

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Abderrahmane Mira

Faculté de la Technologie



Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : automatique

Spécialité : automatique et système

Thème

Etude et réalisation d'une mini voiture à base d'Arduino uno

Préparé par :

Boukhezar said

Oubellout nabil

Dirigé par :

Encadreur : M.Hadji Slimane

Co-encadreur : Mme.Saadi Hadjer

Examiné par :

M. Belkaid

Mme.mezzah

Année universitaire : 2021/2022



Remerciement

Nous remercions dieu pour le courage, la patience et la santé qui ont été utile au long de notre parcours.

Nous tenant à dresser notre vif remerciement à notre promoteurs Mm Saadi et Mr hadji qui ont dirigé ce projet et pour son entière disponibilité, son aide et ces précieux conseils.

Notre sincère gratitude va aussi aux membres de jury qui ont bien accepté de juger ce présent travail..

Il est particulièrement agréable d'exprimer ici notre reconnaissance envers tous ceux qui ont rendu possible ce travail.

Que tous ceux qui ont contribué de près ou de loin à notre formation trouvent ici notre sincères remerciement.



DEDICACES

Nous dédions ce modeste travail :

A nos parents qui ont espérés de voir ce jour.

A nos frères et sœurs.

A toutes nos familles.

A nos amis.

*A tous nos collègues de la promotion Automatique et système
2021/2022*

*A toutes les personnes qui nous ont aidés à réaliser
ce mémoire*

Nabil et Said



DEDICACES

Nous dédions ce modeste travail :

A nos parents qui ont espérés de voir ce jour.

A nos frères et sœurs.

A toutes nos familles.

A nos amis.

*A tous nos collègues de la promotion Automatique et système
2021/2022.*

*A toutes les personnes qui nous ont aidés à réaliser
ce mémoire*

sommaire

Chapitre 1 : généralités sur les voiture autonome

I.1. introduction.....	02
I.2. historique.....	02
I.3.définition de la voiture autonome.....	03
I.4.principe de fonctionnement.....	04
I.5.système embarqués.....	05
I.5.1.définition.....	05
I.5.2.pourquoi le système embarqués dans le véhicule autonome.....	06
I.6.niveau d'autonomisation.....	06
I.7.avantages.....	08
I.8.inconvénients.....	09
I.9.conclusion.....	10

Chapitre II :description de la plateforme de développement Arduino

II. introduction.....	11
II.1.définition de module Arduino.....	11
II.1.1.différents types d'arduino.....	11
II.2.1.Arduino uno.....	12
II.1.2.1.caractéristiques de la carte Arduino.....	13
II.1.2.2 constitution de la carte Arduino uno.....	14
II.1.2.2.1.le microcontrôleur Atmega 328.....	14
II.1.2.2.2.les sources d'alimentation de la carte arduino uno.....	15
II.1.2.2.3. les 14 entrées / sortie numérique	15
II.1.2.2.4.les ports de communication.....	15
II.1.2.3.Application de la carte Arduino UNO	16

II.1.2.4avantages de la carte Arduino uno.....	16
II.1.2.5inconconvénients de la carte Arduino uno.....	16
II.1.3.Driver moteur L 298N.....	17
II.1.3.1.alimentation.....	17
II.1.3.2.principe de fonctionnement	18
II.1.4.capteur HC-SR04 a ultrason.....	18
II.1.4.1.principe de fonctionnement du capteur à ultrason.....	18
II.5.1.capteur LDR.....	19
II.5.1principe de fonctionnement de capteur LDR.....	19
II.1.6 Nodemcu8266.....	20
II.1.6.1.caractéristique.....	20
II.1.7.servomoteur.....	21
II.1.7.1.principe de fonctionnement d'un servomoteur.....	21
II.1.8. Batterie.....	22
II.1.8.1.caractéristique.....	22
II.1.8.2.avantage.....	23
II.1.9.motoréducteur	23
II.1.9.1.spécification.....	23
II.2.partie logicielle.....	24
II.3.1.environnement de programmation des carte Arduino (IDE Arduino).....	24
II.2.1. injection de programme dans la carte arduino uno.....	24
II.2.2.structure d'un programme arduino.....	25
II.2.3.instruction de base de programme.....	26
II.4.conclusion.....	26

Chapitre III :Implémentation et résultat des test de la voiture autonome

III.1.introduction.....	27
III.2.1.schéma synoptique de la réalisation.....	27
III.2.2.fonctionnement général de notre voiture autonome.....	28
III.2.3.branchement des composant électronique à la carte Arduino.....	29
III.2.4.liaison carte Arduino et nodemcu.....	30
III.2.5.calculé courant pour quelque composant électronique.....	31
III.3.programmation des carte électronique de la voiture autonome.....	31
III.3.1.logigramme de fonctionnement du système complet réaliser	31
III.3.2.déroulement du programme de la carte Arduino.....	33
III.3.3.interface commande via wifi	38
III.3.4.programme implanté sur nodemcu8266.....	39
III.4.étalonnage du capteur de lumière LDR.....	41
III.5. Étalonnage du capteur ultrason :.....	42
III.6.essai et résultat des différent manipulations.....	44
III.6.1.actionneur le test de la voiture activé	45
III.6.2.teste de voiture en présence d'obstacle.....	45
III.6.3.test déviation de la voiture vers la droit par évitement d'obstacle.....	45
III.6.4.test de déviation de la voiture vers la gauche par évitement d'obstacle.....	46
III.6.5. Test de marche de la voiture dans un milieu obscur	46
III.7.conclusion.....	46

Liste des figures

Figure I .1 : Equipement de la voiture autonome.....	05
Figure I.2 : Niveaux d'automatisation dans les voitures autonomes.....	08
Figure II.1 : Vue d'ensemble et constitution de la carte Arduino.....	14
Figure II.2 : Microcontrôleur Atmega328	14
Figure II.3 : driver moteurL298N.....	17
Figure II.4 : capteur HC-SR04 à ultrason.....	18
Figure II.5 : capteur LDR.....	19
Figure II.6 : Nodemcu8266.....	20
Figure II.7 : servomoteur.....	21
Figure II.8 : batterie 10.8 V.....	22
Figure II.9 : motoréducteur.....	23
Figure II.10 : présentation des parties principales du logiciel.....	24
Figure II.11 : sélection du type de la carte et le port USB.....	25
Figure III.1 : Schéma synoptique de notre système réalisé.....	26
Figure III.2 : branchement des composants électronique de la carte Arduino.....	28
Figure III.3 : branchement détaillée des capteur et actionneur avec la carte Arduino uno...	29
Figure III.4 : liaison carte Arduino et Nodemcu.....	39
Figure III.5 : branchement de la LED.....	30
Figure III.6 : logigramme du system Complete réaliser.....	31
Figure III.7 : interface de commande via wifi.....	37
Figure III.8 : courbe caractéristique de LDR.....	40
Figure III.9 : montage d'étalonnage du capteur LDR	40
Figure III.10 : étalonnage du capteur ultrason	41
Figure III.11 : voiture active	43
Figure III.12 : voiture en présence d'obstacle.....	44
Figure III.13 : déviation de la voiture vers la droite pour éviter un obstacle.....	44
Figure III.14 : déviation de la voiture vers la gauche.....	44
Figure III.15 : marche dans un milieu obscur.....	45

LISTE DES TABLEAUX

Tableau III.1 : Résultat de mesure d'étalonnage du capteur LDR.....	41
Tableau III.2 : Mesures d'étalonnage du capteur ultrason.....	42

Introduction générale

Les gens ont toujours voulu être libérés des travaux lourds, fastidieux et sans intérêt.

Les avancées technologiques telles que nous les connaissons depuis un demi-siècle répondent même bien à ces souhaits avec deux outils généraux : les ordinateurs pour le travail mental et les robots pour le travail manuel. Le premier était quelque peu en avance sur le second grâce à la découverte des semi-conducteurs, des matériaux solides capables de trier les informations sans mobilité mécanique. Le second s'appuie encore sur des mécanismes anciens qui ont des pièces fragiles et mobiles et qui coûtent cher.

Malgré son apparition récente, la robotique trouve son origine dans les civilisations les plus anciennes. Il est même peut-être vrai que toutes les autres sciences n'existent que pour permettre à la robotique de se développer pour aider l'humanité à créer cet « esclave » qui appliquera ses ordres à ses doigts et à ses yeux et il la libérera.

De l'expansion technologique où l'homme préhistorique s'est empressé d'inventer des membres pour faciliter son travail, comme des massues, jusqu'aux robots mobiles autonomes les plus performants, les robots mobiles n'ont cessé de connaître des avancées étonnantes. En effet, des générations de robots mobiles se sont succédé à chaque fois, offrant plus de performances en termes de perception, d'autonomie et de capacités de prise de décision. Ces progrès sont le résultat inévitable du développement de la mécanique, de la microélectronique et de l'informatique. C'est pourquoi en très peu de temps, génération après génération, des robots se sont succédé, voyant naître une branche de la robotique visant un champ de vision plus large : c'est l'ère des robots mobiles.

L'objectif principal de notre travail est de réaliser une mini voiture autonome à base d'une Arduino UNO, la tâche principale de la voiture est d'éviter les obstacles.

Ce travail est alors organisé en trois chapitres :

- Le premier chapitre présente des généralités concernant les voitures autonomes réelles,
- Le second chapitre introduit quelques composants de notre projet comme les capteurs les actionneurs, le langage de programmation et en particulier IDE Arduino,
- Le troisième chapitre et le dernier décrit les étapes de réalisation de notre voiture autonome et quelques résultats de tests de mise en œuvre de cette dernière.

I.1. Introduction

Le concept de la voiture autonome désigne un véhicule capable de rouler sans conducteur sur une route ouverte, ce type d'automobile doit pouvoir connaître son emplacement très précisément pour pouvoir localiser sa position exacte. Il est présenté comme le futur de la mobilité motorisée.

L'objectif de ce chapitre est d'avoir une idée globale sur la conception et le fonctionnement de la voiture autonome réelle.

I.2. Historique

Le terme voiture autonome n'est pas nouveau. Plusieurs tentatives avaient été réalisées depuis les années 1970, et avec l'évolution des technologies liées aux capteurs, à la télématique et à la puissance intrinsèque des processeurs numériques, ont abouti à de bons résultats et ce domaine a connu une bonne évolution technologique.

En 1977 au Japon dans le laboratoire de Tsukuba Le premier véhicule sans conducteur fut développé, ce dernier se déplaçait à la vitesse de 30 km/h et suivait un marquage afin de ne pas sortir du circuit. Du fait qu'il ne pouvait rouler sans ce marquage on ne peut pas réussir à classer ce premier véhicule selon les différents niveaux présentés ci-après [1].

En 1984 Mercedes-Benz testa une camionnette automatique équipée de caméras, dont le logiciel de reconnaissance était développé par une équipe de l'université de la Bundeswehr à Munich sous la direction d'Ernst Dickmanns. Le véhicule atteint 100 km/h sur un réseau routier sans trafic.

En 1986 le projet ALV (Autonomous Land Vehicle) financé par l'Agence pour les projets de recherche avancée de défense (DARPA), aboutit à un démonstrateur autonome capable de suivre une route à 30 km/h. Le laboratoire de Robotique de l'université Carnegie-Mellon de Pittsburgh démarre le développement des véhicules automatiques Navlab.

En 1987 la Commission européenne finança le programme européen Prometheus, à hauteur de 800 millions d'euros, qui contribua, entre autres, au développement d'outils technologiques dédiés à la conduite automobile automatique. La même année, le laboratoire de recherche Hughes Aircraft (HRL) complète le véhicule ALV en le rendant tout-terrain et capable d'évoluer à 3 km/h dans un environnement complexe (végétation, rochers, ravins).

En 1994, à l'occasion d'une conférence scientifique sur le thème, Daimler-Benz réalise une démonstration, en situation réelle de trafic sur l'autoroute A1 à partir de Paris, de deux

véhicules autonomes (VaMP and Vita-2) pilotés par logiciels de l'équipe de Dickmanns et capables de réaliser conduite en file, changement de file et dépassement avec une vitesse de pointe de 130 km/h.

En 1995 un de ces véhicules réalisa le trajet Munich-Copenhague et retour (1 600 km) avec une vitesse atteinte de 175 km/h. La plus longue section de conduite automatique continue fut 158 km. La même année un véhicule Navlab réalisa l'opération No Hands Acrosses American sur le trajet de Washington, D.C. à San Diego.

En août 1997 a lieu à San Diego une importante démonstration du « consortium américain de l'autoroute automatisée » National Automated Highway System Consortium (NAHSC) où divers véhicules autonomes peuvent être comparés. À cette occasion une infrastructure spécifique avait été préparée par l'insertion de plots magnétiques servant au guidage dans certaines sections d'autoroute[2].

Aujourd'hui les entreprises se lancent dans la course à la meilleure voiture autonome. Comme le montre le graphique ci-dessous, les entreprises américaines sont les plus avancées dans ce domaine.

I.3. Définition de la voiture autonome

La notion de voiture autonome ou automatisée, désigne un véhicule capable de rouler sans conducteur sur la route ouvert. On parle plus rarement d'automobile sans pilote, de voiture automate, de véhicule intelligent sans conducteur, etc.

Il existe deux types de voiture autonome : self-contained et interconnected. Le premier type, self-contained ou indépendant, s'appuie uniquement sur les informations générées par le véhicule lui-même. La voiture est équipée d'une série de technologies qui recueillent les informations et donnent une connaissance de l'environnement à 360°. Cela permet à la voiture de réagir de manière proactive aux obstacles.

Le second type, interconnected ou interconnecté, implique que la voiture est connectée à un ou plusieurs réseaux de communication. La voiture reçoit des informations sur l'infrastructure, les autres véhicules, le trafic, la météo, des données GPS. Elle est donc contrôlée par le réseau. Nous appelons les voitures de cette deuxième catégorie les voitures autonomes. Cependant, vu l'interdépendance de ces voitures entre elles et avec l'infrastructure, il serait plus juste de les appeler voitures automatisées plutôt qu'autonomes.

Dans ce travail, le terme autonomie se comprend comme l'indépendance de la voiture par rapport à son passager [3].

I.4. Principe de fonctionnement

Les véhicules autonomes peuvent détecter des obstacles et s'arrêter si nécessaire à l'aide d'une technologie sophistiquée les véhicules de navia ouais mots et locales motors sont en mesure d'embarquer les passagers a la demande et les déposer à leur destination souhaitée ,ces véhicules sont capables de reconnaître les feux de circulation et les panneaux de signalisation aux intersections et d'agir en conséquence.

Les voiture autonomes n'ont pas besoin d'un conducteur pour se rendre d'un point A à B, c'est la technologie qui est le conducteur elle comprend de nombreux capteurs tels que les capteurs lidar les capteurs à ultrasons, les caméras et les capteurs radar.il sont situés à l'avant et à l'arrière du véhicule, les caméras et les capteurs l'eider se trouvent aussi fréquemment sur le toit, sans eux le véhicule serait pratiquement aveugle.

Le lidar est un élément essentiel dans de nombreux véhicule, il fonctionne de la même manière que le radar cependant il n'utilise pas d'onde radio mais de nombreux rayons lumineux qui sont réfléchies par les objets dans l'environnement et détectés par le capteur lidar ,de cette façon un nuage de points est généré à partir de l'environnement ; cependant, il ne reconnaît pas les panneaux de signalisation routière, les caméras utilisent l'intelligence artificielle pour reconnaître et interpréter les panneaux de signalisation ,elle sont aussi utilisées pour détecter des obstacles et sont nécessaires pour le système de maintien de la voix, par contre en cas de pluie ou de bouillard le leader et les caméras sont absolument incapables de fonctionner dans cette situation ; le radar est nécessaire pour éviter les collisions, un inconvénient majeur du radar, cependant, les objets doivent être suffisamment grands pour être détectés, il se peut donc que les piétons ou les cyclistes ne soient pas reconnues par le radar, la position exacte du véhicule est déterminé par GPS et des cartes numérique qui contiennent beaucoup d'informations y compris des informations telles que les panneaux d'intersection et les limitations de vitesse et comme le GPS ne peut pas toujours être garantie, par exemple dans les tunnels et les canyon de rue (l'indomitri)est également nécessaire. Par exemple le diamètre et la vitesse des roues peuvent être utilisés pour estimer la position future du véhicule.

Les données de tous les capteurs sont ensuite combinées et traitées dans un ou plusieurs processeurs NVIDIA,AMD et d'autres fabricants qui proposent de telles puces pour une conduite autonome. La figure I.1 illustre les équipements d'une voiture autonome.

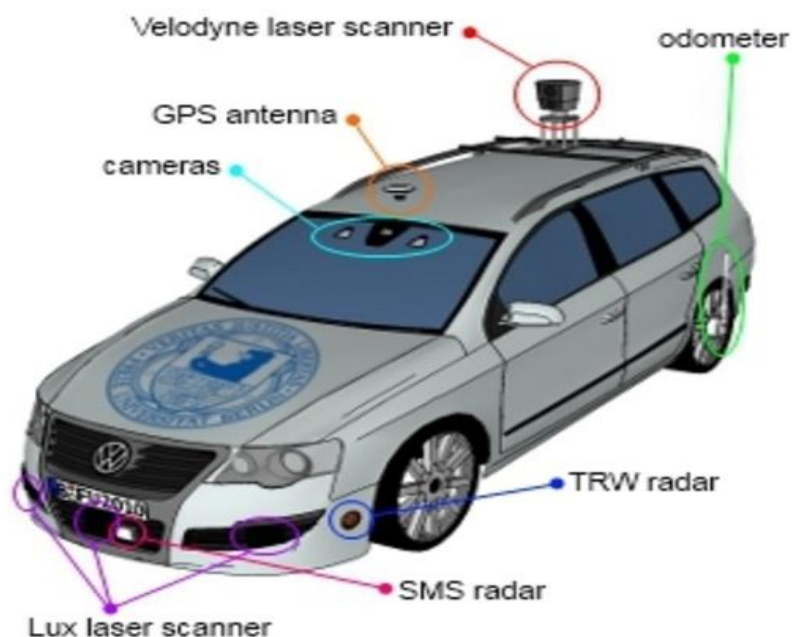


Figure I.1 : Equipements de la voiture autonome

I.5. Système embarqués

Les voitures autonomes ont besoin d'une technologie développée, cela veut dire l'intégration de l'électronique et l'information afin d'avoir des fonctionnalités intelligentes et c'est ce que on appelle « les systèmes embarqués »

I.5.1 Définition

Un système embarqué est un ensemble d'éléments informatiques et électroniques interagissant entre eux de façon autonome et complémentaire. Ces systèmes sont conçus de manière à pouvoir répondre spécifiquement aux besoins de leur environnement respectif. Le terme « système » désigne l'ensemble des éléments qui constituent le système embarqué, souvent ces systèmes sont composés de sous-systèmes étant donné leur complexité. Le terme « embarqué » représente la mobilité et l'autonomie du système en interaction directe avec son environnement dans l'exécution de tâches précises, afin de répondre à la finalité de celui-ci.

Contrairement aux systèmes classiques, les systèmes embarqués sont conçus pour réaliser des tâches bien précises. Certains doivent répondre à des contraintes de temps réel pour des raisons de fiabilité et de sécurité, indispensables selon l'utilisation du système. Un système embarqué regroupe à la fois la partie software (logicielle) et la partie hardware (matériaux) étroitement liées afin de produire les résultats escomptés [8].

I.5.2. Pourquoi le système embarqué dans le véhicule autonome ?

Aujourd'hui, un véhicule contient une grande quantité d'électronique et d'informatique : on retrouve plus de 100 capteurs, 30 à 50 calculateurs selon le type de véhicule et parfois près d'un million de lignes de codes pour les véhicules de dernière génération. Cette évolution s'explique par les demandes exigeantes des consommateurs et l'envie de différenciation des concurrents sur le marché de l'automobile. S'ajoute à cela les contraintes économiques et écologiques où l'électronique embarquée répond à ces nouvelles attentes. De nouvelles fonctionnalités impliquent parfois une intégration électronique et informatique par le biais de systèmes embarqués [8].

Il existe plusieurs degrés d'autonomie des voitures automatisées. Certaines d'entre elles se contentent d'assister le conducteur, tandis que d'autres sont pleinement autonomes.

I.6. Niveaux d'autonomisation

□ Niveau 0

Le conducteur est maître du véhicule à tout moment. Il contrôle les fonctions primaires telles que les freins, le volant, l'accélérateur et il reste le seul responsable du comportement de la voiture. Il doit surveiller la route et manœuvrer le véhicule de manière sûre. Les véhicules équipés de systèmes d'alerte mais sans autorité sur les fonctions primaires sont considérés comme appartenant au niveau 0 [4]. Les voitures de niveau 0 sont équipées d'allumage de phares automatique ou de capteurs de recul, par exemple.

□ Niveau 1

Dans ce niveau, l'automatisation ne concerne qu'une ou plusieurs fonctions de contrôle spécifiques et même si plusieurs fonctions sont automatisées, elles ne sont pas interconnectées. Le conducteur garde le contrôle général de son véhicule et reste responsable du fonctionnement sécuritaire mais il peut décider de céder partiellement le contrôle d'une fonction principale. La voiture va à ce moment-là prendre le contrôle d'une fonction ou apporter un support/contrôle supplémentaire au conducteur. Mais en aucun cas, le système ne remplace la supervision de l'être humain ni ne devient responsable du comportement du véhicule. Le système peut assister le conducteur dans l'exercice d'une des fonctions primaires, soit la direction soit le freinage, mais pas les deux ! Le véhicule ne pouvant pas contrôler différentes fonctions en même temps, le conducteur ne peut pas se détourner physiquement de la conduite en lâchant le volant et en retirant les pieds des pédales. On peut retrouver, par exemple, un régulateur de vitesse, un système de freinage automatique ou encore un système de maintien sur une bande dans des voitures de niveau 1 [5].

□ Niveau 2

Les véhicules de niveau 2 disposent de fonctions primaires programmées pour agir en parallèle. Le véhicule et le conducteur se partagent l'autorité dans certaines situations. Le conducteur reste responsable du comportement de la voiture et reste disponible pour reprendre le contrôle rapidement car le système peut le lui céder sans l'alerter préalablement. Les voitures du niveau 2 autorisent, par exemple, l'activation du Cruise control auto adaptatif et le maintien sur une bande en même temps lorsque le trafic est dense et que la vitesse ne dépasse pas les 30 km/h. La différence avec les voitures de niveau 1 est que le conducteur peut maintenant se désengager physiquement de la conduite du véhicule en lâchant le volant et en enlevant ses pieds des pédales. La NHTSA précise que le désengagement est physique car le conducteur doit continuellement surveiller la route et reprendre le contrôle lorsque les limitations requises ne sont plus rencontrées. Dans notre exemple, le conducteur reprend le contrôle lorsque la circulation se fluidifie et qu'il peut rouler à plus de 30 km/h[4].

□ Niveau 3

Les véhicules de niveau 3 ont la capacité de contrôler l'ensemble de leurs fonctions sous certaines conditions environnementales et de trafic limitées. C'est au système de la voiture de surveiller les changements de ces conditions qui requièrent la retransmission du contrôle au conducteur. Ce dernier reste disponible pour des reprises de contrôle exceptionnelles, mais dans un délai raisonnable. La spécificité du niveau 3 est que le conducteur ne surveille pas la route en permanence : la voiture détecte à l'avance les éléments perturbateurs à sa conduite autonome et le prévient qu'une reprise de contrôle sera nécessaire. Avec les véhicules de niveau 3, le conducteur entre sa destination et la voiture s'occupe du reste. Elle déterminera l'itinéraire, accélérera, freinera et se dirigera d'elle-même. Ces véhicules savent désormais se diriger dans les carrefours, s'arrêter aux feux de circulation et utiliser les ronds-points.

L'action du conducteur est requise lorsque la voiture approche de zones d'incertitude comme une zone de travaux, par exemple [4].

□ Niveau 4

Ici on est au niveau de conduite automatique totale. En effet, le véhicule contrôle toutes les fonctions, qu'elles soient primaires ou secondaires (comme allumer les phares ou faire fonctionner les essuie-glaces), et surveille les conditions de circulation durant tout le trajet. À ce niveau, le conducteur peut se contenter d'introduire sa destination, il ne doit plus surveiller la route ni se tenir prêt à reprendre le contrôle. Le conducteur ne contrôle donc le véhicule à aucun moment pendant le déplacement. La voiture va jusqu'à choisir elle-même le chemin le

plus rapide en fonction des conditions de circulation et éviter, par exemple, les zones congestionnées. On retrouve au niveau 4 aussi bien des véhicules occupés que des véhicules inoccupés [6].

□ Niveau 5

Il n'y a plus du tout de conducteur, puisque le véhicule peut conduire lui-même pour aller chercher une place de parking ou pour aller chercher des clients (quand il s'agit d'un robot-taxi. A ce niveau de service, l'automatisation est complète, sur tous types de trajets, par tous les temps. La figure I.2 montre les niveaux d'automatisation dans les voitures autonomes.

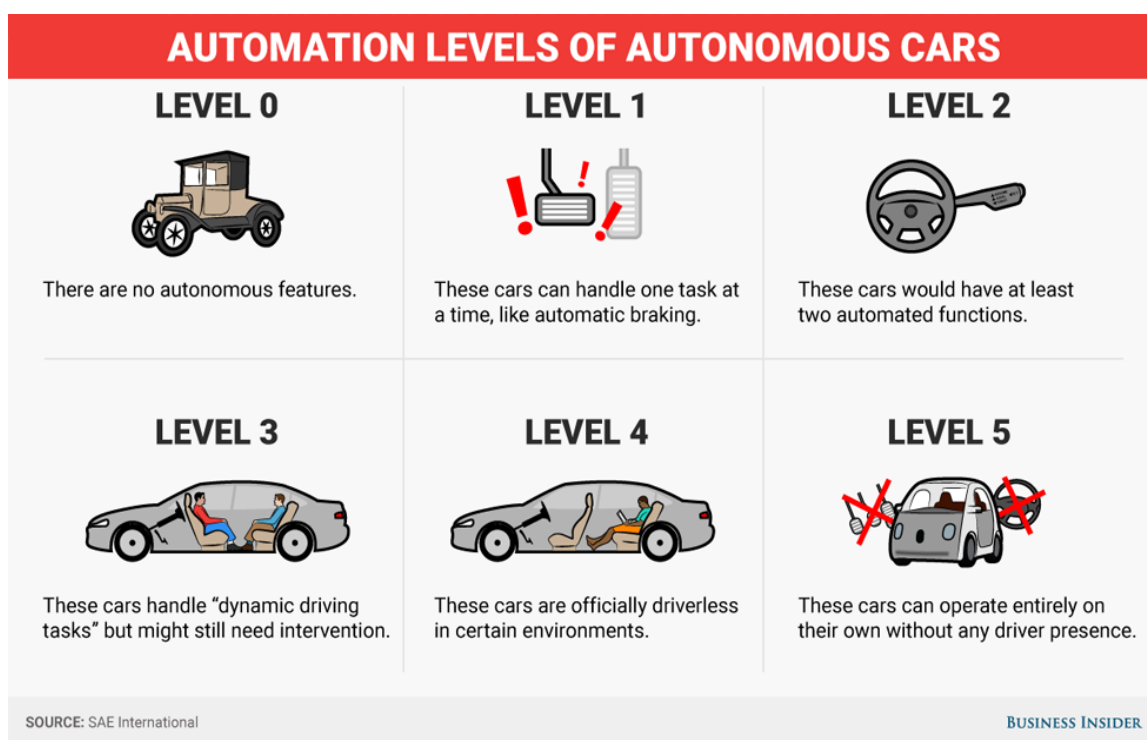


Figure I.2 : Niveaux d'automatisation dans les voitures autonomes

I.7. Avantages

Probablement, l'un des plus grands atouts des voitures autonomes est qu'elles n'ont jamais ressenti de stress et de lassitude dans les embouteillages. Cela n'aura plus de raison d'être. Plus besoin de jouer avec l'embrayage en attendant que la route se libère. Relaxez-vous, prenez un livre, et attendez tranquillement que votre voiture sorte seule du trafic.

1. Limiter le nombre d'accidents de la route

La plupart des accidents de la route sont le fait d'une erreur humaine. Avec les voitures autonomes, leur nombre devrait réduire drastiquement pour une raison toute simple ; elles ne peuvent ni se fatiguer, ni être distraite en conduisant.

2. Se garer ? Un jeu d'enfant !

Selon une estimation, les voitures autonomes pourront libérer près de 15% de places de parking en plus et ce, grâce aux différents capteurs et cameras. Fini de râler face aux mauvais conducteurs qui prennent deux places au lieu d'une !

3. Elles seront plus économiques

Pour un trajet donné, les voitures autonomes savent exactement combien de carburant utiliser. De plus, grâce à leur système électronique intégré, elles sont capables de prévoir les chemins les plus courts et les plus économiques.

4. Permettre aux personnes en situation d'handicap de se déplacer plus facilement

Trop de personnes ont des difficultés à se déplacer. Les transports en commun, bien souvent mal adaptés, n'aident pas à changer cet état de fait. Avec les voitures autonomes, plus besoin de faire appel à l'aide des services publics ou d'un tiers, les personnes en situation d'handicap peuvent bouger sans problèmes.

I.8. Inconvénients

1. Pertes d'emplois

L'autonomisation des taxis, bus et autres camions aura des conséquences lourdes sur les conducteurs et leur emploi. Volvo teste déjà des camions autonomes pour aider les mineurs en Norvège. L'impact sur l'économie est difficilement mesurable, mais une telle révolution des transports ne peut qu'entraîner une crise sociale.

2. La vitesse de décision

Quand nous parlons de vitesse de décision, nous faisons références aux événements annexes et imprévus pouvant survenir sur la route. Par exemple, si un animal apparaît soudainement devant le véhicule, comment la voiture va-t-elle réagir ? Va-t-elle éviter l'animal et foncer sur le bas-côté de la route. Un problème de moral que les ingénieurs informatiques auront bien du mal à résoudre.

3. Les risques de piratages informatiques sont importants

Le fléau du monde moderne, le piratage informatique du système interne de la voiture compromet la sécurité du véhicule et peut entraîner son contrôle à distance. De plus, un hacker peut accéder à vos informations personnelles telles que vos numéros de téléphone et de carte de crédit ou votre lieu de résidence.

4. Comment la voiture autonome va-t-elle réagir en cas de météo désastreuse ?

Les intempéries telles que la neige fondue et les fortes pluies peuvent interférer avec les capteurs et les caméras. Cela peut fausser les informations reçues et entraîner la mise en danger des personnes situées dans l'habitacle. Espérons que cette technologie s'améliorera au cours des années à venir.

5. Qui est responsable en cas d'accident ?

Le statut juridique des voitures autonomes est à définir. En cas d'accident, qui est responsable ? Le constructeur ? L'éditeur de logiciel ? Le « conducteur » ? Impossible pour l'heure de prévoir comment le cas sera traité, car il n'y a pas de jurisprudence sur le sujet [7].

I.9. Conclusion

Dans ce chapitre nous avons traité le fonctionnement et la conception de la voiture autonome réelle ainsi que les différents niveaux d'autonomie pour avoir une idée générale sur l'environnement de conception de notre projet.

Le chapitre suivant sera consacré à la présentation des différents composants de la mini voiture autonome.

Introduction

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi de système embarqué ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit. Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation. Nous allons en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à celle de l'électronique.

II .1.Définition du module Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre.

Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles), pour programmer cette carte, on utilise l'logiciel IDE Arduino [9].

II .1.1. Différents types d'Arduino

Actuellement, il existe plus de 20 versions du module Arduino, nous citons quelques-uns afin d'éclaircir l'évaluation de ce produit scientifique et académique :

- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega.
- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.

- L'Arduino Nano, une petite carte programmée à l'aide port USB, cette version utilise un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Méga, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisation microcontrôleur ATmega328
- L'Arduino Mega2560, utilisation d'un microcontrôleur ATmega2560, et possède toute une mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier [10].

II.1.2.Arduino uno

L'Arduino (uno) est une petite carte avec une électronique programmable. Avec un Arduino, vous pouvez contrôler divers composants et grandeurs électriques. Pensez aux lumières, aux interrupteurs, aux moteurs et aux divers capteurs, elle possède une interface USB pour pouvoir la programmer.

II.1.2.1. Caractéristiques de la carte Arduino

Le tableau II.1 résume les principales caractéristiques de la plateforme de développement Arduino UNO sur laquelle notre choix s'est porté.

Microcontrôleur	Atmeg328
Tension de fonctionnement	5V
Tension d'alimentation	7 - 12V
Tension d'alimentation (limites)	6 - 20V
Broches E/S numériques	14(dont 6disposent d'une sorties PWM)
Broches d'entrées analogiques	6(utilisables en broches E/S numériques)
Intensité maximale disponible par broches E/S(5V)	40mA
Intensité maximale disponible pour la sortie 3.3V	50mA
Intensité maximale disponible pour la sortie 5V	Fonction de l'alimentation utilisée- 500MaMax si port USB utilisé seul
Mémoire programmable Flash	32KB(Atemga328)dont 0.5KB sont utilisée Par bootloader
Mémoire SRAM (mémoire volatile)	2KB(Atemga328)
Mémoire EEPROM(mémoire non volatile)	1 KB (Atmga328)
Vitesse d'horloge	16MHz

II.1.2.2. Constitution de la carte Arduino UNO

La figure II.1 représente une vue d'ensemble de la carte Arduino UNO avec les différentes parties l'a constituant.

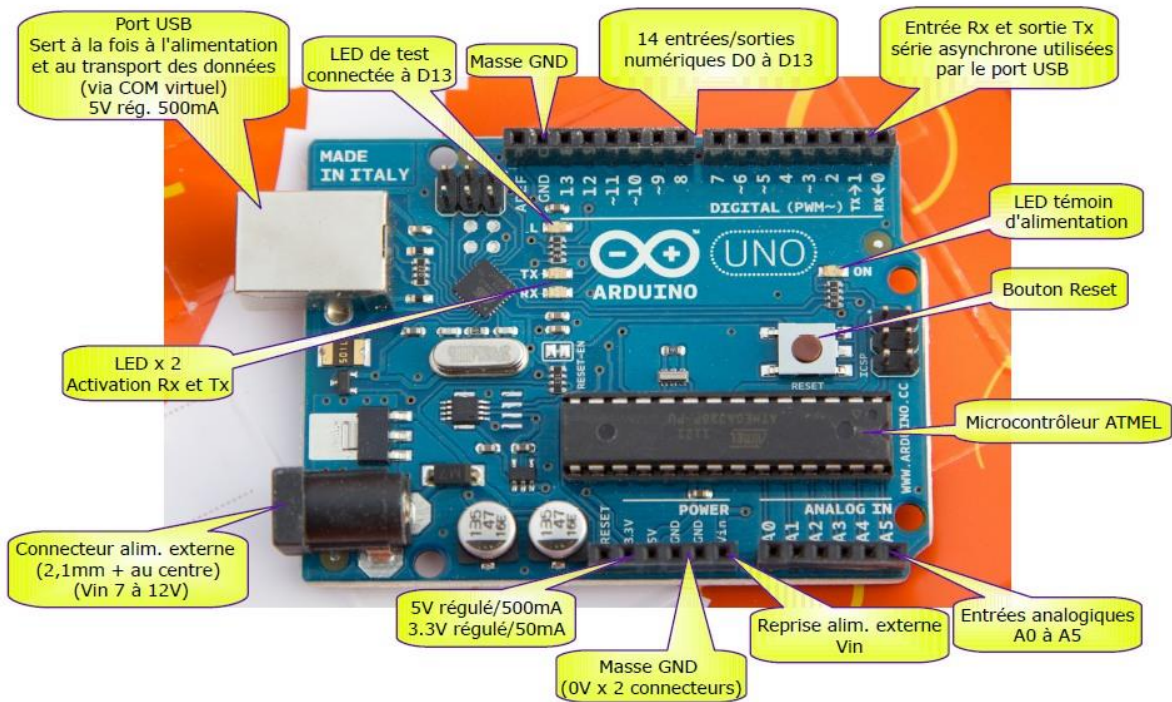


Figure II.1. Vue d'ensemble et constitution de la carte Arduino

II.1.2.2.1. Le microcontrôleurAtmega 328

Microcontrôleur (μ c, uc, ou encore MCU en anglais) est un circuit intégré et compact, conçu pour régir une opération spécifique et dans un système intégré. Il comprend un processeur, une mémoire et des périphériques d'entrée et de sortie sur une seule carte ou une seule puce[2].

La figure II.2 montre une photo du microcontrôleur en question.



Figure II.2. Microcontrôleur ATmega328

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur.

II.1.2.2.2. Les sources d'alimentation de la carte Arduino UNO

- **VIN**. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- **3V3** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50 mA.
- **GND** : Broche de masse (ou 0V)[11].

II.1.2.2.3. Les 14 entrées/sorties numériques

Ces broches peuvent être configurées pour fonctionner comme des broches numériques d'entrée pour lire des valeurs logiques (0 ou 1) ou numériques. Elles fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité.

II.1.2.2.4. Les ports de communication

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

II.1.2.3 Application de la carte Arduino

Arduino peut être utilisé pour développer des applications matérielles industrielles légères ou des objets interactifs. Il peut recevoir en entrée une très grande variété de capteurs, il peut contrôler une grande variété d'actionneurs, et aussi en électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle.

II.1.2.4. Avantages de la carte Arduino UNO

Il existe plusieurs modèles de carte ARDUINO, mais le modèle UNO est le plus répandu et vous permet déjà un large éventail de possibilités et ses nombreux avantages que nous pouvons résumer comme suit :

- Le prix. ...
- La communauté. ...
- La simplicité. ...
- Le multiplateforme. ...
- Les « shields ». ...
- Aucune limite

II .1.2.5. Inconvénients de la carte Arduino

- Nombre de broches insuffisant pour les grands projets,
- La mémoire disponible risque d'être un peu juste pour les projets ambitieux,
- Ne peut pas être utilisée comme hôte USB pour simuler un périphérique tel qu'un clavier ou une souris.

II.1.3. Driver moteur L298N

La puce L298N est un circuit intégré de chez ST Microélectronique. Elle contient essentiellement de deux ponts en H, permettant de piloter chacun un moteur électrique DC (dans un sens, ou dans l'autre), et d'une logique de commande à « faible courant » pour piloter ces ponts à « fort courant ». La figure II.3 montre une photo du driver moteur L298N.

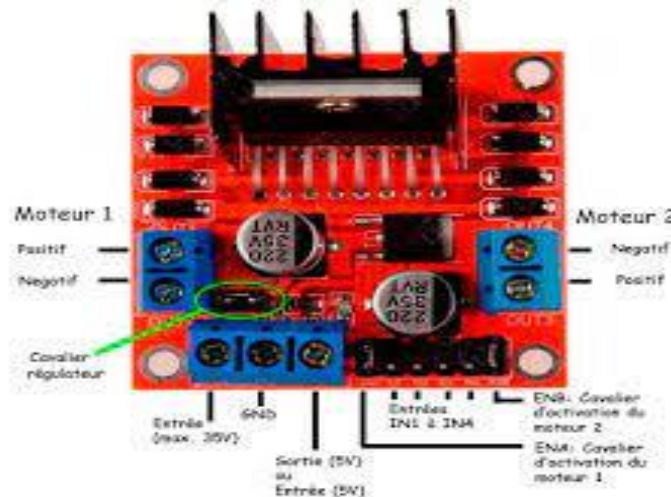


Figure II.3. Driver moteur L298N

II.1.3.1 Alimentation

- **Alimentation « partie puissance » (broche Vs)**

Comme vu dans le paragraphe précédent, **le L298N nécessite deux alimentations pour fonctionner : une pour la partie puissance, et une autre pour la partie commande.** Ici, nous allons nous intéresser à l'alim de « puissance », c'est-à-dire celle qui alimentera les moteurs. Celle-ci est notée « Vs » sur la documentation du constructeur.

Sa valeur pratique est en fait à déterminer en fonction des moteurs que vous comptez alimenter. Par exemple : si vous cherchez à alimenter un moteur DC 12V, il faudra que cette alimentation soit d'au moins 14 volts. Car **il faut prendre en compte la chute de tension engendrée par les ponts en H du L298N.** À ce sujet, il faut savoir que la chute de tension de ces ponts en H peut aller de 1,8 à 4,9 volts, dans le pire des cas (si vous essayez de tirer l'intensité maximale). Donc, il faut faire attention à ne pas sous-alimenter vos moteurs.

- **Alimentation « partie commande » (broche Vss)**

C'est la **seconde alimentation à fournir au L298N, pour alimenter non pas les moteurs cette fois-ci, mais plutôt la logique de commande, interne à ce circuit intégré.** Car les transistors de puissances sont commandés via un dispositif logique interne nécessitant une

alimentation plus basse. Cette tension est notée « Vss » par le fabricant. La tension de commande Vss devra être comprise entre 4,5 et 7 volts (et typiquement : 5 volts).

En pratique, on évitera d'alimenter en permanence le L298N en 7 volts, pour ne pas écourter sa durée de vie. Car c'est ici sa tension maximale admissible (« Absolute Rating »), qu'il ne faut pas maintenir longtemps. **On préférera donc une alimentation stabilisée à 5V**, plutôt qu'autre chose [12].

II.1.3.2. Principe de fonctionnement

Le module L298N : est basé sur le circuit intégré du même nom, le circuit L298N est un double pont en H permettant de modifier la direction et l'intensité de la tension au borne de deux charge électriques. Les caractéristiques du module L298N sont:

- Pilotage de moteur de tension nominale de 5 à 35V,
- 2A courant maximum (peak),
- Tension 5V acceptées sur les broches Input.

II.1.4. Capteur HC-SR04à ultrason

Un capteur de distance à ultrason utilise le même principe qu'un capteur laser, mais en utilisant des ondes sonores (inaudible) au lieu d'un faisceau de lumière. Ils sont bien moins chers qu'un capteur laser, mais aussi bien moins précis. Cependant, contrairement aux capteurs à infrarouge, la lumière ambiante et l'opacité de la surface en face du capteur ne joue pas sur la mesure. La figure II.4 montre une photo du capteur à ultrason en question.



Figure II.4 : Capteur HC-SR04à ultrason

II.1.4.1. Principe de fonctionnement du capteur a ultrason

1. On envoie une impulsion HIGH de 10µs sur la broche TRIGGER du capteur.
2. Le capteur envoie alors une série de 8 impulsions ultrasoniques à 40KHz (inaudible pour l'être humain).
3. Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retourne dans l'autre sens vers le capteur.
4. Le capteur détecte l'écho et clôture la prise de mesure.

5. Le signal sur la broche ECHO du capteur reste à HIGH durant les étapes 3 et 4, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance.

II .1.5. Capteur LDR

Une photorésistance est un composant électronique dont la résistivité varie en fonction de la quantité de lumière incidente. On peut également la nommer *résistance photo-dépendante* (light-dependent resistor (**LDR**)). La figure II.5 montre une photo du capteur LDR.

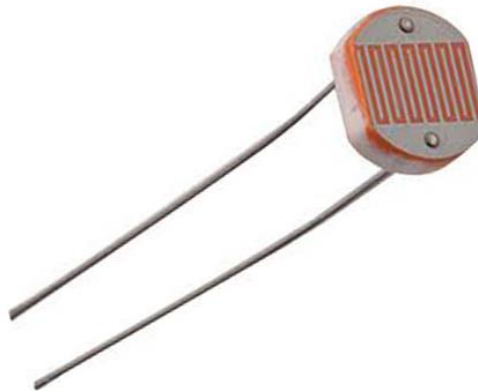


Figure II.5 : Capteur LDR

II .1.5.1. Principe de fonctionnement du capteur LDR

Une photorésistance est composée d'un semi-conducteur à haute résistivité. Si la lumière incidente est de fréquence suffisamment élevée, elle transporte une énergie importante. Au-delà d'un certain niveau propre au matériau, les photons absorbés par le semi-conducteur donneront aux électrons liés assez d'énergie pour passer de la bande de valence à la bande de conduction.

La compréhension de ce phénomène entre dans le cadre de la théorie des bandes. Les électrons libres et les trous d'électron ainsi produits abaissent la résistance du matériau [13].

II.1.6. Nodemcu8266

L'ESP8266 : est un circuit intégré à microcontrôleur avec connexion Wifi développé par le fabricant chinois Espressif. La figure II.6 montre une photo du module en question.

En août 2014, les passionnés d'électronique commencent à s'intéresser à la version ESP-01 de ce circuit intégré produite par une entreprise tierce, AI-Thinker. Celui-ci, de taille réduite, permet de connecter un microcontrôleur à un réseau Wifi et d'établir des connexions TCP/IP avec des commandes Hayes. La traduction progressive de la documentation (à l'origine uniquement en chinois) motivée par le bas prix du circuit a permis le développement d'une communauté de développeurs et de passionnés de l'ESP8266.

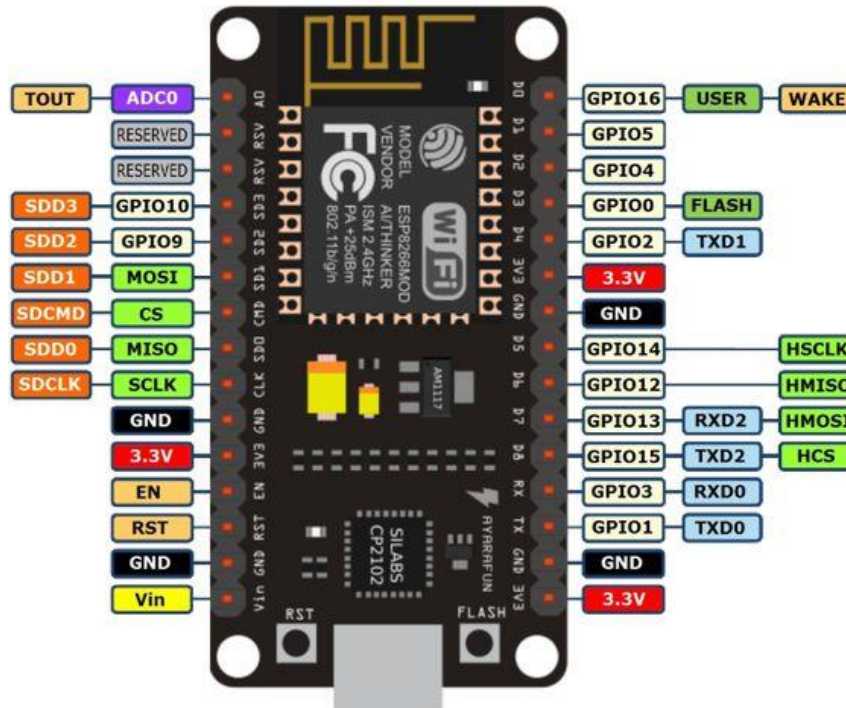


Figure II.6 : Nodemcu8266

II.1.6.1.Caractéristique

- 32-bit RISC CPU: TensilicaXtensa LX106, 80 MHz ;
- 64 Kio de RAM instruction, 96 Kio de RAM data ;
- QSPI flash externe - 512 Kio à 4 Mio (supporte jusqu'à 16 Mio) ;
- IEEE 802.11 b/g/n Wi-Fi ;
- TR switch intégré, balun, LNA, amplificateur de puissance et matching network ;
- Authentification par WEP ou WPA/WPA2 ou bien réseau ouvert ;
- Certaines variantes supportent une antenne externe ;
- 16 broches GPIO;
- Interfaces SPI, I2C ;
- Interface I2S avec DMA (partageant les broches avec les GPIO) ;

- UART sur des broches dédiées, plus un UART dédié aux transmissions pouvant être géré par GPIO2 ;
- Un convertisseur ADC de 10-bits.

Les variantes ont toutes des spécificités. Certaines auront plus de broches de GPIO accessibles (entre 2 seulement pour l'esp8266 de base, et 11 pour un ESP12), d'autres auront plus de mémoire flash disponible (entre 512K, 1MB14 or 4MB suivant les versions choisies).[14]

II.1.7. Servomoteur

Les [servomoteurs](#) sont des moteurs un peu particuliers, qui peuvent tourner avec une liberté d'environ 180° et garder de manière relativement précise l'angle de rotation que l'on souhaite obtenir. On utilise des servomoteurs couramment en modélisme pour contrôler des systèmes mécaniques (gouverne d'avion, accélérateur de moteur thermique, etc.). Les servomoteurs sont aussi couramment utilisés en robotique pour faire des minirobots, des actionneurs ou des indicateurs rotatifs. Voir la figure II.7 d'un servomoteur.

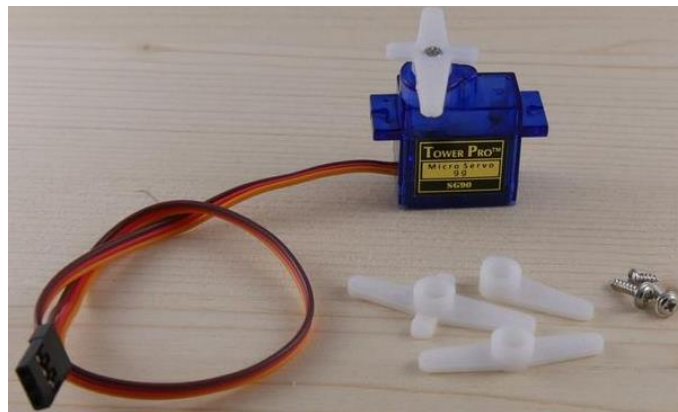


Figure II.7 : Servomoteur

II.1.7.1. Principe de fonctionnement d'un servomoteur

Le fonctionnement interne d'un servomoteur est assez basique. Un petit circuit électronique permet de contrôler un moteur à courant continu en fonction de la position d'un potentiomètre intégré au servomoteur.

La sortie du moteur à courant continu est reliée mécaniquement à une série d'engrenages qui augmente la force (le couple) du servomoteur en réduisant la vitesse de rotation de celui-ci.

Quand le moteur tourne, les engrenages s'animent, le bras bouge et entraîne avec lui le potentiomètre. Le circuit électronique ajuste continuellement la vitesse du moteur pour que le potentiomètre (et par extension le bras) reste toujours au même endroit.

Il suffit de donner une consigne au servomoteur ("reste à 45°" par exemple) et le servomoteur fera son maximum pour rester au plus près de cette consigne. Cette consigne est transmise au moyen d'un signal numérique, d'une impulsion pour être précis.

Pour que le servomoteur reste à une position donnée, il faut transmettre toutes les 20 millisecondes (soit à une fréquence de 50Hz) une impulsion d'une longueur comprise entre 1 et 2 millisecondes.

- Une impulsion de 1 milliseconde correspond à un angle de 0°.
- Une impulsion de 2 millisecondes correspond à un angle de 180°.

II.1.8. Batterie

Une batterie est un appareil qui stocke et accumule l'énergie électrique pour la restituer. Tout comme la pile, la batterie est composé de deux électrodes – un pôle positif (ou cathode) et un pôle négatif (ou anode) – plongées dans une solution, appelée électrolyte, faisant office de conducteur. La figure II.8 montre une photo de la batterie utilisée dans ce bloc.



Figure II.8 :Batterie10.8V

II.1.8.1. Caractéristiques

- Capacité de la batterie : 2500 mAH ;
- Tension de la batterie : 10.8 V ;
- Pile de la batterie : deux lithium-ion nécessite des piles ;
- Type de batterie : li-ion.

II.1.8.2. Avantage

- Utilise des cellules LG ou SAMSUNG de la plus haute qualité, offrant d'excellentes performances.
- Protection contre la surcharge et la décharge pour augmenter la durée de vie de la batterie.
- Aucun effet de mémoire, ne peut être chargé ou déchargé au besoin sans perte de capacité.
- Faible autodécharge pour une productive maximale et moins de temps d'arrêt.

II .1.9. Motoréducteur

Le motoréducteur est un appareil composé d'un moteur électrique monophasé ou triphasé et d'un réducteur. Le principe de celui-ci est de réduire la vitesse de sortie grâce à un système de pignon, tout en augmentant le couple. Il simplifie la transmission poulie / courroie. Une photo du motoréducteur est représentée par la figure II.9.



Figure II.9 : Motoréducteur

II .1.9.1. Spécifications

- Vitesse hors-charge (3V) : 100 tours/min ;
- Vitesse hors-charge (6V) : 200 tours/min ;
- Courant hors-charge (3V) : 60mA ;
- Courant hors-charge (6V) : 71mA ;
- Courant d'arrêt (3V) : 260mA ;
- Courant d'arrêt (6V) : 470mA ;
- Dimension : 55mm x 48.3mm x 23mm ;
- Poids : 45g.

II.2. Partie Logicielle

II.3.1.Environment de programmation des cartes Arduino (IDE Arduino)

L'environnement de développement Arduino (Figure II.10) est une application Java multiplateforme (fonctionnant sur plusieurs systèmes d'exploitation), servant d'éditeur de code et de compilateur, qui peut transférer le firmware (le programme) au travers de la

liaison série asynchrone. Le langage de programmation est une variante du C/C++, allégée et restreinte à l'utilisation de la carte, de ses entrées/sorties et de ses bibliothèques associées [7].

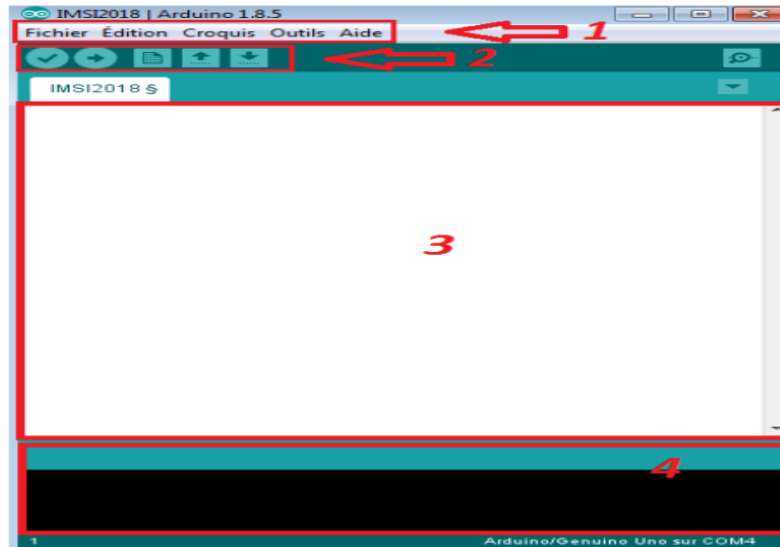


Figure II.10 : Présentation des parties principales du logiciel[15]

- Cadre N°1 : ce sont les options de configuration du logiciel,
- Cadre N°2 : il contient les boutons qui vont nous servir lors de la programmation des cartes.
- Cadre N°3 : ce bloc va contenir le programme que nous allons créer
- Cadre N°4 : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme, c'est le débogueur.

II.2.1. Injection de programme dans la carte Arduino

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) « 1 » et le numéro de port USB (COM 4 dans notre cas) « 2 » comme à titre d'exemple la figure II.11.

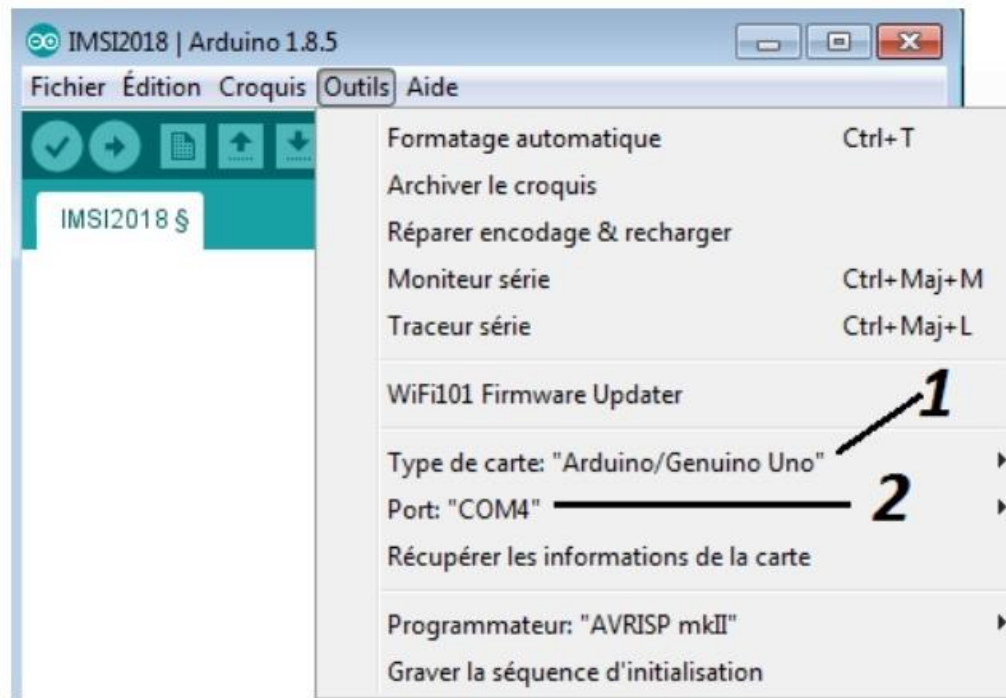


Figure II.11 : Sélection du type de la carte et le port USB.

II.2.2. Structure d'un programme Arduino

Un programme Arduino est une suite d'instructions élémentaires sous forme textuelle (Ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes. Un programme Arduino contient deux fonctions principales[10] :

```
//déclaration des variables et des constantes
Voidsetup() //fonction d'initialisation de la carte
{//contenu de l'initialisation
}
Voidloop()//fonction de la boucle infinie
{//suite des lignes de programmes qui seront exécutées à l'infinie
}
```

II.2.3. Instructions de base de programmation

a) Instructions d'affectation des ports de la carte

La carte Arduino possède des ports digitaux et analogiques qui peuvent être utilisés comme entrée ou sortie :

- Pour l'utilisation comme entrée :pinMode(num de pin,INPUT)
- Pour l'utilisation comme sortie :pinMode(num de pin,OUTPUT).

b) Instructions d'activation d'une sortie

digitalWrite(num de pin,HIGH).

c) Instructions de désactivation d'une sortie

digitalWrite(num de pin,LOW).

d) Lecture d'une entrée logique

digitalRead(num de pin).

e) Lecture d'une entrée analogique

analogRead(num de pin)

II.4.Conclusion

Dans ce chapitre, nous avons présenté une vue d'ensemble de la plateforme de développement Arduino utilisé dans le cas de notre application. Nous avons de ce fait divisé ce chapitre en deux grandes parties : matérielle et logicielle ou software et Hardware en anglais.

Dans la partie matérielle, nous avons présenté la carte Arduino et les différents composants électronique liés à la conception du robot mobile.

En ce qui concerne la partie logicielle, nous avons expliqué comment injecter (charger) un programme dans une carte Arduino et les principales instructions pouvant être utilisées.

Le prochain chapitre sera consacré aux détails de conception de notre système et aux différents tests effectués pour confirmer la bonne mise en marche de notre voiture autonome.

III.1. Introduction

Après avoir présenté les différents aspects liés aux voitures autonomes réelles dans le premier chapitre et après avoir eu une idée détaillée dans le second chapitre sur la plateforme de développement Arduino qui est utilisé comme élément de base servant à la commande et au traitement numérique de notre application, nous allons passer maintenant à notre travail effectué qui se divise en deux grandes parties :

1. Partie hardware
2. Partie software

III.2. Description globales du système réalisé

III.2.1. Schéma synoptique de la réalisation

La figure III.1 représente le schéma synoptique de notre système réalisé.

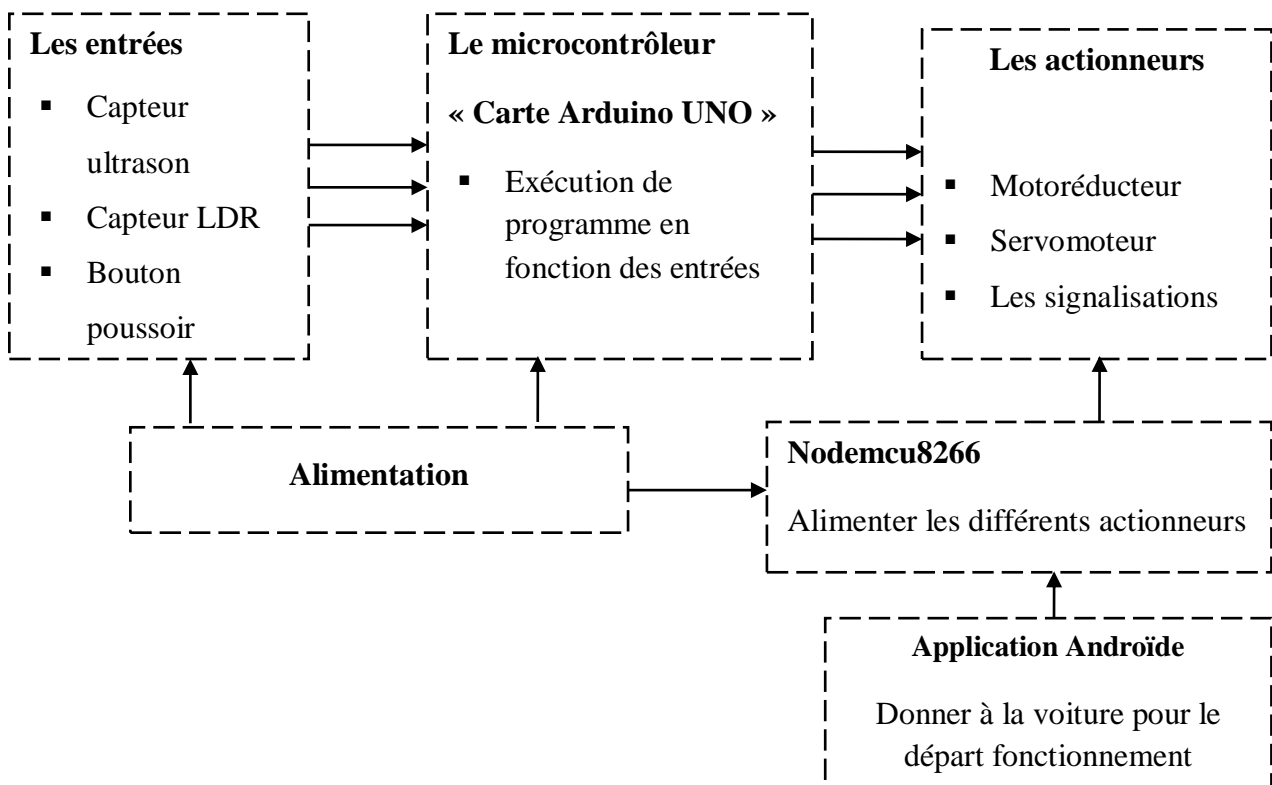


Figure III.1 : Schéma synoptique de notre système réalisé

III.2.2. Fonctionnement général du système de notre voiture autonome

Pour comprendre le fonctionnement général de notre système réalisé, nous allons donner quelques détails sur les différentes tâches réalisées dans un ordre d'enchaînement logique et chronologique pour pouvoir couvrir toutes les parties et aspects liés au travail, de l'étape d'acquisition des signaux nécessaires via les différents capteurs, à la commande des différentes actions via la carte de développement Arduino UNO, puis à l'action elle-même que doivent prendre les actionneurs sur la base des informations issus des capteurs ultrasons pour se déplacer ou faire une rotation à gauche ou à droite afin de réaliser aussi la tâche d'évitement des obstacles.

Cette voiture est conçu pour un but, de naviguer dans un endroit inconnu plein d'obstacles et d'avoir une possibilité d'éviter ces obstacles et de choisir un autre itinéraire. Pour cette raison sur la face avant de la voiture on trouve un capteur ultrason pour la détection des obstacles qui se trouve dans le rayon configuré.

Afin d'avoir la possibilité de mesurer la distance centrale et celle des deux côtés, le capteur ultrasons est fixé sur un servomoteur pour pouvoir l'orienter à la direction choisie pour la mesure de la distance.

La voiture fonctionne selon une logique programmée, elle avance en absence d'obstacles. Dans le cas de détection d'un obstacle proche la voiture s'arrête ; après, le servomoteur s'oriente vers la droite et la gauche afin de mesurer les distances.

Après avoir comparé les distances gauche et droite si elles sont supérieures à la marge indiquée, la voiture tourne vers le côté où l'obstacle est plus loin.

Dans le cas où les distances droite, gauche et centre sont inférieures à la marge programmée, la voiture fait un recule vers l'arrière pendant 500ms, après, elle refait les mesures des distances pour prendre la nouvelle décision.

La voiture est dotée des signaux de signalisation pour signaler l'arrêt ou la direction choisie.

Le mouvement de la voiture est assuré par deux moteurs à courant continu où la vitesse et le sens de rotation sont assurés via un driver L298.

Les actionneurs de la voiture ne sont alimentés que si seulement on donne l'ordre d'alimentation via une application android où la liaison est faite via le module nœudemcu 8266.

III.2.3. Branchement des composants électroniques à carte Arduino

La figure III.2 illustre le branchement des différentes broches de la carte Arduino.

		Carte Arduino UNO	13	
			12	LED
			11	IN4
			10	LED
			9	IN3
	5v		8	IN2
	GND		7	IN1
	GND		6	ENA
	A0		5	ENB
	A1		4	LED
	A2		3	LED
Trig	A3		2	LAMPE
echo	A4		1	TX
	A5		0	RX

Figure III.2 : Branchement des composants électroniques à la carte Arduino

	Entrée digital
	Sortie digital
	Entrée analogique
	Porte de communication
	+VCC (5V)
	La masse

La figure III. 3 illustre un schéma électrique du branchement détaillé de quelques capteurs et actionneurs avec la carte Arduino UNO.

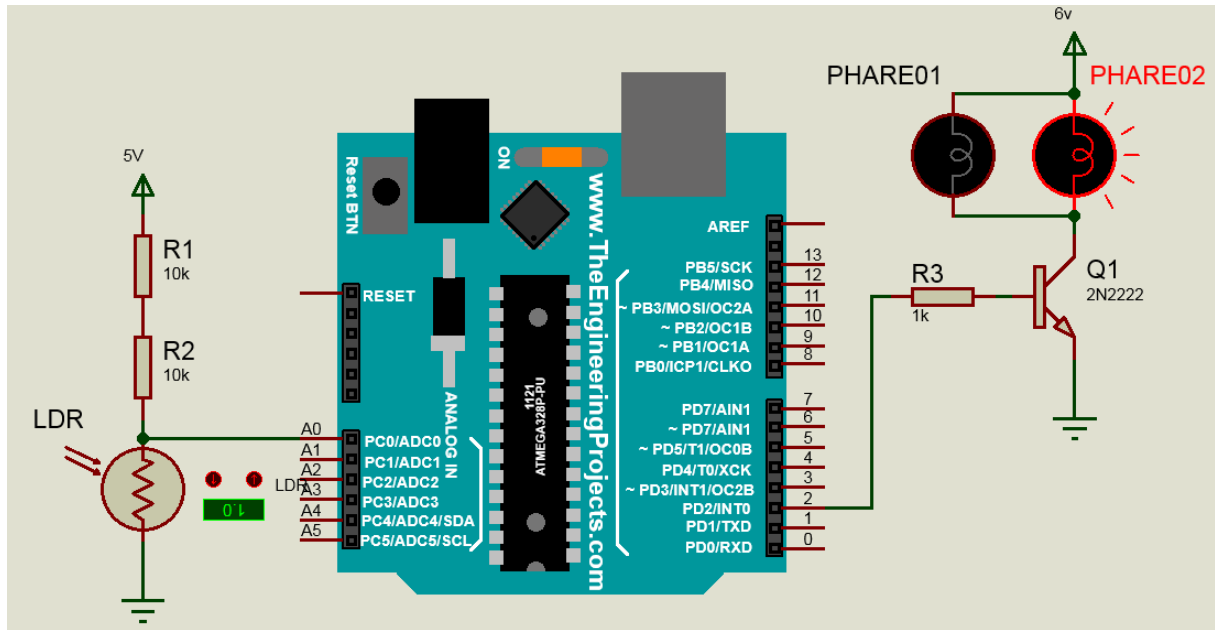


Figure III.3 : Branchement détaillé des capteurs et actionneurs avec la carte Arduino UNO

III.2.4. Liaison carte Arduino et nodemcu

Cette liaison est faite pour un but de donner l'ordre d'activer les actionneurs (motoréducteur, servomoteur et signalisation) tel que le présente la figure III.4.

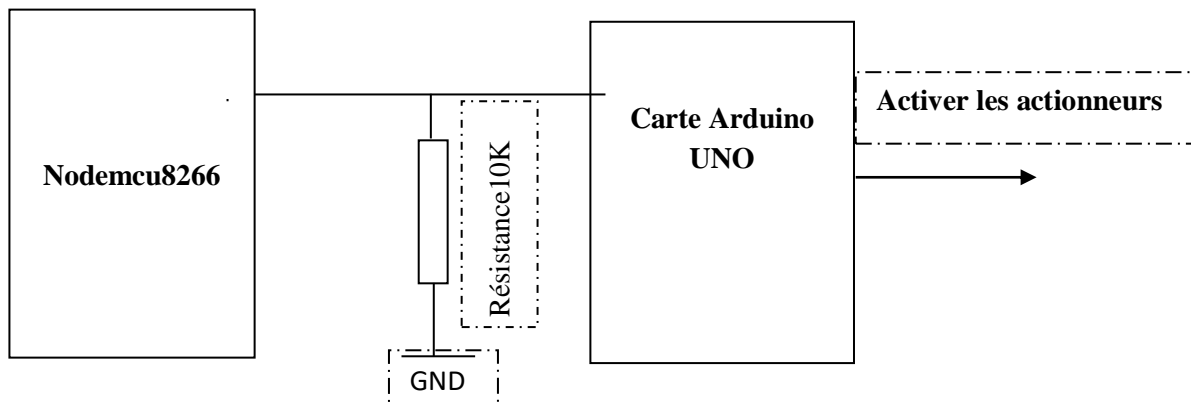


Figure III.4 : Liaison carte Arduino et nodemcu

III.2.5.Calcul courant pour quelques composants électroniques

La figure III.5 montre le branchement de la LED et le calcul de la résistance branchée en série pour la protéger de toute valeur de courant élevée.

$$V_{led} = 2 \text{ V}$$

$$V_r = 5 - v_{led}$$

$$V_r = 3 \text{ v}$$

$$I_r = I_{led} \text{ (montage serie)}$$

$$I_r = I_{led} = 10 \text{ mA}$$

$$R = V_r / I_r = 3 / 0.01 = 330 \Omega$$

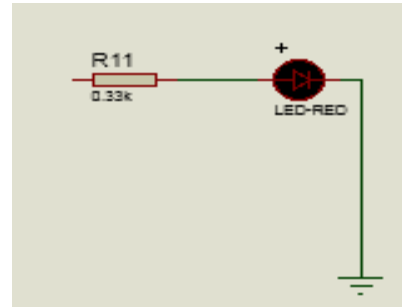
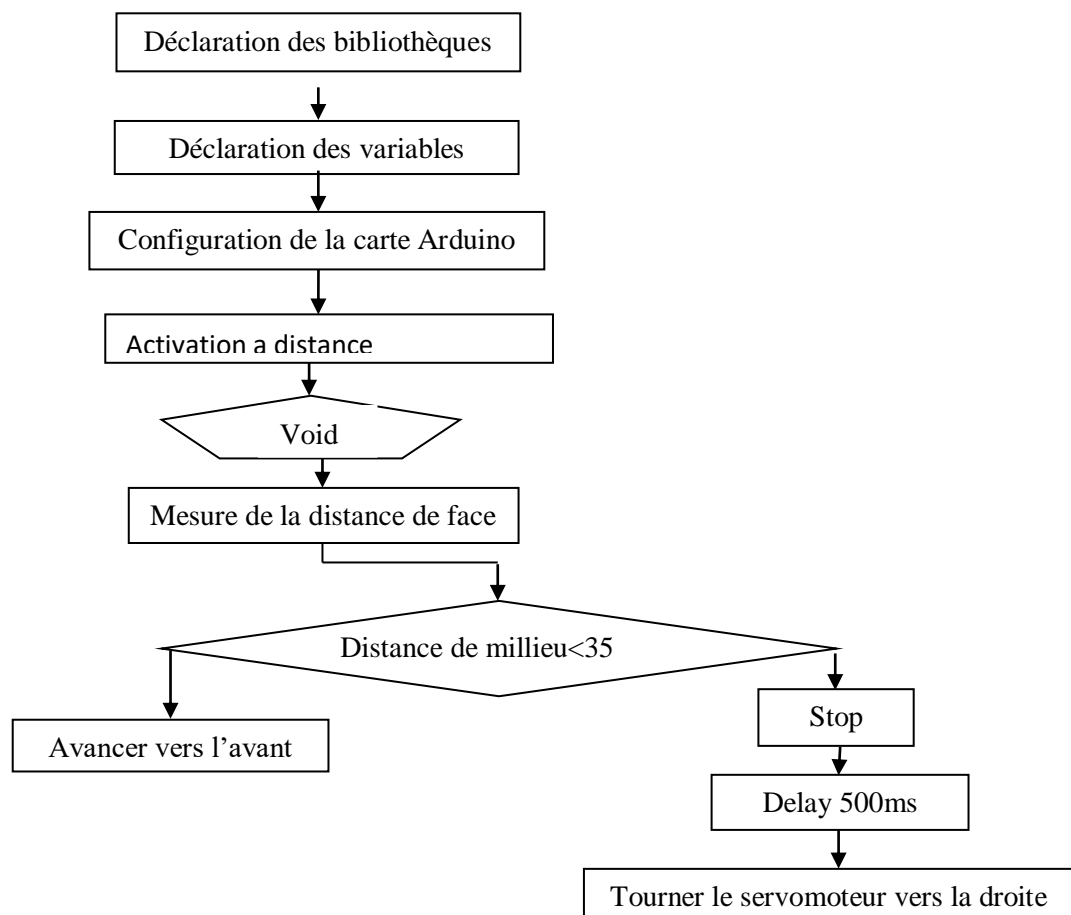


Figure III.5 :Branchement de laLED

III.3.Programmation des cartes électroniques de la voiture autonome

III.3.1.Logigramme de fonctionnement du système complet réalisé

La figure III.6 représente le logigramme qui explique le fonctionnement du système complet réalisé ou spécialement la partie logicielle de notre projet.



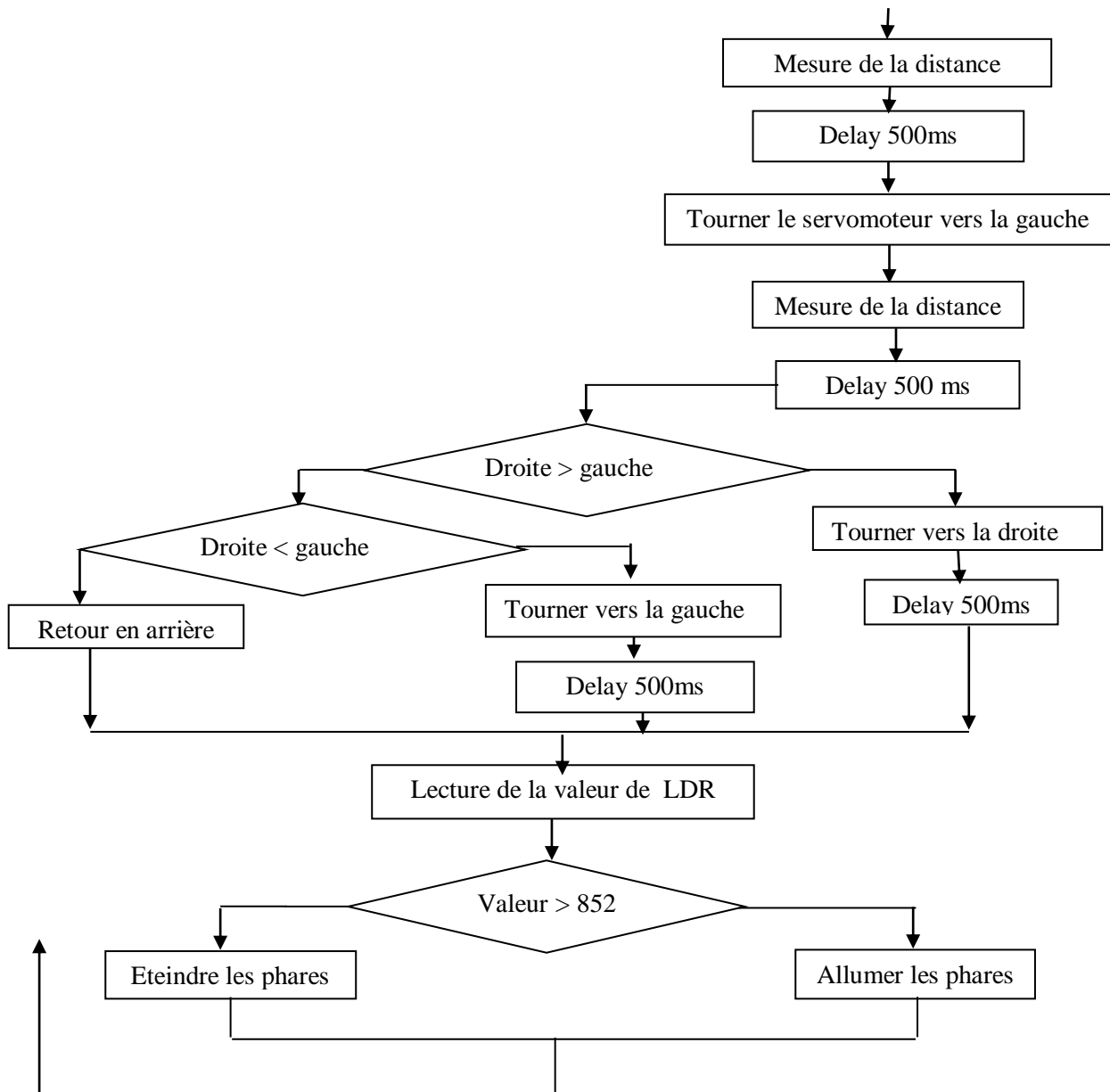


Figure III.6 : Logigramme su système complet réalisé

III.3.2. Déroulement du programme de la carte Arduino

Dans cette partie du mémoire, nous allons présenter les principaux programmes qui réalisent les différentes tâches précédemment cité dans le logigramme global.

```
#include<Servo.h>//servolibrary

Servomy servo; // createservoobject to control servo

intphare=2,tayfost=4,tachlyht=10,ahbas=12;

int valeur=0;

#defineEcho A4 //branchement capteur ultrason

#defineTrig A3 //branchement capteur ultrason

#define ENB 5

#define IN1 7

#define IN2 8

#define IN3 9

#define IN4 11

#define ENA 6

#definecarSpeed 80//vitesse initial des moteurs

intrightDistance = 0, leftDistance = 0, middleDistance = 0;

//variables pour mesure des distanceviacapteur ultrason

voidforward(){

analogWrite(ENA, carSpeed);//effectuer la vitesse rotation moteur(1)

analogWrite(ENB, carSpeed);//effectuer la vitesse rotation moteur(2)

digitalWrite(IN1, LOW);//désactiver le sens arrier

digitalWrite(IN2, HIGH);//désactiver le sens avant

digitalWrite(IN3, HIGH);//désactiver le sens avant

digitalWrite(IN4, LOW);//désactiver le sens arrier

Serial.println("Forward");//afficher sur le srialmoniteur le mot "Forwad"

}

digitalWrite(tayfost,LOW);

digitalWrite(tachlyht,LOW);

digitalWrite(ahbas,LOW);
```

Déclaration des
variables et des
bibliothèques

Fonction
d'avancer
vers l'avant

```
void back() {  
analogWrite(ENA, carSpeed);  
analogWrite(ENB, carSpeed);  
digitalWrite(IN1, HIGH);  
digitalWrite(IN2, LOW);  
digitalWrite(IN3, LOW);  
digitalWrite(IN4, HIGH);  
Serial.println("Back");  
}
```

Fonction recule vers
l'arrière

```
digitalWrite(tayfost,LOW);  
digitalWrite(tachlyht,LOW);  
digitalWrite(ahbas,HIGH);
```

```
void left() {  
analogWrite(ENA, carSpeed);  
analogWrite(ENB, carSpeed);  
digitalWrite(IN1, LOW);  
digitalWrite(IN2, HIGH);  
digitalWrite(IN3, LOW);  
digitalWrite(IN4, HIGH);  
Serial.println("Left");  
}
```

Fonction Tourner vers la
droite

```
digitalWrite(tayfost,LOW);  
digitalWrite(tachlyht,HIGH);  
digitalWrite(ahbas,LOW);
```

```
void right() {  
  analogWrite(ENA, carSpeed);  
  analogWrite(ENB, carSpeed);  
  digitalWrite(IN1, HIGH);  
  digitalWrite(IN2, LOW);  
  digitalWrite(IN3, HIGH);  
  digitalWrite(IN4, LOW);  
  Serial.println("Right");  
}
```



Fonction tourner vers
la gauche

```
digitalWrite(tayfost,HIGH);  
digitalWrite(tachlyht,LOW);  
digitalWrite(ahbas,LOW);  
}
```

```
void stop() {  
  digitalWrite(ENA, LOW);  
  digitalWrite(ENB, LOW);  
  
  Serial.println("Stop!")  
  
  digitalWrite(tayfost,LOW);  
  digitalWrite(tachlyht,LOW);  
  digitalWrite(ahbas,HIGH);  
}
```



Fonction
Arrêt

```
int getDistance()  
{//fonction pour la mesure de la distance
```

```
digitalWrite(Trig, LOW);  
delayMicroseconds(2);  
digitalWrite(Trig, HIGH);  
delayMicroseconds(10);  
digitalWrite(Trig, LOW);  
return (int)pulseIn(Echo, HIGH) / 58;  
}
```

Fonction
pour mesure
la distance

```
void setup() {  
myservo.attach(3); // attach servo on pin 3 to servo object  
Serial.begin(9600); // fréquence de communication avec le serial monitor  
pinMode(Echo, INPUT); // déclarer la brache écho comme entrer "A4"  
pinMode(Trig, OUTPUT); // déclarer la brache trig comme sortie "A3"  
pinMode(IN1, OUTPUT); // déclarer la brache IN1 comme sortie  
pinMode(IN2, OUTPUT); // déclarer la brache IN2 comme sortie  
pinMode(IN3, OUTPUT); // déclarer la brache IN3 comme sortie  
pinMode(IN4, OUTPUT); // déclarer la brache IN4 comme sortie  
pinMode(ENA, OUTPUT); // déclarer la brache ENA comme sortie  
pinMode(ENB, OUTPUT); // déclarer la brache ENB comme sortie  
stop();  
}  
  
pinMode(phare, OUTPUT);  
pinMode(tayfost, OUTPUT);  
pinMode(tachlyht, OUTPUT);  
pinMode(ahbas, OUTPUT);  
digitalWrite(phare, LOW);  
digitalWrite(tayfost, LOW);  
digitalWrite(tachlyht, LOW);
```

Configuration
de la carte
Arduino

digitalWrite(ahbas,LOW);

```
void loop() {
  valeur=analogRead(A0);
  //valeur=(valeur*100)/1023;
  Serial.println(valeur);
  // valeur=(valeur*100)/1023;
  if(valeur>852)
  {
    digitalWrite(phare,HIGH);
  }
  else
  {
    digitalWrite(phare,LOW);
    myservo.write(90); //setservo position according to scaled value
    delay(500);
    middleDistance = getDistance();//mesurer la distance et
    la enregistrer sur la variable "middleDistance"

    if(middleDistance<= 35) {
      stop();
      delay(500);
      myservo.write(10);
      delay(1000);
      rightDistance = getDistance();
      Serial.println(rightDistance);

      delay(500);
      myservo.write(90);
      delay(1000);
      myservo.write(180);
      delay(1000);
      leftDistance = getDistance();
      Serial.println(leftDistance);
      delay(500);
      myservo.write(90);
      delay(1000);
      if(rightDistance>leftDistance) {
        right();
        delay(50);
      }
      else if(rightDistance<leftDistance) {
        left();
      }
    }
  }
}
```



```
delay(50);
}
else if((rightDistance<= 40) || (leftDistance<= 40 )) {
back();
delay(180);
}
else {
forward();
}
}
else {
forward();
}
}
```

III.3.3.Interface de commande via wifi

C'est une interface de remote controlde notre voiture autonome via le wifi à la l'aide d'un Smartphone ou un micro portable cette interface est fait avec page web en HTML injectée sur le Nodemcu 8266.Voir la figure III.7.



FigureIII .7 :Interface de commande via wifi

III.3.4. Programme implanté sur le nodemcu8266

Cette partie décrit le programme implémenté sur le nodemcu8266.

```
#include<ESP8266WiFi.h>
```

```
const char* ssid = "NabilSaid";
```

```
const char* password = "12341234";

int ledPin = D0;
WiFiServer server(80);

void setup()
{
  // initialisation de la communication série
  Serial.begin(115200);

  delay(100);

  // initialisation de la sortie pour la led
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  // Connexion wifi
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  // connection en cours ...
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }

  // Wifi connecter
  Serial.println("WiFi connecter");

  // Démarrage du serveur.
  server.begin();
  Serial.println("Serveur démarré !");

  // Affichage de l'adresse IP
  Serial.print("Utiliser cette adresse URL pour la connexion :");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("");
}

void loop()
{
  WiFiClient client;

  // Vérification si le client est connecté.
```

```
client = server.available();
if (!client)
{
  return;
}

// Attendre si le client envoie des données ...
Serial.println ("nouveau client");
while (!client.available()){
  delay (1);
}

String request = client.readStringUntil ('\r');
Serial.println (request);
client.flush();

int value = LOW;
if (request.indexOf("/LED=ON") != -1) {
  digitalWrite(ledPin, HIGH); // allumer la led
  value = HIGH;
}
if (request.indexOf("/LED=OFF") != -1) {
  digitalWrite(ledPin, LOW); // éteindre la led
  value = LOW;
}

// Réponse
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
client.println("<!DOCTYPE HTML>");
client.println("<html>");

client.print("Etat de la led : ");

if(value == HIGH) {
  client.print("On");
} else {
  client.print("Off");
}
client.println("<br><br>");
client.println("<a href=\"/LED=ON\"><button>Allumer</button></a>");
client.println("<a href=\"/LED=OFF\"><button>Eteindre</button></a><br />");
client.println("</html>");

delay(1);
Serial.println("Client deconnecter");
Serial.println("");
}
```

III.4.Étalonnage du capteur de lumière LDR

Le capteur LDR va être étalonné, où la source d'excitation est une source de lumière et l'affichage des valeurs est fait à l'aide de moniteur série de l'IDE Arduino. La figure III.8 représente la courbe d'étalonnage du capteur LDR.

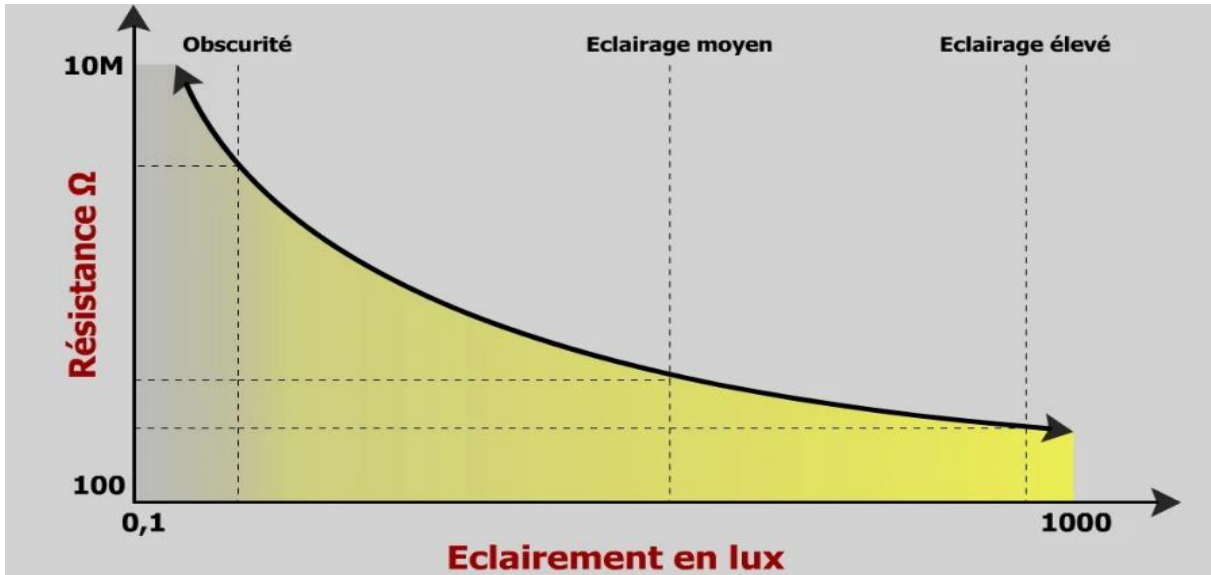


Figure III.8 : Courbe caractéristique de LDR

➤ Montage d'étalonnage du capteur LDR

La figure III.9 représente le schéma électrique d'étalonnage du capteur LDR.

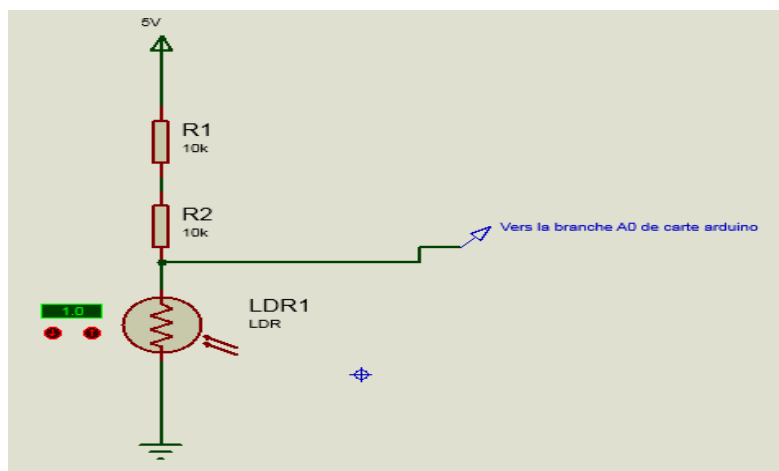


Figure III.9 : Montage d'étalonnage du capteur LDR

➤ Résultat d'étalonnage de capteur LDR

Le tableau III.1 représente les résultats de mesure de lumière en fonction du temps en se basant sur le schéma d'étalonnage de la figure III.9.

Le temps	Mesure en « Lumière normale » de LDR	Mesure en présence de lumière	Mesure en absence de lumière
8 s	785	307	1015
16 s	762	234	1022
24 s	693	99	988
32 s	685	95	1022
40 s	852	88	1021
48 s	852	88	1021
56 s	852	90	1016

Tableau III.1 : Résultat de mesure d'étalonnage du capteur LDR

Après avoir fait l'étalonnage nous avons pris la valeur 852 bits comme valeur intermédiaire pour l'activation des phares de la voiture.

L'excitation c'est le milieu environnant où se trouve la voiture. Donc, nous avons choisi trois milieux différents car nous n'avons pas de Luxmètre pour la mesure de la lumière. Les mesures sont prises chaque 8 seconds, la lecture est faite via le moniteur série de IDE Arduino.

III.5 Étalonnage du capteur ultrason

➤ Procédure d'étalonnage

Afin d'établir la distance à respecter entre la voiture et l'obstacle, nous avons comparé la distance obtenue par le capteur ultrason et les distances réelles. La lecture de la distance mesurée se fait sur le « monitor série de l'IDE Arduino ».



Figure III.10 : Etalonnage du capteur ultrason

➤ Résultats obtenus

Le tableau III.2 représente les résultats de mesure de distance et d'étalonnage du capteur ultrason. Cette expérience est réalisée dans le but de confirmer la précision du capteur ultrason en comparant les distances fixées dans le programme aux valeurs de distances mesurées réellement.

Distance réelle	Distance mesurée
10	10
20	20
30	31
40	41
70	73

Tableau III.2 : Mesures d'étalonnage du capteur ultrason

➤ **Programme implanté sur Arduino pour étalonnage de capteur ultrason**

Dans cette partie nous présentons le programme écrit pour l'étalonnage du capteur ultrasons utilisé pour la détection et évitement d'obstacles.

```
#define Echo A4
#define Trig A3
int rightDistance = 0, leftDistance = 0, middleDistance = 0;

//Ultrasonic distance measurement Sub function

int getDistance() {
    digitalWrite(Trig, LOW);
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    return (int)pulseIn(Echo, HIGH) / 58;
}

void setup() {
```

```
Serial.begin(9600);  
  
pinMode(Echo, INPUT);  
  
pinMode(Trig, OUTPUT);  
  
}  
  
void loop() {  
  
middleDistance = getDistance();  
  
Serial.println(middleDistance);  
  
delay(500);  
  
}
```

III.6. Essais et résultats des différentes manipulations

Dans cette partie dernière partie des résultats, nous allons présenter quelques essais des différentes manipulations sur notre voiture autonome et qui montrent le bon fonctionnement de notre système réalisé illustré par les différentes figures de la figure III.13 à la figure III.17.

III.6.1. Actionneurs et test de voiture activée

\$

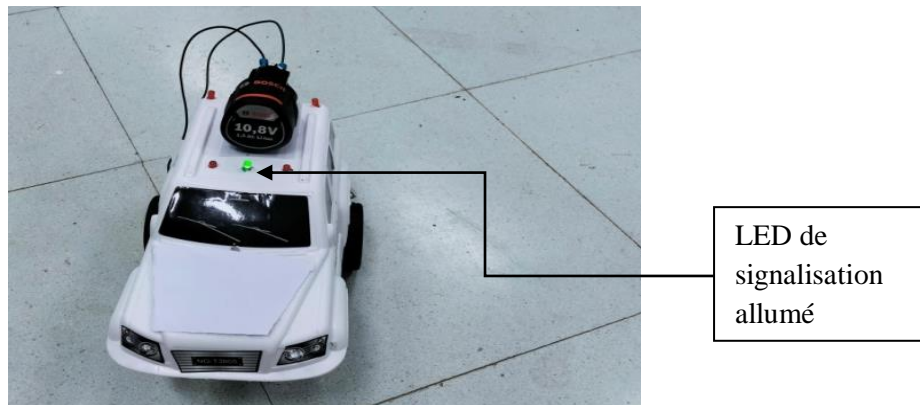


Figure III.11 :Voiture activée

III.6.2. Tests de voiture en présence d'obstacle

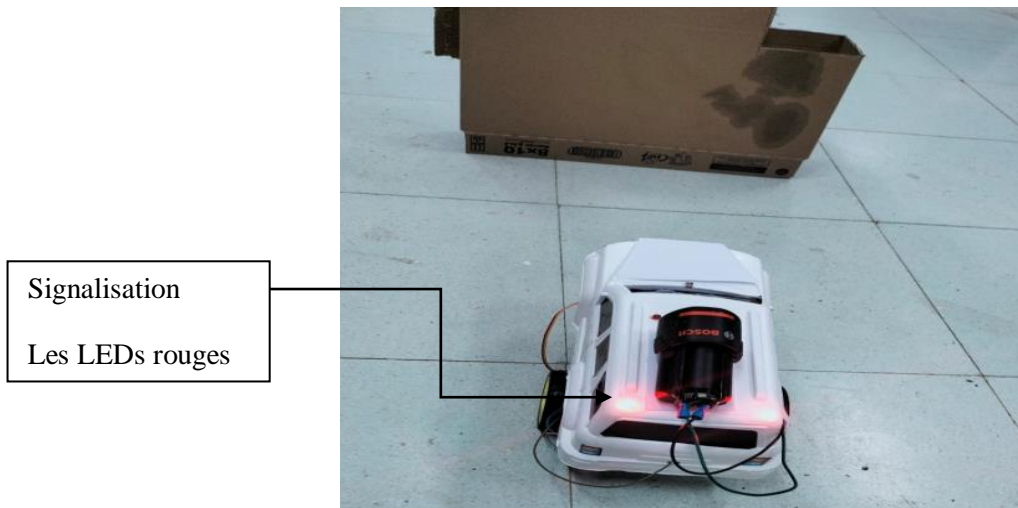


Figure III.12 : Voiture en présence d'obstacle

III.6.3. Test de déviation de la voiture vers la droite pour l'évitement d'obstacles

Figure

Signalisation de
la LED droite



III.13: Déviation de la voiture vers la droite pour éviter un obstacle

III.6.4. Test de déviation de la voiture vers la gauche pour l'évitement d'obstacles



Figure III.14 : Déviation de la voiture vers la gauche

III.6.6. Test de marche de la voiture dans un milieu obscur



Figure.III.15 :Marche dans un milieu obscur

III.7. Conclusion

Dans ce dernier chapitre, nous avons explicité les différentes étapes qui nous ont permis de réaliser notre système de voiture autonome de sa partie acquisition, à la partie commande à sa dernière partie d'action et de prise de décision pour éviter un obstacle.

Nous avons présenté aussi une description générale du programme Arduino et nodemcu utilisé dans notre projet et nous avons vers la fin effectué les différents tests relative u fonctionnement de notre projet et qui montrent le bon fonctionnement de notre système ou voiture autonome.

Conclusion générale

Dans ce thème master, nous avons pu exploiter toutes nos connaissances en électronique, électricité, mécanique, automatisme et en informatique afin de réaliser un prototype d'une voiture autonome.

Nous avons réalisé une mini voiture autonome capable de circuler dans un endroit inconnu, plein d'obstacles physiques où elle est capable d'éviter ces obstacles et trouver un accès.

Dans cette optique, nous avons doté cette voiture de capteurs ultrason pour éviter ces obstacles, ce capteur ultrason mesure la distance entre le robot et l'obstacle.

Afin d'assurer la mesure des différentes distances (avant, droite et gauche), ce dernier est monté sur un support capable de tourner vers la droite et gauche à l'aide d'un servomoteur relié mécaniquement à ce support.

Cette voiture se déplace sur deux roues arrière et une roulette avant qui facilite l'orientation vers les deux sens. L'orientation de la voiture est faite par un mécanisme des deux roues arrière par la rotation inverse d'une roue par rapport à l'autre.

Comme ce robot a la capacité d'activer les phares avant en fonction de la lumière du milieu entouré. Cette fonction est réalisée à l'aide d'un capteur LDR.

Le traitement des informations des capteurs et la décision de l'état des actionneurs sont réalisés par la carte Arduino UNO. Nous avons utilisé la carte nodemcu 8266 afin d'activer les actionneurs de cette voiture à distance via un Smartphone ou un PC.

Ce travail nous a permis de traiter des problèmes d'ordre pratique et de vérifier nos connaissances théoriques acquises tout au long de notre formation.

Grâce au travail continu, nous avons pu atteindre notre but et satisfaire le cahier de charge, mais cela ne veut pas dire qu'il est complet, nous proposons que le robot réalisé soit la base de toute une série d'améliorations que nous n'avons pas eu la chance de les faire par manque de temps et de matériel.

Dans un but de la continuité et l'amélioration de ce travail, nous estimons l'utilisation de capteurs plus performants comme les capteurs laser ou les caméras CCD et d'utiliser plus de capteurs ou un plateau de capteurs rotatifs pour couvrir plus l'environnement autour.

Bibliographie

- [1] B. Abdelaziz, « Véhicule autonome : Les systèmes de perception embarqués », Institut National des Sciences Appliquées de Rouen, STPI/P6, 2019
- [2] « [https://www.kronobase.org/chronologie-categorie V%C3%A9hicule+autonome.html](https://www.kronobase.org/chronologie-categorie-V%C3%A9hicule+autonome.html) ». consultée le 01/06/2022
- [3] Paul Belleflamme, « Analyse prospective sur l'implémentation de la voiture autonome », UNIVERSITE CATHOLIQUE DE LOUVAIN LOUVAIN SCHOOL OF MANAGEMENT, ANNEE ACADEMIQUE 2014-2015
- [4] Albert ROSSIER, « Les systèmes embarqués dans l'automobile » Haute École de Gestion de Genève (HEG-GE) Filière Informatique de Gestion
- [5] « <https://tpegtjb.wixsite.com/voiture-autonome/copie-de-plan-i> », consultée le 02/06/2022
- [6] « <https://mbamci.com/6-niveaux-voiture-autonome/> » consultée le 02/06/2022
- [7] « <https://www.nhtsa.gov/document/highway-safety-related-highway-safety-provisions> ». consultée le 03/06/2022
- [8] <https://www.oscaronews.com/voitures-autonomes-avantages-et-inconvenients> consultée le 05/06/2022
- [9] « <http://www.generationrobots.com/fr/152-arduino> » . consultée le 07/06/2022
- [10] Nasri yanis , naimi aissa mémoire étude et conception d'un dispositif électronique d'électeur de gaz , juin 2018 université oran
- [11] « <https://www.gcworks.fr/tutoriel/arduino/Uno.html> » consultée le 07/06/2022
- [12] « <https://www.aranacorp.com> » consultée le 08/06/2022
- [13] « <http://for-ge.blogspot.com/2015/05/la-photoresistance-ldr.html> » consultée le 12/06/2022
- [14] « <https://fr.wikipedia.org/wiki/ESP8266> » consultée le 12/06/2022
- [15] **Erik Batman** le grand livre d'arduino, édition Eyrolles 2015