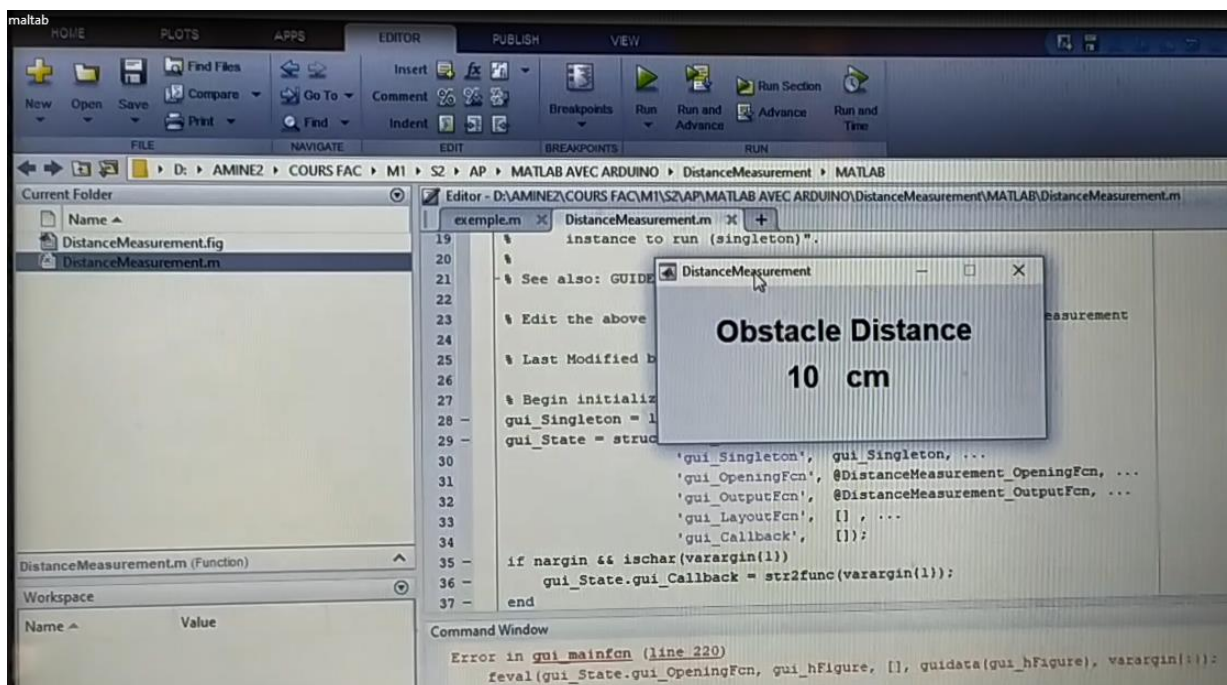


Figure 3.22. Distance de l'obstacle sur matlab.

3.3.4 Test du module Bluetooth avec l'application

L'application Bluetooth utilisée s'appelle « Bluetooth Terminal HC-05 », téléchargeable gratuitement sur le Google Play Store, a été utilisée dans ce projet pour permettre l'envoi de données de la voiture vers un téléphone portable à proximité.



Figure 3.23. Icône de l'Application Bluetooth

Après que le GPS ait réussi à localiser le robot mobile par satellite, L'Arduino peut transmettre directement ces données à l'application mobile grâce au module HC-05. Comme on peut le voir sur la figure, les données captées du GPS, ont été transmises correctement vers l'application :



Figure 3.24. Réception des données GPS.

3.3.5 Test de la voiture

Ensuite on passe au test des moteurs, on laisse la voiture bouger dans toutes les directions pour voir le comportement des roues et des moteurs ainsi que l'autonomie des batteries, comme on peut le voir sur cette figure l'assemblage final du robot mobile :

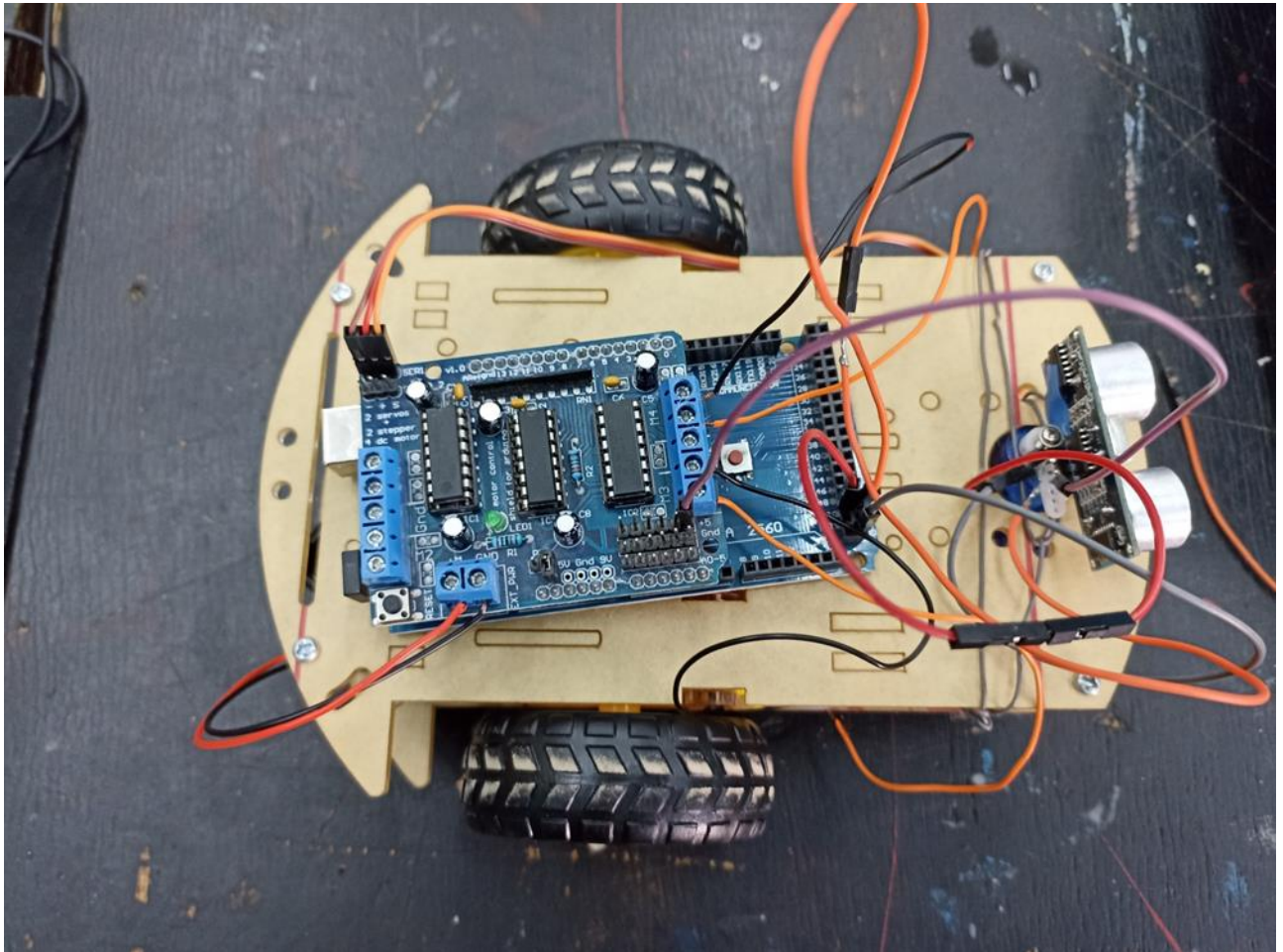


Figure 3.25. Test de la voiture.

La voiture réussit donc à bouger dans toutes les directions sans aucun problème de motricité, ou de manque d'alimentation. Les moteurs tournent facilement à plein régime, donc les batteries choisies suffisent pour les alimenter.

Après différents tests sur les différents composants, on conclue donc que les résultats de la simulation ainsi que de la réalisation correspondent et sont assez proches, donc l'expérience a été faite avec succès.

3.4 Organigramme et Schémas

3.4.1 Organigramme général d'évitement des obstacles

Dans l'organigramme qui suit, il y'a une vérification continue pour savoir s'il y'a une voie libre ou non, cela se fait selon la détection du capteur à ultrasons placé sur le servo moteur qui tourne dans 3 sens : en face, à gauche et à droite, et qui scannerà donc les environs et les distances pour savoir la présence des obstacles.

C'est ensuite selon ces conditions qu'il donnera ces informations à l'Arduino, qui par la suite gère le mouvement de véhicule, en donnant les ordres aux moteurs.

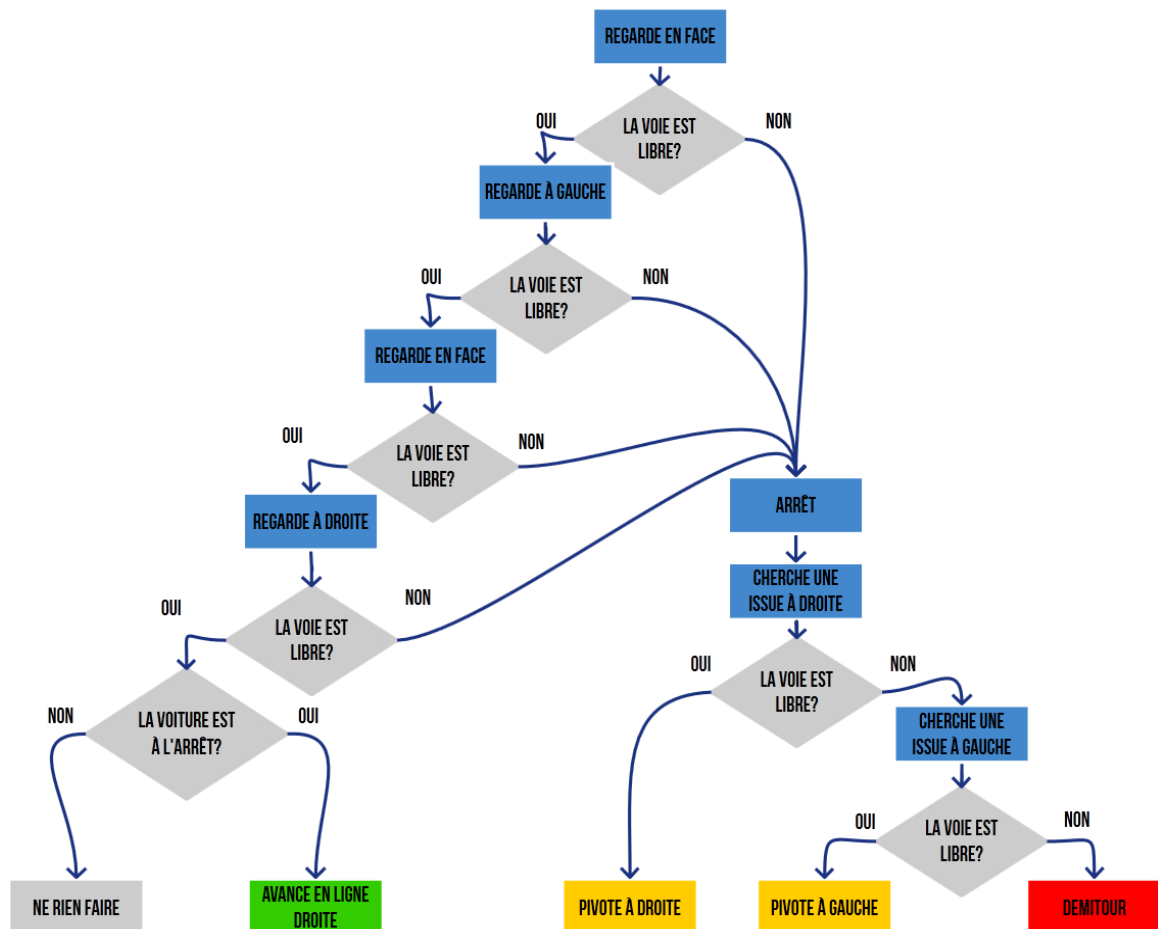


Figure 3.26. Organigramme d'évitement d'obstacle.

3.4.2 Lecture de distance pour actionner les moteurs et les LEDs d'affichage

Dans le schéma de la figure qui suit, le capteur à ultrasons capte la distance de l'obstacle, selon les positions du servo moteur, qui regardera dans toutes les directions possibles pour chercher une issue en cas de présence d'obstacles, cette dernière permettra alors de donner l'applicatif (la carte Arduino), puis donner l'ordre aux moteurs de tourner dans la direction adaptée à la situation, ainsi que l'allumage de LEDs pour avertir l'entourage.

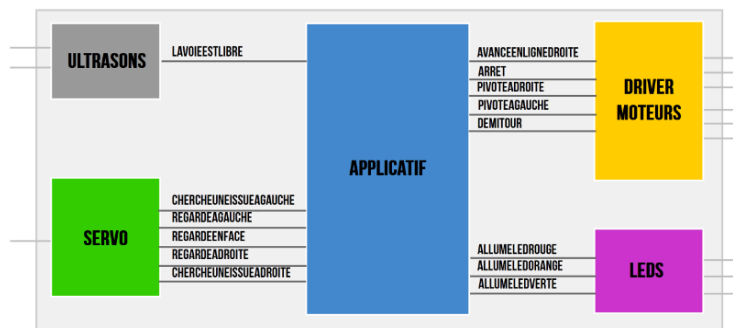


Figure 3.27. Schéma d'instructions général.

Pour la figure qui suit, le capteur à ultrasons émet un ECHO (sortie), et le TRIG (entrée), ces données seront transmises à l'Arduino, qui émet un ordre au servomoteur de tourner dans les directions citées précédemment, la PWM est nécessaire pour contrôler analogiquement le degré du servo moteurs, ce qui donnera par la suite l'ordre aux moteurs de tourner dans le sens nécessaire au déplacement de la voiture, et les LEDs pour signaler la présence des obstacles autour.



Figure 3.28. Schéma des entrées/sorties.

Conclusion générale

L'évolution de la technologie permet d'innover et d'aller loin dans la recherche scientifique, comme on a pu le voir sur ce mémoire de master, l'évolution de la voiture autonome au fil du temps, s'accélère de plus en plus, surtout avec le développement de technologies de transport, comme dans le futur, qui nous réserve l'arrivée des véhicules électriques et autonomes.

Les objectifs de ce projet ont été d'essayer d'utiliser les différents moyens qui constituent l'univers de la programmation Arduino pour arriver à réaliser un prototype d'une voiture autonome, capable d'être assez indépendante pour se déplacer de manière sécurisée.

L'intérêt de ce projet est de laisser la conduite à la machine pour diminuer le facteur de taux d'erreur humain, cela permettra dans le futur d'avoir des véhicules entièrement autonomes (niveau 5), et ainsi de baisser le taux d'accidents de voitures qui peuvent être causées par un facteur humain.

Les moyens mis en œuvre comme la programmation par l'Arduino IDE, ainsi que la simulation par Proteus nous ont permis d'arriver à avoir des bons résultats satisfaisants que ce soit sur le plan théorique ou pratique.

Comme perspectives, on peut améliorer ce projet en y ajoutant de l'intelligence artificielle avec Raspberry Pi, avec des modèles d'apprentissage, aussi on peut y ajouter de la transmission à plus grande distance pour un meilleur suivi du véhicule.

Datasheet de la carte Arduino MEGA 2560

- Microcontrôleur : ATmega2560.
- de 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM).
- de 16 entrées analogiques (qui peuvent être utilisées en broches entrées/sorties numériques).
- de 4 UART (port série matériel).
- d'un quartz de 16MHz.
- 3 ports série.
- Bus I2C et SPI.
- Gestion des interruptions.
- Version : Rev 3.
- d'une connexion USB.
- d'un connecteur d'alimentation jack.
- d'un connecteur ICSP (programmation "in-circuit").
- d'un bouton de réinitialisation (reset).
- Dimensions : 107 x 53 x 15 mm.
- Intensité par E/S : 40 mA.
- Mémoire flash : 256 KB.

- Mémoire SRAM : 8 KB.

- Mémoire EEPROM : 4 KB.

Datasheet du driver moteur L293d

2 connexions pour les servos 'hobby' 5V connectés à la minuterie dédiée haute résolution de l'Arduino.

4 ponts en H : le jeu de puces L293D fournit 0,6 A par pont (1,2 A en crête) avec une protection contre l'arrêt thermique et des diodes internes de protection contre les rebonds. Peut faire fonctionner des moteurs sur 4.5V DC à 25V DC.

Jusqu'à 4 moteurs CC bidirectionnels avec sélection de vitesse individuelle sur 8 bits (donc, environ 0,5 % de résolution).

Jusqu'à 2 moteurs pas à pas (unipolaires ou bipolaires) à simple bobinage, double bobinage ou pas à pas entrelacés.

Les résistances d'abaissement maintiennent les moteurs désactivés pendant la mise sous tension.

Grands connecteurs de bornier pour brancher facilement les fils (18-26AWG) et l'alimentation.

Bouton de réinitialisation Arduino en haut et bornier à 2 broches et cavalier pour connecter l'alimentation externe, pour alimentations séparées logique/moteur.

Datasheet du moteur « Adafruit DC Motor »

Moteur de boîte de vitesses TT DC avec un rapport de démultiplication de 1:48, et il est livré avec 2 fils de 200 mm.

Vous pouvez alimenter ces moteurs avec 3V DC jusqu'à 6V DC, ils iront bien sûr un peu plus vite aux tensions plus élevées. Nous avons saisi un moteur et trouvé ces statistiques lors de son fonctionnement à partir d'une alimentation de table.

Détails techniques :

- Tension nominale : 3 ~ 6 V.
- Courant continu à vide : 150 mA +/- 10 %.
- Min. Vitesse de fonctionnement (3V) : 90+/- 10% RPM.
- Min. Vitesse de fonctionnement (6V) : 200+/- 10% RPM.
- Couple de décrochage (3V) : 0.4kg.cm.
- Couple de décrochage (6V) : 0.8kg.cm.
- Rapport de démultiplication : 1:48.
- Dimensions du corps : 70 x 22 x 18 mm.
- Longueur des fils : 200 mm et 28 AWG.
- Poids : 30,6 g.

Datasheet du GPS « GY-NEO-6M-0-001 »

- Type de récepteur 50 canaux, GPS L1(1575.42MHz).
- Précision de la position horizontale 2,5 m.
- Taux de mise à jour de la navigation 1 Hz (5 Hz maximum).
- Temps de capture Démarrage à froid : 27 s.
- Démarrage à chaud : 1 s.
- Sensibilité de navigation -161dBm.
- Protocole de communication NMEA, UBX binaire, RTCM.
- Débit en bauds série 4800-230400 (par défaut 9600).
- Température de fonctionnement -40°C ~ 85°C.
- Tension de fonctionnement 2,7 V ~ 3,6 V.
- Courant de fonctionnement 45mA.
- Impédance TXD/RXD 510Ω.

Bibliographie

- [1] Olivier Lejeune, Robotique, positron-libre, (consulté le 27/03/2022), disponible sur : <https://www.positron-libre.com/robotique/robotique.php>
- [2] CCM, Robotique, linternaute, (consulté le 25/03/2022), disponible sur : <https://www.linternaute.fr/dictionnaire/fr/definition/robotique/>
- [3] Voiture autonome, Kidioui, (consulté le 15/03/2022), disponible sur : <https://voiture.kidioui.fr/lexique-automobile/voiture-autonome.html>
- [4] Robotique, CEA, (consulté le 30/03/2022), disponible sur : <https://www.cea.fr/comprendre/Pages/nouvelles-technologies/essentiel-sur-robotique.aspx>
- [5] Robotique, journaldunet, (consulté le 31/03/2022), disponible sur : <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501853-robotique-definition-cas-d-application-et-lois/>
- [6] Robotique industrielle, Actemium, (consulté le 31/03/2022), disponible sur : <https://www.actemium.fr/offres/robotique/la-robotique-industrielle-en-bref/>
- [7] FUTURA, Robotique, futura-sciences, (consulté le 18/03/2022), disponible sur : <https://www.futura-sciences.com/tech/definitions/robotique-robotique-603/>
- [8] Niveaux d'autonomie des véhicules, usine-digitale, (consulté le 16/04/2022), disponible sur : <https://www.usine-digitale.fr/article/reperes-quels-sont-les-differents-niveaux-d-autonomie-des-vehicules.N596558> ,
- [9] Lukas, niveaux d'autonomie d'un véhicule autonome, meilleure-innovation, (consulté le 17/04/2022), disponible sur : <https://www.meilleure-innovation.com/niveaux-vehicule-autonome/>

[10] Jeremy Cohen, autonomie, Medium, (consulté le 6/9/2022), disponible sur : <https://medium.com/france-school-of-ai/lia-et-le-v%C3%A9hicule-fut-autonome-2abd8a348eb5>

[11] IA, talend, (consulté le 15/05/2022), disponible sur : <https://www.talend.com/fr/resources/what-is-artificial-intelligence/>

[12] Arduino, mémoire online, (consulté le 11/06/2022), disponible sur : https://www.memoireonline.com/10/20/11898/m_Station-meteorologique-sur-arduino3.html

[13] Introduction à l'Arduino, Arduino, (consulté le 7/9/2022), disponible sur : <https://www.arduino.cc/en/Guide/Introduction/>

[14] Atmega2560, composants 101, (consulté le 15/06/2022), disponible sur : <https://composants101.com/microcontrôleurs/atmega2560-microcontrôleurs>

[15] Guide des balais des moteurs, mersen, (consulté le 6/9/2022), disponible sur : <https://www.mersen.com/sites/default/files/publications-media/2-ptt-guide-technique-balais-mersen.pdf>

[16] PWM, interface-z, (consulté le 10/07/2022), disponible sur : <https://www.interface-z.com/conseil/pwm.php>

[17] Capteur à ultrasons, microsonic, (consulté le 20/07/2022), disponible sur : <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe.html>

[18] Ultrasonic sensors, banner engineering, (consulté le 25/7/2022), disponible sur : <https://www.bannerengineering.com/fr/fr/company/expert-insights/ultrasonic-sensors-101.html#/>

[19] GPS, VROUM, (consulté le 31/7/2022), disponible sur : <https://blog.euromaster.fr/lexique-auto/gps>

[20] Jeremy Cohen, fusion de capteurs, medium, (consulté le 10/8/2022), disponible sur : <https://medium.com/france-school-of-ai/la-fusion-de-capteurs-587f91a1423a/>

- [20] Jeremy Cohen, IA, medium, (consulté le 18/8/2022), disponible sur :
<https://medium.com/france-school-of-ai/>
- [21] Effet doppler, alloprof, (consulté le 30/8/2022), disponible sur :
<https://www.alloprof.qc.ca/fr/eleves/bv/sciences/l-effet-doppler-s1546>
- [22] Vitesse Radiale, Wikipédia, (consulté le 5/9/2022), disponible sur :
https://fr.wikipedia.org/wiki/Vitesse_radiale
- [23] Arduino IDE, editions-eni, (consulté le 11/9/2022), disponible sur :
<https://www.editionseni.fr/open/mediabook.aspx?idR=45f8874f5e46660935b41d4120bdcb22>
- [24] Elies Guzman, Arduino IDE, Malavida, (consulté le 13/9/2022), disponible sur :
<https://arduino.fr.malavida.com/windows/#gref>
- [25] Arduino IDE, WiX, (consulté le 13/9/2022), disponible sur :
<https://pecquery.wixsite.com/arduino-passion/ide-arduino>
- [26] Arduino IDE, arduino blaise pascal, (consulté le 15/9/2022), disponible sur :
<https://arduino.blaisepascal.fr/presentation/logiciel/>
- [27] Arduino IDE, softonic, (consulté le 16/9/2022), disponible sur :
<https://arduino-ide.fr.softonic.com/?ex=DINS-276.0>
- [28] Proteus, elektronique, (consulté le 12/9/2022), disponible sur :
<http://www.elektronique.fr/logiciels/proteus.php>
- [29] Batterie 18650, Fenixlightning, (consulté le 12/9/2022), disponible sur :
<https://www.fenixlighting.com/blogs/news/the-ultimate-guide-to-the-18650-battery>
- [30] Servomoteur, Blaisepascal, (consulté le 13/9/2022), disponible sur :
<https://arduino.blaisepascal.fr/controle-dun-servomoteur/>