

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ IBN-KHALDOUN DE TIARET
FACULTÉ DES SCIENCES APPLIQUÉES
DÉPARTEMENT DE GENIE ELECTRIQUE



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Génie Electrique

Spécialité : Electronique des Systèmes Embarqués

THÈME

*Etude et réalisation d'un robot suiveur
de ligne contrôlé par une carte Arduino*

Préparé par : HAMDI Rym Wafaa

Devant le Jury :

Nom et prénoms

Mr BENATIA Abdelrahmane

Mr GHELLAB Abdelkader

Mr KOUADRIA Mouhamed

Qualité

Président

Examineur

Encadreur

Promotion 2019/2020

Dédicace

*Du profond de mon cœur je dédie ce travail à tous ceux
qui me sont chers,*

Mon père

Puisse Dieu l'accueillir dans son vaste paradis.

Ma chère mère

*Je vous remercie pour tout le soutien et l'amour que
vous me portez depuis mon enfance et j'espère que votre
bénédictio n m'accompagne toujours.*

*Que ce modeste travail soit le fruit de vos innombrables
sacrifices.*

Mes sœurs

Mes amis

Tous les membres de ma famille

Tous mes amis et mes collègues de promotion 2020

Remerciement

Nous tenons tout d'abord à remercier DIEU de nous avoir donné le courage et la détermination nécessaire pour finaliser ce travail qui compte tant pour notre avenir intellectuel et professionnel.

*Nous tenons notre immense gratitude à **nos parents** pour leur accompagnement tout au cours de ce cycle d'étude.*

En second lieu, Nous remercions tout particulièrement notre encadreur dr «KOUADRIA MOHAMED» qui a fait preuve d'une grande patience tout au long de notre recherche en nous prodiguant ses conseils et ses remarques éclairées qui nous ont été particulièrement précieuses.

*Nos remerciements sincères aux **membres de jury**, de nous avoir fait l'honneur de lire et d'évaluer notre travail et malgré leurs innombrables occupations.*

Nous remercions à monsieur « bouzidi».

*Nous remercions **nos professeurs** pour leurs enseignements lesquels nous ont donné le goût de la rigueur scientifique et nous ont beaucoup aidés à rédiger, à structurer ce mémoire et à améliorer son contenu.*

Table des matières

| | |
|----------------------------|---|
| Introduction Générale..... | 1 |
|----------------------------|---|

Chapitre I

Généralités sur le robot suiveur de ligne

| | |
|--|---|
| I.1 Introduction..... | 3 |
| I.2 Définitions..... | 3 |
| I.2.1 Actionneur | 3 |
| I.2.2 Capteur | 3 |
| I.2.3 ligne..... | 3 |
| I.2.4 Courbe (ligne courbe) | 3 |
| I.2.5 Robot | 3 |
| I.2.6 Robot suiveur de ligne | 4 |
| I.3 Avantages et inconvénients..... | 4 |
| I.4 Principe de fonctionnement d'un RSL..... | 4 |
| I.5 Domaines d'application | 5 |
| I.6 Organisation générale d'un robot suiveur | 7 |
| I.6.1 Les capteurs | 7 |
| I.7 Conclusion | 9 |

Chapitre II

Présentation de cartes de développement à microcontrôleur

| | |
|--|----|
| II.1 Introduction | 10 |
| II.2 Carte à microcontrôleur pour système embarqué..... | 10 |
| II.3 Cartes à microcontrôleur de type Arduino | 15 |
| II.3.1 Caractéristiques des cartes de type Arduino..... | 16 |
| II.3.2 Avantages et inconvénients | 17 |
| II.4 Eléments principaux d'une carte de développement Arduino..... | 18 |

| | |
|------------------------------|----|
| II.4.1 Partie Matériel | 19 |
| II.4.2 Partie logiciel | 20 |
| II.5 Conclusion | 25 |

Chapitre III

Etude et conception du robot suiveur de ligne

| | |
|--|----|
| III.1 Introduction | 26 |
| III.2 Etude des circuits de suiveur de ligne | 26 |
| III.2.1 Sans carte Arduino | 26 |
| III.2.1.1 Schéma synoptique | 26 |
| III.2.1.2 Schéma électrique | 26 |
| III.2.1.3 Principe de fonctionnement | 27 |
| III.2.1.4 Calcul des circuits | 28 |
| III.2.2 Avec carte Arduino | 31 |
| III.2.2.1 Schéma synoptique | 31 |
| III.2.2.2 Schéma électrique | 31 |
| III.2.2.3 Principe de fonctionnement | 32 |
| III.2.2.4 Calcul des circuits | 33 |
| III.3 Etude comparative entre les cas sans et avec Arduino | 34 |
| III.4 Conclusion..... | 35 |

Chapitre IV

Simulations des circuits

| | |
|--|----|
| IV.1 Introduction | 36 |
| IV.2 Partie logicielle | 36 |
| IV.2.1. Plateforme de programmation Arduino..... | 36 |
| IV.2.2. Plateforme de développement et de simulation Proteus | 37 |
| IV.3. Simulations des circuits étudiés | 38 |
| IV.3.1.Sans carte Arduino | 38 |

| | |
|---|----|
| IV.3.1.1 Organigramme fonctionnel..... | 38 |
| IV.3.1.2 Simulation du schéma électronique..... | 39 |
| IV.3.2 Avec carte Arduino | 41 |
| IV.3.2.1 Organigramme fonctionnel..... | 41 |
| IV.3.2.2 Simulation du schéma électronique..... | 42 |
| IV.4 Conclusion | 48 |

Chapitre V

Réalisations pratiques. Cas avec carte Arduino

| | |
|---|----|
| V.1 Introduction | 49 |
| V.2 Organigramme de fonctionnement..... | 49 |
| V.3 Matériel et logiciel utilisés | 49 |
| V.4 Réalisation pratique sur une maquette..... | 50 |
| V.4.1.Construction du Robot..... | 50 |
| V.4.2. Les différentes fonctions réalisées dans notre système | 50 |
| V.4.3. chargement du programme principal..... | 51 |
| V.5 Mise en œuvre | 53 |
| V.6 Nomenclatures..... | 56 |
| V.7 Etude économique des circuits | 57 |
| V.8 Conclusion..... | 57 |
| Conclusion générale | 58 |
| Références bibliographie..... | 59 |
| Annexe | 61 |

Liste des figures

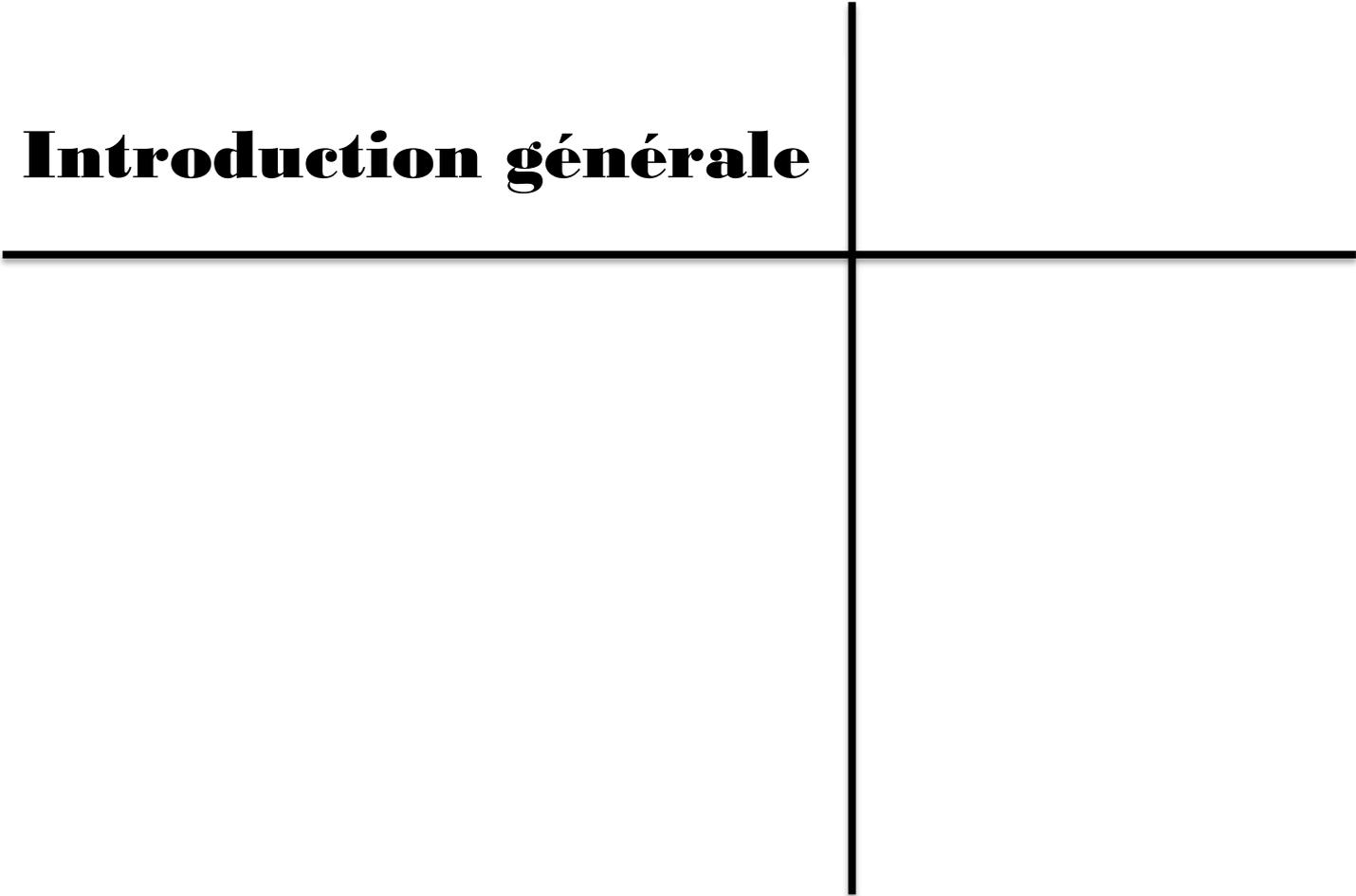
| | | |
|----------------------|---|----|
| Figure I.1 | Robot suiveur de ligne..... | 6 |
| Figure I.2 | Représentation schématique d'une organisation générale d'un suiveur de ligne..... | 7 |
| Figure I.3 | Liste de quelques circuits d'entrées..... | 7 |
| Figure I.4 | Exemple de cartes de développements électroniques à base de microcontrôleurs... | 8 |
| Figure I.5 | Liste de quelques circuits de sorties. | 9 |
| | | |
| Figure II.1 | Domaines de télécommunication intelligente..... | 13 |
| Figure II.2 | Industrie intelligente. | 13 |
| Figure II.3 | Utilisation des microcontrôleurs dans le commercial..... | 13 |
| Figure II.4 | Voiture intelligente. | 14 |
| Figure II.5 | Les armes intelligentes..... | 14 |
| Figure II.6 | Communication entre matlab et ARDUINO. | 14 |
| Figure II.7 | Shields pour carte ARDUINO | 15 |
| Figure II.8 | Brochage d'une carte Arduino de type UNO. | 19 |
| Figure II.9 | Schéma de téléversement..... | 21 |
| Figure II.10 | Schéma du logiciel..... | 22 |
| | | |
| Figure III.1 | Schéma synoptique du sans carte Arduino..... | 26 |
| Figure III.2 | Circuit électrique générale du robot Sans carte Arduino. | 27 |
| Figure III.3 | Circuit électrique de la première partie (circuit d'entrées). | 28 |
| Figure III.4 | Circuit électrique de la deuxième partie (circuit de commande)(a). | 29 |
| Figure III.5 | Circuit électrique de la deuxième partie (b). | 30 |
| Figure III.6 | Circuit électrique de la troisième partie (circuit de puissance). | 30 |
| Figure III.7 | Moteur à courant continu. | 31 |
| Figure III.8 | Schéma synoptique avec carte Arduino. | 31 |
| Figure III.9 | Circuit électrique générale du robot avec carte Arduino..... | 32 |
| Figure III.10 | L293D..... | 34 |
| | | |
| Figure IV. 1 | Structure de programme ARDUINO..... | 37 |
| Figure IV.2 | Représentation par une photo du logiciel Proteus. | 37 |
| Figure IV.3 | Organigramme fonctionnel du suiveur de ligne sans carte Arduino. | 39 |

| | |
|--|----|
| Figure IV.4 Simulation du circuit sans carte Arduino(1)..... | 40 |
| Figure IV.5 Simulation du circuit sans carte Arduino(2)..... | 40 |
| Figure IV.6 Simulation du circuit sans carte Arduino(3)..... | 41 |
| Figure IV.7 Organigramme fonctionnel du suiveur de ligne avec carte Arduino..... | 42 |
| Figure IV.8 Simulation du circuit électrique avec la carte Arduino(a)..... | 43 |
| Figure IV.9 Simulation du circuit électrique avec la carte Arduino(b)..... | 44 |
| Figure IV.10 Simulation du circuit électrique avec la carte Arduino(c)..... | 45 |
| Figure IV.11 Simulation du circuit électrique avec la carte Arduino(d)..... | 46 |
| Figure IV.12 Simulation du circuit électrique avec la carte Arduino(e)..... | 47 |
| | |
| Figure V. 1 Organigramme représente le fonctionnement général du système. | 49 |
| Figure V. 2 La Construction du Robot. | 50 |
| Figure V.3 Paramétrage de la carte étape1..... | 51 |
| Figure V.4 Paramétrage de la carte étape2..... | 51 |
| Figure V.5 Paramétrage de la carte étape3..... | 52 |
| Figure V.6 Téléversement de programme vers la carte Arduino. | 53 |
| Figure V.7 Schéma de la plaque d'essai + moteurs à courant continue +Arduino. | 53 |
| Figure V.8 Schéma de la plaque d'essai + moteurs à courant continu +Arduino+ circuit de puissance. | 54 |
| Figure V.9 Schéma de la plaque d'essai + moteurs à courant continu +Arduino+ circuit de puissance +buzzer. | 54 |
| Figure V.10 Schéma du circuit complet(a). | 55 |
| Figure V.11 Schéma du circuit complet(b). | 55 |
| Figure V.12 Schéma du circuit complet(c). | 56 |

Liste des tableaux

| | | |
|----------------------|--|----|
| Tableau I.1 | Les éléments de base pour le fonctionnement d'un robot suiveur de ligne..... | 5 |
| Tableau II.1 | Quelques types de cartes à microcontrôleur pour les systèmes embarqués. | 12 |
| Tableau II.2 | Caractéristiques de quelques cartes de développement de type Arduino les plus utilisées..... | 16 |
| Tableau II.3 | Avantages et Inconvénients de différentes cartes de type Arduino. | 18 |
| Tableau II.4 | Les parties de Arduino IDE..... | 20 |
| Tableau III.1 | Comparaison entre les cas de circuits utilisés sans et avec carte <i>Arduino</i> | 34 |
| Tableau V. 1 | Tableau représentant le coût des composants du projet. | 57 |

Introduction générale



Introduction générale

Un système embarqué est un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires [01]. Nous trouvons ce type de système dans des domaines divers tels que le transport (avionique, espace, automobile, ferroviaire), dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation, ..., etc.

Aujourd'hui, il existe des systèmes autonomes ayant pour but de suivre une ligne bien déterminée, on parle alors des robots suiveurs de ligne ayant la capacité de servir dans l'industrie, dans les restaurants ou encore dans les transports en commun (bus sans chauffeur) [02].

Ce projet de fin d'étude marque la fin de nos études. Il s'attarde sur la conception d'un robot mobile contrôlé par une carte ARDUINO ayant la capacité de se déplacer sur une trajectoire bien précise.

Comme le sujet a fait l'objet de plusieurs études et n'a jamais été traité au sein de notre université, il a été décidé de porter notre choix dans ce sens.

L'exploitation de ce type de suiveur, essentiellement due à la sécurité des personnes, gain de temps, fonctionnement à longue durée sans arrêt, ...etc.

Le contenu de ce projet de fin d'étude est structuré en cinq chapitres.

Le premier chapitre est consacré à une généralité sur le thème de notre travail, où nous présentons dans un premier temps quelques définitions et types suiveurs de ligne. Dans un deuxième temps, nous donnons le principe de fonctionnement et l'organisation générale d'un suiveur de ligne.

Le cadre général de la carte à microcontrôleur « Arduino » fera l'objet du deuxième chapitre. Dans ce cas, nous allons voir les différentes cartes Arduino ainsi que les avantages et les inconvénients, et plus particulièrement une description détaillée et les caractéristiques des cartes impliquées dans notre projet.

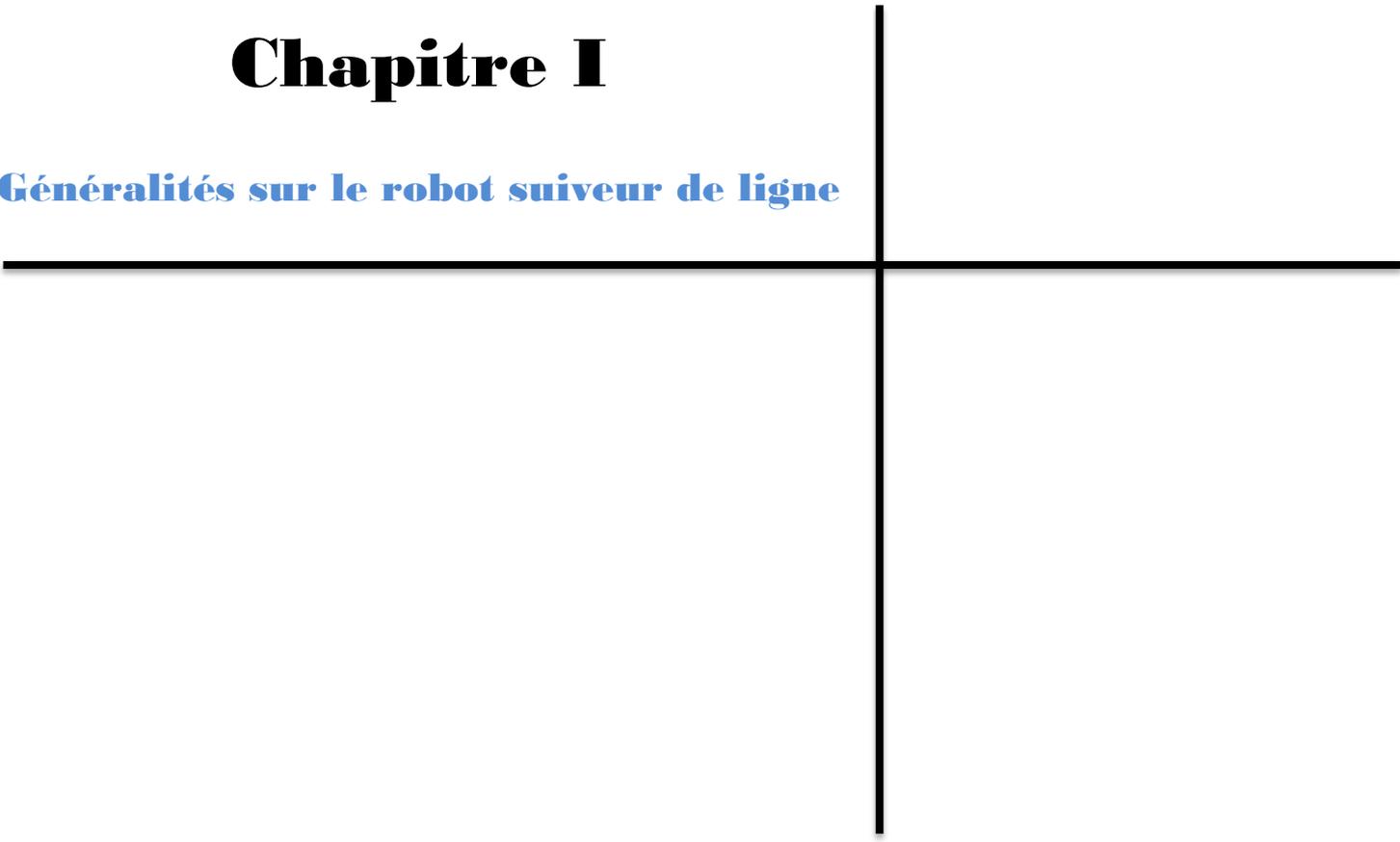
Le troisième chapitre concerne l'étude détaillée du système de notre projet dans les deux cas de situation : sans et avec Arduino.

Le quatrième chapitre sera consacré à la présentation du simulateur (ISIS Proteus) utilisé pour la conception des schémas électroniques correspondant aux différents circuits étudiés, et à l'illustration des différents organigrammes correspondant aux cas de figures des simulations effectuées. Aussi, ce dernier chapitre concerne la maquette du robot suiveur de piste et l'évaluation du coût de sa réalisation.

Nous terminons ce mémoire par une conclusion portant sur les travaux effectués et par une présentation des perspectives de recherche pouvant être envisagées.

Chapitre I

Généralités sur le robot suiveur de ligne



I.1 Introduction

Nous donnons dans ce chapitre un petit rappel sur les robots suiveur de ligne (RSL) et ces domaines d'application. Ce dernier utilise les composants électroniques tels que les composants d'entrées, de sorties et les circuits de commande pour son fonctionnement.

Dans ce cadre, le présent chapitre sera abordé par une présentation de quelques définitions et du cadre général dans le but de citer les avantages et les inconvénients. Ensuite, un contexte sera établi pour donner les différents domaines d'application des RSL, ce qui permettra de montrer le cas de RSL au niveau local (en Algérie) et niveau de certain pays étrangers. Après cela, un autre contexte sera présenté pour donner l'organisation générale du système d'un RSL.

I.2 Définitions

Une définition des principaux termes auxquels nous aurons éventuellement recours dans notre étude est donnée dans ce qui suit :

I.2.1 Actionneur

Appareil ou organe permettant d'agir sur une machine ou un processus en vue de modifier son comportement ou son état [03].

I.2.2 Capteur

Dispositif permettant de capter un phénomène physique et de le restituer sous forme de signal [04].

I.2.3 ligne

Un point qui se déplace engendre une ligne. Une ligne n'a qu'une seule dimension, soit la longueur [05].

I.2.4 Courbe (ligne courbe)

Une ligne est courbe lorsqu'un point change constamment de direction [06].

I.2.5 Robot

Le Petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable [07].

I.2.6 Robot suiveur de ligne

C'est un appareil programmé selon certaines conditions dont la fonction d'usage est le suivi d'une trajectoire bien déterminée.

I.3 Avantages et inconvénients

➤ Un robot suiveur de ligne a une autonomie de plusieurs heures tandis qu'un employé prendrait des pauses.

Comme toute installation à base d'une technologie intelligente récente, le robot suiveur de ligne compte un certain nombre d'inconvénients [08] :

➤ Ce type de robots a un peu d'impact sur la société car ils remplacent les vrais employés (serveurs par exemple) ce qui augmente le taux de chômage,

➤ ce robot n'existe pas dans tous les domaines (cas des industries par exemple) car il reste assez coûteux (entre 4000 à 6000 euros) et son utilisation reste très limitée dans le monde.

➤ Aspect «énergies» peu présent.

➤ Réglages délicats.

Donc, le principe de fonctionnement d'un suiveur de ligne à base d'une technologie

I.4 Principe de fonctionnement d'un RSL

Le principe de fonctionnement du robot c'est qu'il devra suivre une ligne noire sur un sol blanc ou bien une ligne blanche sur un sol noir, on peut le faire fonctionner en modifiant les connecteurs sur un fond sombre avec une ligne blanche.

On suppose que le robot avance dans un sens que pour lui il y a la droite et la gauche.

Si le moteur a tendance à dévier trop à gauche, il faut que les moteurs compensent pour le faire aller plus à droite et inversement. Pour effectuer cette action, le choix d'utilisation est porté sur deux capteurs "suiveurs de ligne" composés chacun d'une photorésistance. Le but en fait, c'est de garder la ligne noire bien au centre du robot entre les 2 capteurs.

intelligente s'articule sur trois éléments suivants :

| Matériel | Algorithmique | Logiciel | Mode de transmission |
|--|---|----------|----------------------|
| ✓ | Partie logiciel : les procédures de programmation | | |
| • Unité de traitement | • Algorithme | | • Système embarqué |
| • Composants électroniques | • Organigramme | | |
| • Les capteurs | • Programme | | |
| • Les pré-actionneurs et les Actionneurs | • Langage | | |

Tableau I.1 Les éléments de base pour le fonctionnement d'un robot suiveur de ligne.

I.5 Domaines d'application

Plusieurs domaines sont touchés par ce type de technologie intelligente, parmi lesquels nous citons : la sécurité routière, les services publics (restauration), l'industrie.

➤ **Au service de l'industrie** : les robots suiveur de ligne dans l'industrie permettent le transport de lourdes charges d'un point A à un point B et ce de manière rapide et sûre.

➤ **La sécurité routière** : un véhicule suiveur de ligne est exploité dans les transports en commun dans le but d'éviter au maximum les accidents.

Par exemple, le cas de transport (est-ouest rouennais) de Rouen est un réseau de bus à haut niveau de service. Ce ne sont pas des bus sans chauffeurs comme dans certaines agglomérations, mais à l'approche d'un arrêt, le chauffeur laisse la main au système, qui permet au bus un accostage précis et régulier de la station (99,98 % d'efficacité en 2008) [09].

➤ **Au service des restaurants** : ce concept fait sensation auprès de certains pays (USA, Chine, Japon) pour améliorer les services auprès des clients [10].

➤ **Les jeux pour enfants** : un concept dans lequel l'enfant dessine lui-même le chemin qu'il veut que la voiture suive grâce à un feutre noir sur un fond clair [09].

➤ **Tournois et concours** : Le *concours de robot Suiveur de ligne* consiste en une course de vitesse entre des *robots* autonomes sur une piste fermée.

Par exemple, Pas moins de 50 participants représentant une quarantaine de wilayas du pays (Algérie) ont pris part au concours national de robotique et de l'intelligence artificielle (OROBOT) qui a été prévu en 2016 à Oran [11].

Il s'agit de la catégorie des robots solveurs (suiveurs de ligne) pour les amateurs, celle de robots Sumo (combats) et enfin celle du "Labyrinthe line", celui qui sort le premier d'un labyrinthe.

Quelques exemples courants sont présentés dans les figures suivantes :



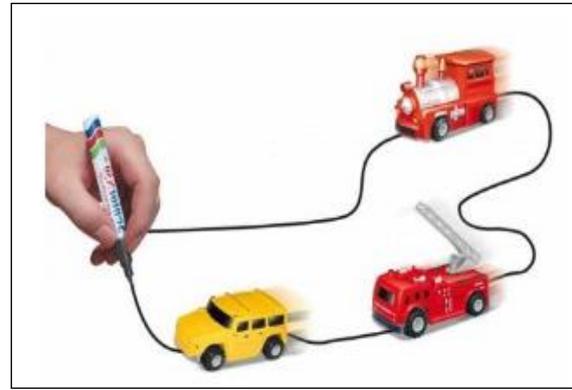
(a)



(b)



(c)



(d)



(e)

Figure I.1 : Robot suiveur de ligne

(a) Transport de charge dans l'industrie / (b) Marquage lu et interprété par le bus suiveur

(c) Robots aux sévices des restaurants/ (d) Jeux pour enfants

(e) 5ème concours national de robotique et intelligence artificielle. Oran 2016.

I.6 Organisation générale d'un robot suiveur

Pour commander intelligemment un robot suiveur de ligne, il faudra mettre en place un système à base d'un circuit programmable. Un tel système est toujours constitué des mêmes éléments, quelque soit la technologie utilisée.

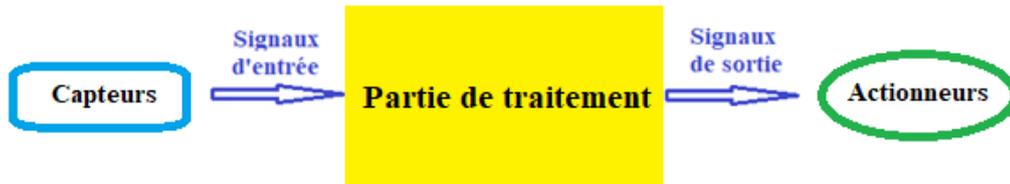


Figure I.2 Représentation schématique d'une organisation générale d'un suiveur de ligne

Il est à noter que, certains de ces capteurs délivrent des tensions très faibles, de l'ordre du

I.6.1 Les capteurs

Ils sont de types très différents, et la plupart d'entre eux délivrent des tensions ou des courants analogiques [12].

Il faut prendre des précautions spéciales pour que leur connexion au circuit de traitement (par exemple, Arduino) n'introduise pas d'erreurs inacceptables.

La tendance actuelle en matière de capteurs consiste à utiliser des techniques numériques. La figure suivante illustre un exemple de cas de circuits introduits dans les applications des suiveurs de lignes.



Figure I.3 Liste de quelques circuits d'entrées

Est un système (ensemble de circuits électroniques) qui génèrent des signaux destinés au

- **Les signaux d'entrées**

Sont des signaux envoyés par les capteurs dont les valeurs sont imposées à la partie de traitement.

- **La partie de traitement**

fonctionnement de différents actionneurs. A travers la figure **Figure I.4** nous donnons un exemple de cartes de développements électroniques à base de microcontrôleurs. Celles-ci permettent d'exécuter des tâches en prenant les entrées de composants électroniques externes (capteurs) et de générer des signaux à leurs sorties. Le cas détaillée sur les types de cartes Arduino est étudié au chapitre 2.



Figure I.4 Exemple de cartes de développements électroniques à base de microcontrôleurs

- **Les signaux de sorties**

Représentent l'activité utile de la partie de traitement. Ils en sont les seuls points observables, et ce sont eux qui permettent d'apprécier l'état de la partie de traitement.

- **Les actionneurs**

Les actionneurs servent à convertir un signal de commande généré par la partie de traitement en une action effective au niveau du système commandé.

Les actionneurs sont de types très divers et peuvent être des LEDs, des moteurs, ...La **Figure I.5** illustre quelques circuits les plus utilisés.

Les actionneurs nécessitent en général une commande de puissance, et il est donc nécessaire de disposer circuits de puissance à l'interface entre le système informatique et les circuits de sorties [12].



Figure I.5 Liste de quelques circuits de sorties

I.7 Conclusion

À travers le présent chapitre, nous avons donné quelques définitions, les avantages et inconvénients d'un RSL ainsi que son organisation sous l'aspect le plus général. Comme nous avons constaté que le taux de pénétration des RSL est presque négligeable en Algérie par rapport à certains pays quoique leur exploitation de ce type de robot compte aussi faible à travers le monde

Dans le prochain chapitre, nous allons voir une synthèse portant, sur la carte Arduino, ses types et les caractéristiques des cartes spécialement choisies pour notre travail.

Chapitre II

**Présentation de cartes de
développement à microcontrôleur**

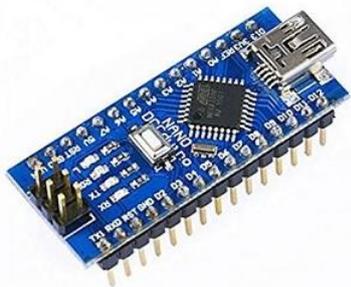
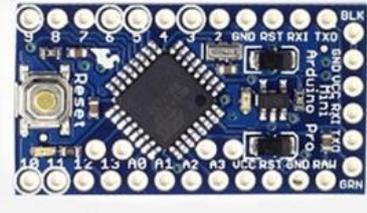
II.1 Introduction

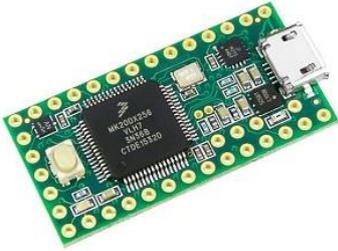
La commande des circuits électroniques à base des cartes à microcontrôleur est de plus en plus utilisée ces dernières années. Donc, différentes applications pour divers domaines ont recours à une multitude de cartes de développement, soit pour une autonomie de fonctionnement destiné à des tâches bien précises, soit comme dispositifs d'interface entre les circuits d'entrées et ceux de sorties pour une meilleure efficacité de commande.

Ainsi, différentes cartes sont présentées dans le présent chapitre. Aussi, une autre intention est portée sur les caractéristiques de certaines cartes de type ARDUINO le plus communément utilisé. Encore, le principe d'utilisation de la carte ARDUINO UNO est bien particulièrement souligné. Donc, un choix sera porté sur une carte à microcontrôleur afin de l'appliquer dans le cas du système à étudier de ce projet.

II.2 Carte à microcontrôleur pour système embarqué

De nombreux modèles de cartes à microcontrôleur existent sur le marché.

| Cartes microcontrôleur pour systèmes embarqués | |
|--|--|
| Carte Arduino Permet aux utilisateurs de créer des applications électroniques interactifs à partir de cartes électroniques matériellement libres sur lesquelles se trouve un microcontrôleur [13]. | Nano 12 € 1823,15 DA  |
| | Mini Pro 6,58 € 1000 DA  |
| | UNO 25 € 3798,02 DA  |

| | | |
|---|--|---|
| | <p>UNO WiFi</p> <p>40,20 € 6107,54 DA</p> |  |
| | <p>Yen2</p> <p>58,80 € 8933,43 DA</p> |  |
| | <p>Mega</p> <p>35 € 5 317,22 DA</p> |  |
| <p>Carte ChipKIT</p> <p>Spécialement conçu pour les Raspberry Pi sur la base du microcontrôleur 32 bits [14].</p> | <p>WI-FIRE</p> <p>97.83 € 14863,21 DA</p> |  |
| <p>Carte Teensy</p> <p>est un système complet de développement de microcontrôleurs basé sur USB, dans un très faible encombrement, capable de mettre en œuvre de nombreux types de projets. Toute la programmation se fait via le port USB [14].</p> | <p>Teensy 3.2</p> <p>33,65 € 5112,41 DA</p> |  |
| <p>Carte Raspberry Pi</p> <p>Elle est destinée à encourager l'apprentissage de la programmation informatique [14].</p> | <p>3B- 1GB</p> <p>72,40 € 11000 DA</p> |  |

| | | |
|--|--|---|
| | <p>4 B – 4 GB</p> <p>111,89 € 17000DA</p> |  |
| <p>Carte LattePanda</p> <p>est un ordinateur monocarte compatible Arduino, fonctionnant sous Windows 10 ou Linux [14].</p> | <p>DFR04182-GB/32GB</p> <p>152.50 € 23169,17 DA</p> |  |
| <p>Carte ArbotiX-M</p> <p>est un système de contrôle à distance évolué pour des robots de petites et moyennes tailles [15].</p> | <p>ArbotiX-M</p> <p>51,90 € 7885,12 DA</p> |  |

Tableau II.1 Quelques types de cartes à microcontrôleur pour les systèmes embarqués

Une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un Généralement, les cartes à microcontrôleurs se caractérisent par [16]:

- ✓ Un plus haut degré d'intégration,
- ✓ Une plus faible consommation électrique,
- gigahertz),
- ✓ Un coût réduit par rapport aux microprocesseurs utilisés dans les ordinateurs personnels.

Les systèmes intelligents sont en augmentation dans tous les domaines de la vie quotidienne. Voici des exemples d'utilisation des microcontrôleurs :

- **Télécommunications** : cartes FAX et MODEM, Minitel, téléphones portables (interfaces homme machine, gestion d'écrans graphiques)...



Figure II.1 *Domaines de télécommunication intelligente*
(a) *Antenne intelligente* (b) *Téléphone portable intelligent.*

Industriels : automates programmables, contrôle de processus divers, supervision...



Figure II.2 *Industrie intelligente.*

- **Commercial** : électroménager, domotique...

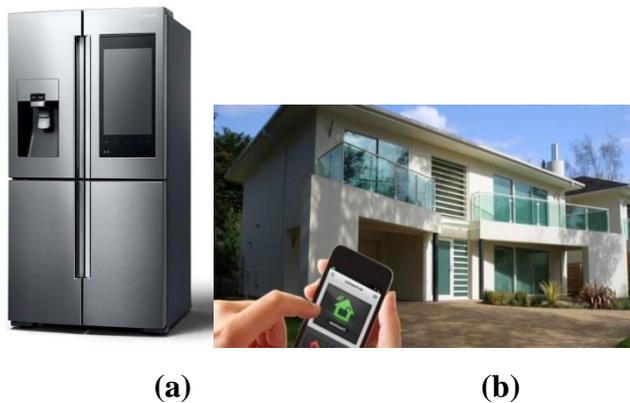


Figure II.3 *Utilisation des microcontrôleurs dans le commercial*
(a) *Réfrigérateur intelligent* (b) *Maison intelligente.*

- **Automobile** : ABS, tableau de bord, contrôle des sièges, des vitres...



Figure II.4 Voiture intelligente

- **Militaire et spatial** : sonde, lanceurs de fusées, missile, robots...



Figure II.5 Les armes intelligentes

- **Domaine scientifique** : Contrôle des sorties Arduino avec le PC. Dans ce cas, il y a une communication directe entre des plateformes externes et ARDUINO, à savoir Matlab, Visual basic, Delphi ...etc. La figure suivante illustre le cas d'asservissement de vitesse d'un moteur à courant continu [17].

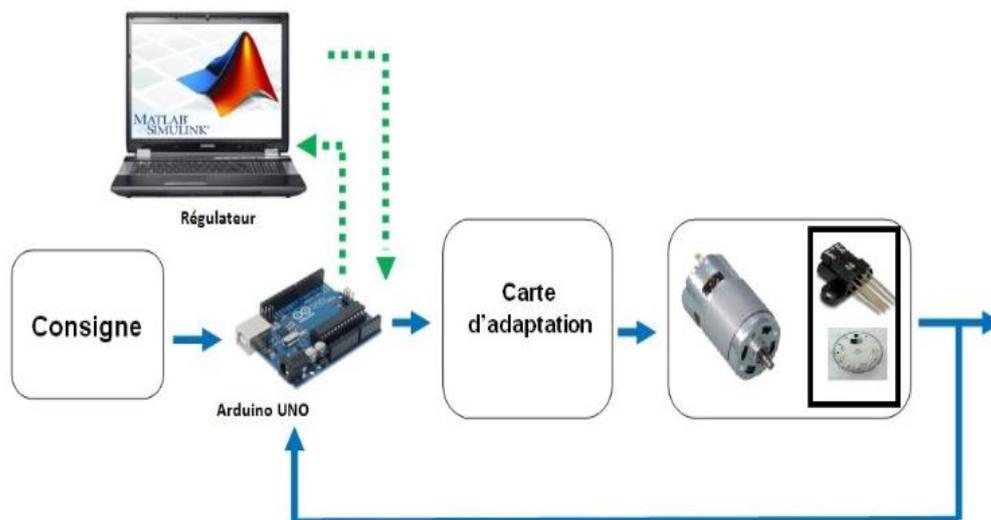


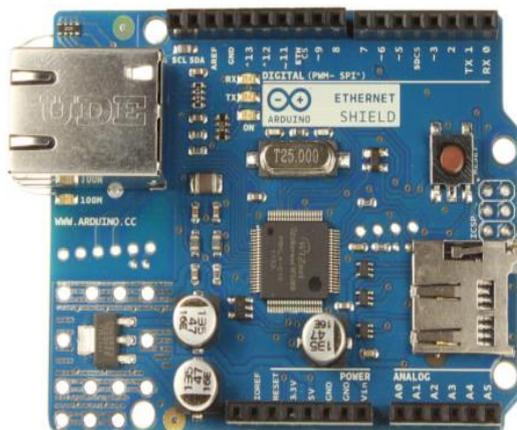
Figure II.6 Communication entre matlab et ARDUINO

II.3 Cartes à microcontrôleur de type Arduino

- En 2004 : l'Arduino UNO a été développé par des enseignants et élèves d'une école de Design italienne, qui avaient pour vocation de démocratiser la programmation et de la rendre plus libre d'accès.

Les cartes à microcontrôleur de type ARDUINO sont disponibles avec de nombreux types de modules qui peuvent être reliés avec celui-ci. Ces modules sont connus sous le nom de « Shield ». Certains des shield les plus couramment utilisés sont:

- **Shields** Moteurs et servos : interface dédiée au pilotage des moteurs
 - **Shields** Afficheurs : permet de contrôler **un afficheur**
 - **Shields** Relais : pour actionner des dispositifs
 - **Shields** Ethernet : **pour** connecter un projet Arduino à Internet par le biais d'un câble RJ45
 - **Shields** WIFI : permettant de connecter la carte à Internet
 - **Shields** GSM : permet d'envoyer des SMS, MMS, GPRS et d'établir des communications audios.
 - **Shields** Bluetooth : permet de commander un dispositif branché sur ARDUINO
- Exemple de cas de shields



(a)



(b)

Figure II.7 Shields pour carte ARDUINO

(a) Shield pour connexion à internet (b) Shield à relais isolés galvaniquement

II.3.1 Caractéristiques des cartes de type Arduino

| Caractéristiques | Carte ARDUINO | | | | | |
|--------------------------------------|-----------------|---------------|-----------------|-----------------|--------------|------------------|
| | Nano | Mini Pro | UNO | Leonardo | Yen | Méga |
| Microprocesseur | ATMega328 | ATmega328 | ATmega328 | ATmega32U4 | ATmega32U4 | ATMega2560 |
| Mémoire flash | 32 ko | 32 Ko | 32 ko | 32 ko | 32 ko | 256 Ko |
| Mémoire SRAM | 2 ko | 2 Ko | 2 ko | 2,5 ko | 2,5 ko | 8 Ko |
| Mémoire EEPROM | 1 ko | 1 Ko | 1 Ko | 1 Ko | 1 ko | 4 Ko |
| broches d'entrées/ sorties | 20 | 14 | 14 | 20 | 20 | 54 |
| broches PWM | 6 | 6 | 6 | 7 | 7 | 14 |
| broches d'entrées analogiques | 6 | 6 | 6 | 12 | 12 | 16 |
| Courant par entrées-sorties | 40 mA | 40 mA | 40 mA | 40 mA | 50 mA | 40 mA |
| Fréquence d'horloge | 16 MHz | 16Mhz | 16Mhz | 16Mhz | 16Mhz | 16 MHz |
| Prise USB | mini-USB B | - | USB | USB | USB | USB |
| Dimensions | 45 x 18 x 18 mm | 33.8mm x 18mm | 6.86cm x 5,3 cm | 6.86cm x 5,3 cm | 7cm x 5,3 cm | 101,52 x 53,3 mm |
| Poids | 5 g | 2g | 25g | 20g | 41g | 37 g |
| Tension de fonctionnement | 5V | 7V – 9V | 5V | 5V | 5V | 5 V |

Tableau II.2 Caractéristiques de quelques cartes de développement de type Arduino les plus utilisées

II.3.2 Avantages et inconvénients

Dans cette partie, nous passons en revue quelques différentes cartes d'Arduino en mettant l'accent principalement sur leurs avantages et leurs inconvénients, **Tableau II.3**

Généralement, le principal avantage d'une carte à microprocesseur est le traitement rapide et une interface facile.

| Type de carte Arduino | Avantages | Inconvénients |
|-----------------------|---|--|
| Uno | <ul style="list-style-type: none"> • Nombreux exemples de montage sont disponibles sur internet. • Nombres suffisant de broches d'E/S pour des projets élémentaires. • Vaste choix de shields • Bon marché | <ul style="list-style-type: none"> • Nombre de broche insuffisant pour les gros projets • La mémoire disponible risque d'être un peu juste pour projets ambitieux. • Ne peut pas être utilisée comme hôte USB pour simuler périphérique tel qu'un clavier ou une souris. • Acquiert des modules externes qui augmenteront inévitablement le coût total d'achat |
| Nano | <ul style="list-style-type: none"> • Encombrement réduit • Possibilité d'être enfichée directement sur la plaque d'essai | <ul style="list-style-type: none"> • Il n'est pas possible d'utiliser des shields |
| Mini Pro | <ul style="list-style-type: none"> • Taille réduite et son faible coût • Fournie sans connecteurs pour être facilement intégrée à tout projet final. | <ul style="list-style-type: none"> • Ne possède de convertisseur USB / série. • Ne possède pas non plus de sortie 3.3V (disponible sur les autres cartes) • Conçu pour les applications où l'espace est très petit • Conçu pour être intégré en permanence dans un projet. • Destiné à un public plus |

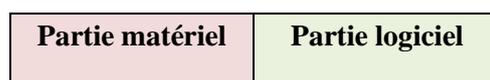
| | |
|-------------|---|
| | expérimenté. |
| Mega | <ul style="list-style-type: none"> • Nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs. • Capacité de mémoire suffisante pour les gros projets. • Plus de broches UART (4ports de communication série) • Plus de broches MLI(15sorties numériques peuvent être utilisées comme MLI. • Compatible avec la plupart des shields conçues pour Arduino |
| Yun | <ul style="list-style-type: none"> • Module wifi préinstallé • Interface Ethernet préinstallé. • Nombreuses possibilités d'extension. <ul style="list-style-type: none"> • Deux fois plus chère que l'Arduino Uno • Consomme plus d'énergie de sorte que le port USB 2.0 peut vite atteindre ses limites. |

Tableau II.3 *Avantages et Inconvénients de différentes cartes de type Arduino.*

II.4 Éléments principaux d'une carte de développement Arduino

Une carte Arduino peut être classée en deux parties :

Carte Arduino



II.4.1 Partie Matériel

La carte de développement Arduino se compose de nombreux composants. Dans ce cas, nous allons nous appuyer sur le type de carte « Arduino UNO » pour montrer les principaux blocs de composants qui aident à son fonctionnement [18] :

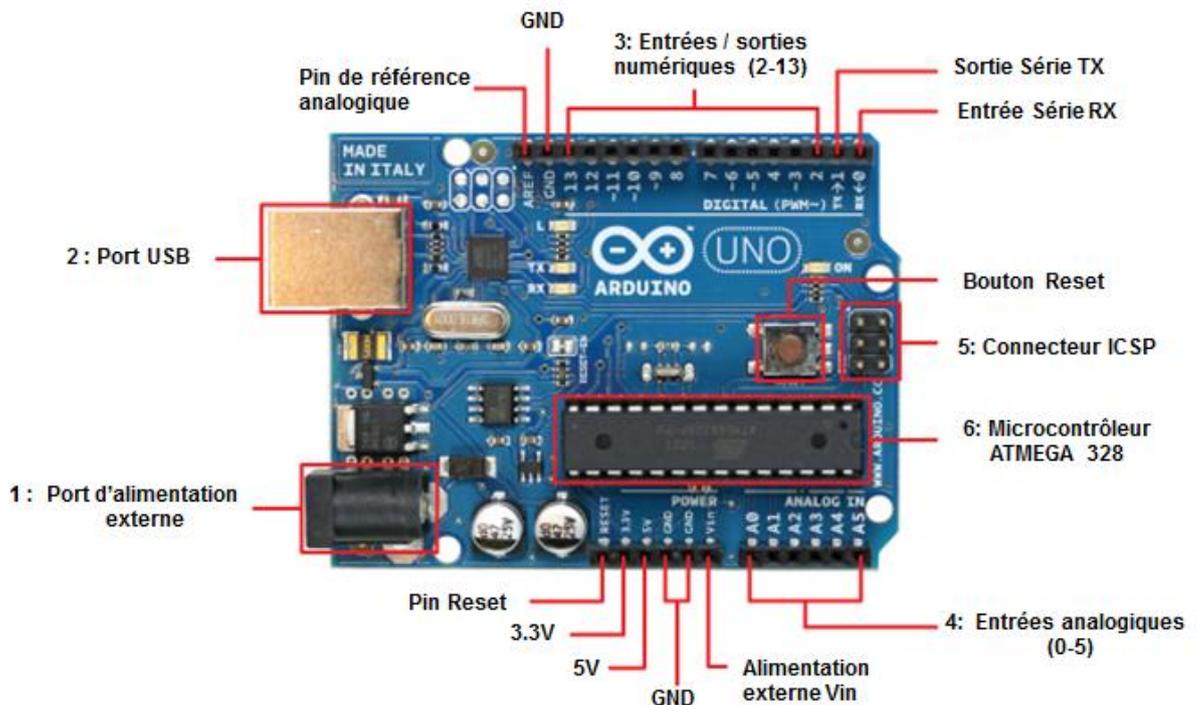


Figure II.8 Brochage d'une carte Arduino de type UNO.

➤ Description

1 : Le port d'alimentation externe pour fonctionner, la carte a besoin d'une alimentation qui est conseillée en général entre 7 V et 12V pour garder une marge en basse tension et éviter que le circuit ne chauffe trop. Cette tension doit être continue et peut par exemple être fournie par une pile 9V.

2 : Le port USB Permet de communiquer avec la carte et de l'alimenter en 5V.

3: Les entrées/sorties numériques :

- 4 entrées/sorties numériques
- 6 (**broches 4, 5, 6, 7, 8, 9**) peuvent assurer une sortie PWM, et peuvent actionner de nombreux composants (LED, transistor, etc.) mais elles ne peuvent pas fournir beaucoup de courant (40 mA pour une carte Arduino UNO). Pour piloter des circuits de plus forte puissance, il faut passer par des transistors ou des relais.

4: Les entrées analogiques lui permettent de mesurer une tension variable (entre 0 et 5 V) qui peut provenir de capteurs ou d'interfaces divers (potentiomètres, etc).

5: Connecteur In-Circuit Serial Programming ICSP pour le téléchargement du programme.

6: Microcontrôleur ATmega328 est un microcontrôleur **ATMEL** de la famille **AVR** 8bits. Pour son brochage, voir la partie annexe.

II.4.2 Partie logiciel

Le code de programme écrit pour Arduino est connu sous le nom de croquis (sketch). Le logiciel utilisé pour développer de tels croquis pour un Arduino est communément appelé Arduino IDE. Cet IDE contient les parties suivantes [19]. L'ensemble de ces points seront traités et vus dans le chapitre 4 :

| | | |
|--------------------|-------------------------------------|---|
| ARDUINO IDE | Éditeur de texte | C'est là que le code simplifié peut être écrit à l'aide d'une version simplifiée du langage de programmation C ++. |
| | Zone de message | Il affiche une erreur et donne également un retour sur l'enregistrement et l'exportation du code. |
| | Texte | La console affiche la sortie de texte par l'environnement Arduino, y compris les messages d'erreur complets et d'autres informations. |
| | Barre d'outils de la console | Cette barre d'outils contient divers boutons tels que Vérifier, Télécharger, Nouveau, Ouvrir, Enregistrer et Serial Monitor... |

Tableau II.4 Les parties de Arduino IDE.

➤ Principe d'utilisation d'une carte de développement ARDUINO

Cette partie explore comment une carte ARDUINO peut être utilisée comme un outil pour les travaux d'étude et de recherche [20]. De ce fait, la communication entre le PC et la carte se fait via le port USB, moyennant installation d'un driver adapté (fourni par ARDUINO).



Figure II.9 Schéma de téléversement.

▪ Les étapes de programmation ARDUINO

Après la préparation du programme, il y'a des étapes à faire afin de terminer la programmation, qui sont les suivantes :

- ✓ Obtention d'une carte ARDUINO et un câble USB
- ✓ Téléchargement de l'environnement ARDUINO
- ✓ Raccordement de la carte ARDUINO au PC
- ✓ Installation des pilotes du périphérique Série-USB
- ✓ Lancement de l'application ARDUINO

▪ Chargement du programme dans la carte

Etape 1 : lancement du logiciel.



Etape 2 : ouverture et modification du programme.

Etape 4 : connexion de la carte au PC avec le cordon USB.

Etape 5 : transfert du programme vers la carte.



Etape 6 : vérification du fonctionnement.

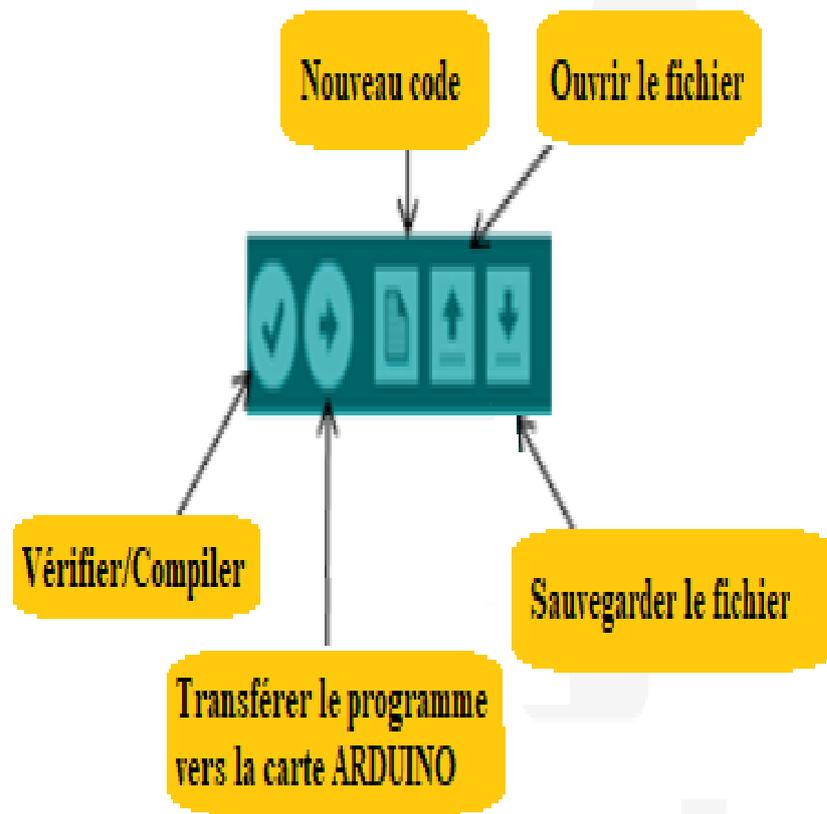


Figure II.10 Schéma du logiciel.

▪ **Structure d'un projet ARDUINO**

Un programme ARDUINO est composé de 3 parties:

Partie1 : déclaration des variables (optionnelle)

Partie2 : représentée par la **Fonction setup ()**. C'est une partie initialisation et configuration des entrées / sorties. Elle est appelée une seule fois lorsque le programme commence.

Partie3 : représentée par la **Fonction loop ()**. C'est la partie principale contenant le programme. Elle est répétée indéfiniment en boucle infinie

▪ Notions du langage ARDUINO

Structure

Fonctions de base

- `void setup()`
- `void loop()`

Structures de contrôle

- `if`
- `if...else`
- `for`
- `switch case`
- `while`
- `do... while`
- `break`
- `continue`
- `return`
- `goto`

Syntaxe de base

- `;` (point virgule)
- `{}` (accolades)
- `//` (commentaire sur une ligne)
- `/* */` (commentaire sur plusieurs lignes)
- `#define`
- `#include`

Opérateurs arithmétiques

- `=` (égalité)
- `+` (addition)
- `-` (soustraction)
- `*` (multiplication)
- `/` (division)
- `%` (modulo)

Opérateurs de comparaison

- `==` (égal à)
- `!=` (différent de)
- `<` (inférieur à)
- `>` (supérieur à)
- `<=` (inférieur ou égal à)
- `>=` (supérieur ou égal à)

Opérateurs booléens

- `&&` (ET booléen)
- `||` (OU booléen)
- `!` (NON booléen)

Opérateurs composés

- `++` (incréméntation)
- `--` (décréméntation) (à revoir)
- `+=` (addition composée)
- `-=` (soustraction composée)
- `*=` (multiplication composée)
- `/=` (division composée)
- `&=` (ET bit à bit composé)
- `|=` (OU bit à bit composé)

Opérateurs bit à bit

- `&` (ET bit à bit)
- `|` (OU bit à bit)
- `^` (OU EXCLUSIF bit à bit)
- `~` (NON bit à bit)
- `<<` (décalage à gauche)
- `>>` (décalage à droite)

Pointeurs

- `*` pointeur
- `&` pointeur

Fonctions

| | | |
|--|---|--|
| <p>Entrées/Sorties Numériques</p> <ul style="list-style-type: none"> • <code>pinMode(broche, mode)</code> • <code>digitalWrite(broche, valeur)</code> • <code>int digitalRead(broche)</code> <p>Entrées analogiques</p> <ul style="list-style-type: none"> • <code>int analogRead(broche)</code> • <code>analogReference(type)</code> <p>Sorties "analogiques" (génération d'impulsion)</p> <ul style="list-style-type: none"> • <code>analogWrite(broche, valeur)</code> - PWM <p>Entrées/Sorties Avancées</p> <ul style="list-style-type: none"> • <code>tone()</code> • <code>noTone()</code> • <code>shiftOut(broche, BrocheHorloge, OrdreBit, valeur)</code> • <code>unsigned long pulseIn(broche, valeur)</code> <p>Communication</p> <ul style="list-style-type: none"> • <code>Serial</code> | <p>Temps</p> <ul style="list-style-type: none"> • <code>unsigned long millis()</code> • <code>unsigned long micros()</code> • <code>delay(ms)</code> • <code>delayMicroseconds(us)</code> <p>Math</p> <ul style="list-style-type: none"> • <code>min(x, y)</code> • <code>max(x, y)</code> • <code>abs(x)</code> • <code>constrain(x, a, b)</code> • <code>map(valeur, toLow, fromHigh, toLow, toHigh)</code> • <code>pow(base, exposant)</code> • <code>sq(x)</code> • <code>sqrt(x)</code> <p>Pour davantage de fonctions mathématiques, voir aussi la librairie <code>math.h</code> : <code>log</code>, <code>log10</code>, <code>asin</code>, <code>atan</code>, <code>acos</code>, etc...</p> <p>Nombres randomisés (hasard)</p> <ul style="list-style-type: none"> • <code>randomSeed(seed)</code> • <code>long random(max)</code> • <code>long random(min, max)</code> | <p>Trigonométrie</p> <ul style="list-style-type: none"> • <code>sin(rad)</code> • <code>cos(rad)</code> • <code>tan(rad)</code> <p>Bits et Octets</p> <ul style="list-style-type: none"> • <code>lowByte()</code> • <code>highByte()</code> • <code>bitRead()</code> • <code>bitWrite()</code> • <code>bitSet()</code> • <code>bitClear()</code> • <code>bit()</code> <p>Interruptions Externes</p> <ul style="list-style-type: none"> • <code>attachInterrupt(interrupti on, fonction, mode)</code> • <code>detachInterrupt(interrupt ion)</code> <p>Interruptions</p> <ul style="list-style-type: none"> • <code>interrupts()</code> • <code>noInterrupts()</code> |
|--|---|--|

Variables et constantes

| | |
|---|--|
| <p>Types des données</p> <p>Les variables peuvent être de type variés qui sont décrits cidessous.</p> <p>Synthèse des types de données Arduino</p> <ul style="list-style-type: none"> • <code>boolean</code> • <code>char</code> • <code>byte</code> • <code>int</code> • <code>unsigned int</code> • <code>long</code> • <code>unsigned long</code> • <code>float</code> (nombres à virgules) • <code>double</code> (nombres à virgules) • Les chaînes de caractères • objet <code>String</code> NEW • Les tableaux de variables • le mot-clé <code>void</code> (fonctions) • <code>word</code> • <code>PROGMEM</code> | <p>Conversion des types de données</p> <ul style="list-style-type: none"> • <code>char()</code> • <code>byte()</code> • <code>int()</code> • <code>long()</code> • <code>float()</code> • <code>word()</code> <p>Portée des variables et qualificateurs</p> <ul style="list-style-type: none"> • Portée des variables • <code>static</code> • <code>volatile</code> • <code>const</code> <p>Utilitaires</p> <ul style="list-style-type: none"> • <code>sizeof()</code> (opérateur <code>sizeof</code>) <p>Référence</p> <ul style="list-style-type: none"> • Code ASCII |
|---|--|

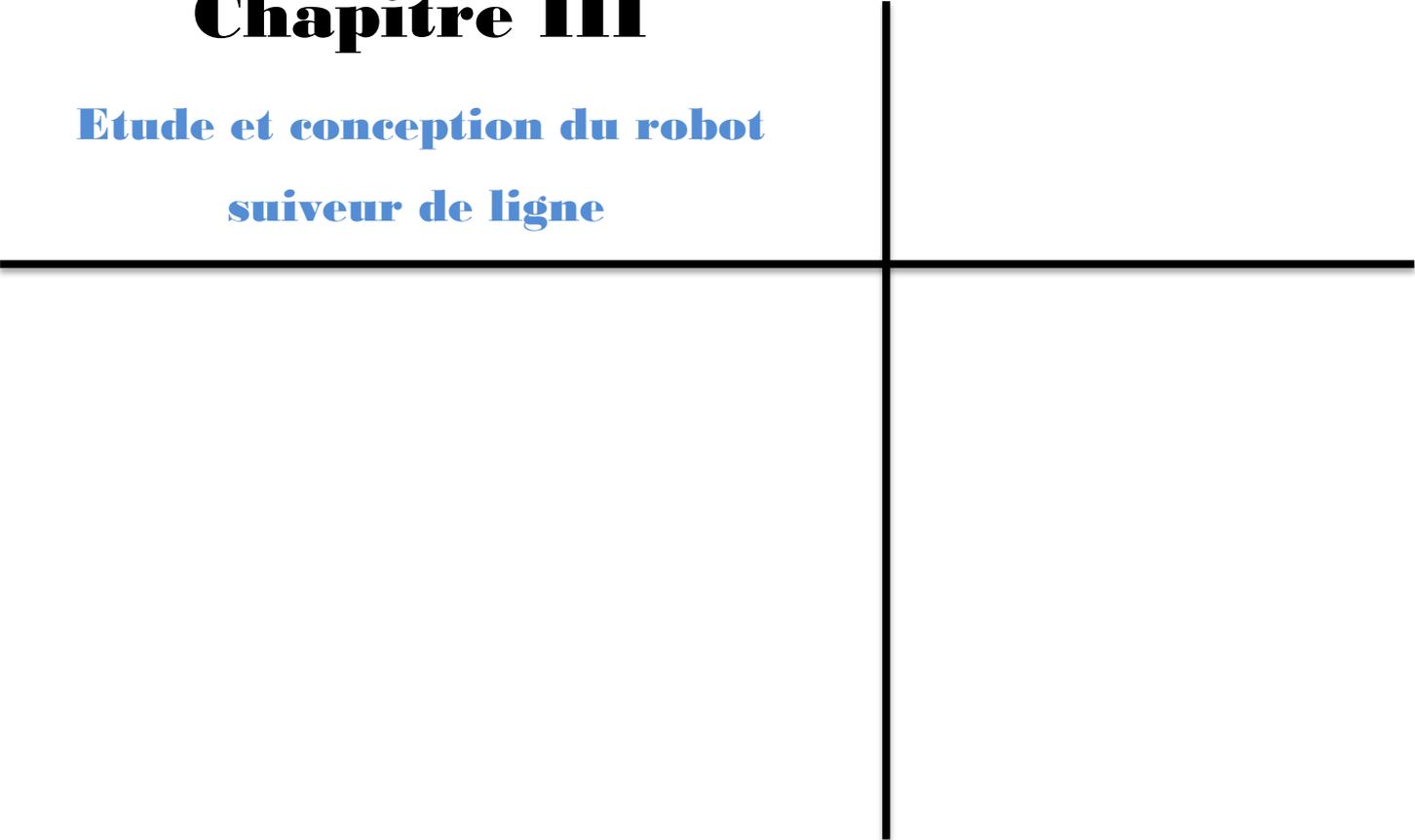
II.5 Conclusion

Dans le présent chapitre, nous avons vu un ensemble de cartes à microcontrôleur pouvant être appliqué dans les systèmes embarqués. A travers ces cartes, nous avons opté pour le choix d'utiliser la carte Arduino UNO et ce, pour un fonctionnement approprié non rencontré avec un système à base de composants électroniques uniquement.

Par conséquent, le chapitre suivant sera consacré à l'étude et la conception du système proposée dans ce travail pour le suivi d'une ligne bien déterminée. Le système à carte Arduino sera comparé à celui non concerné par la carte à travers différentes circuiteries afin de monter la nécessité d'utilisation des circuits programmés

Chapitre III

**Etude et conception du robot
suiveur de ligne**



III.1 Introduction

Dans ce chapitre, deux cas de situation seront traités, le premier est sans carte Arduino et le deuxième avec carte Arduino.

Le présent projet consiste à l'étude d'un robot suiveur de ligne avec détecteur d'obstacles avec la carte Arduino et suiveur de ligne sans carte Arduino et capable de se déplacer tout seul suivant une ligne définie.

Pour cela, le robot doit être doté d'une logique programmée (dans le deuxième cas) dont la partie commande conçue autour d'un microcontrôleur et utilisant deux batteries pour l'alimentation des moteurs et du microcontrôleur ainsi que d'autres composants du montage.

III.2 Etude des circuits de suiveur de ligne

Dans notre travail, deux types de circuits sont considérés : sans et avec carte Arduino. ce qui nous permettra de comparer les différents composants utilisés et faire ressortir les avantages et les inconvénients entre les deux circuits étudiés.

III.2.1 Sans carte Arduino

III.2.1.1 Schéma synoptique

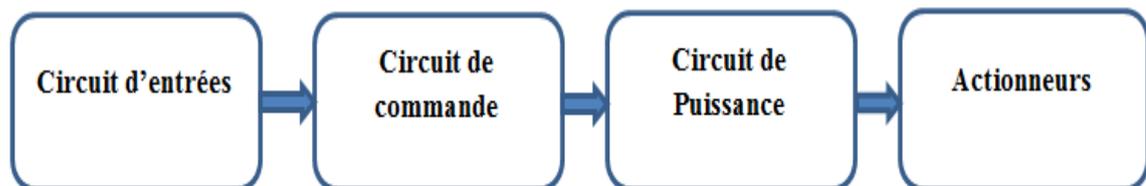


Figure III.1 Schéma synoptique du sans carte Arduino

III.2.1.2 Schéma électrique

Ici, je représente le circuit électronique global du système concerné **Figure III.2**. Ainsi, il est scindé en trois parties principales, ce qui permettra leur étude séparément.

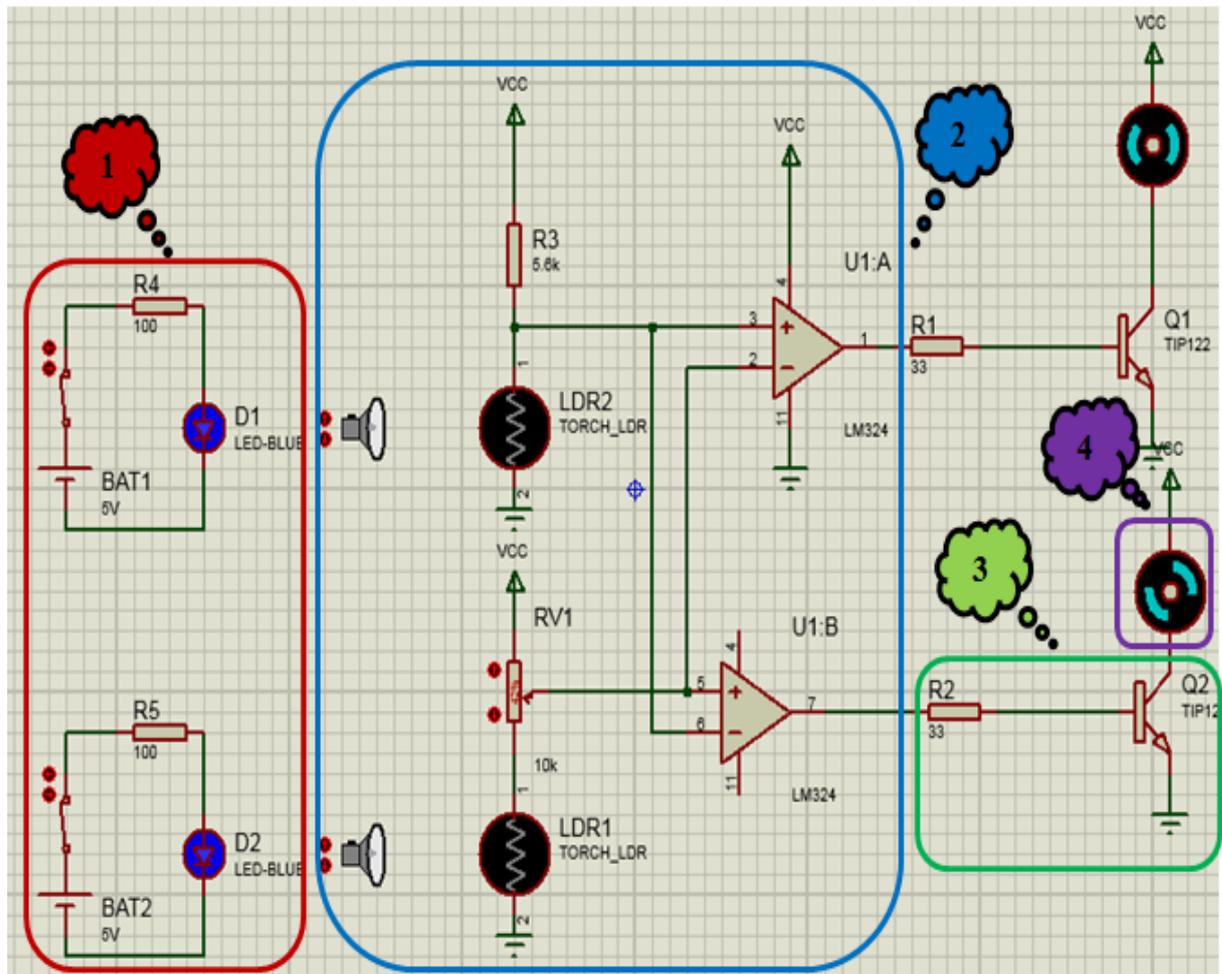


Figure III.2 Circuit électrique générale du robot Sans carte Arduino.

III.2.1.3 Principe de fonctionnement

Le principe de fonctionnement du robot c'est d'être dans ce cas sur un sol sombre avec une ligne blanche, on peut le faire fonctionner en modifiant les connexions sur un fond clair avec une ligne noire.

On suppose que le robot suiveur avance sur un sens, cela veut dire que pour lui il y a la droite et la gauche.

Si le robot a tendance à dévier trop à gauche il faut que les moteurs compensent pour le faire aller plus à droite et inversement. Et le but c'est de garder la ligne blanche bien au centre du robot entre les deux photorésistances au-dessous du robot.

Quand le robot dévie, la ligne blanche se rapproche de la photorésistance de droite donc la photorésistance devient plus éclairée et moins résistante et la tension a ses bornes va descendre donc la roue droite s'arrête et la roue gauche continue de se rouler donc le robot vire à droite.

III.2.1.4 Calcul des circuits

➤ Partie 1 : Circuit d'entrées

Cette partie est constituée des LED électroluminescentes et de résistance, comme montré par « 1 » sur le **Figure III.2**.

Donc cette partie de circuit est repris pour la démarche de calcul à effectuer sur le choix des composants concerné **Figure III.3**.

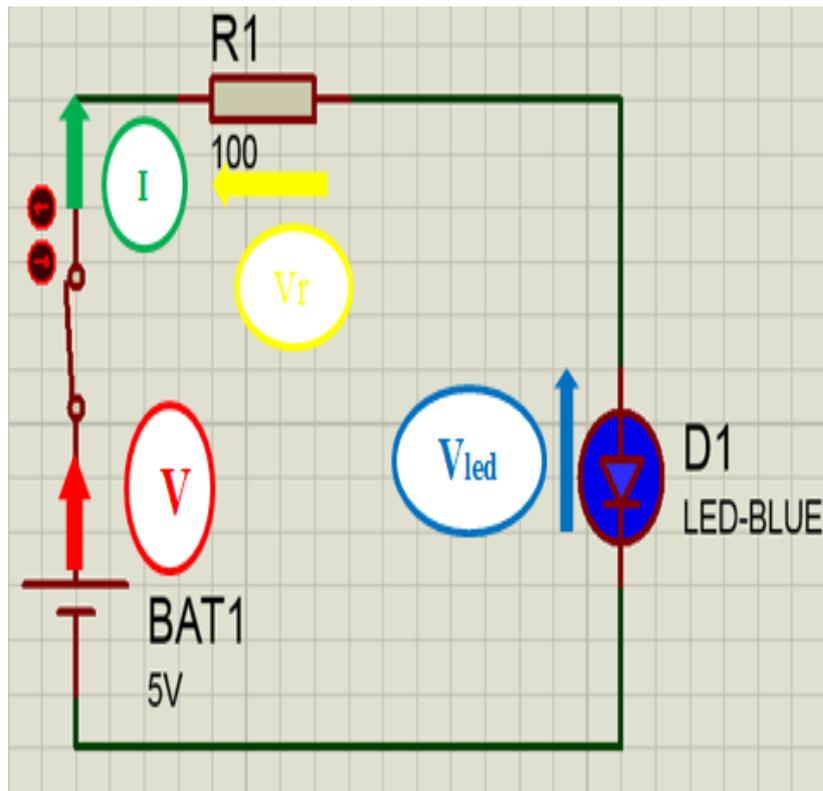


Figure III.3 Circuit électrique de la première partie (circuit d'entrées).

La LED bleu peut supporter que les faibles tensions entre 1.6V et 3.8V et un courant de 20mA.

On suppose que : $V_{led} = 3V$

$$V = V_r + V_{led}$$

$$V_r = V - V_{led} = 5 - 3 = 2V$$

$$V_r = R \times I$$

$$R = \frac{V_r}{I} = \frac{2}{0.02}$$

$$\rightarrow R = 100 \Omega$$

➤ **Partie 2 : Circuit de commande**

Cette partie est montrée sur la **Figure III.2**. Ainsi, pour le calcul, je vais considérer le premier comparateur **Figure III.4**.

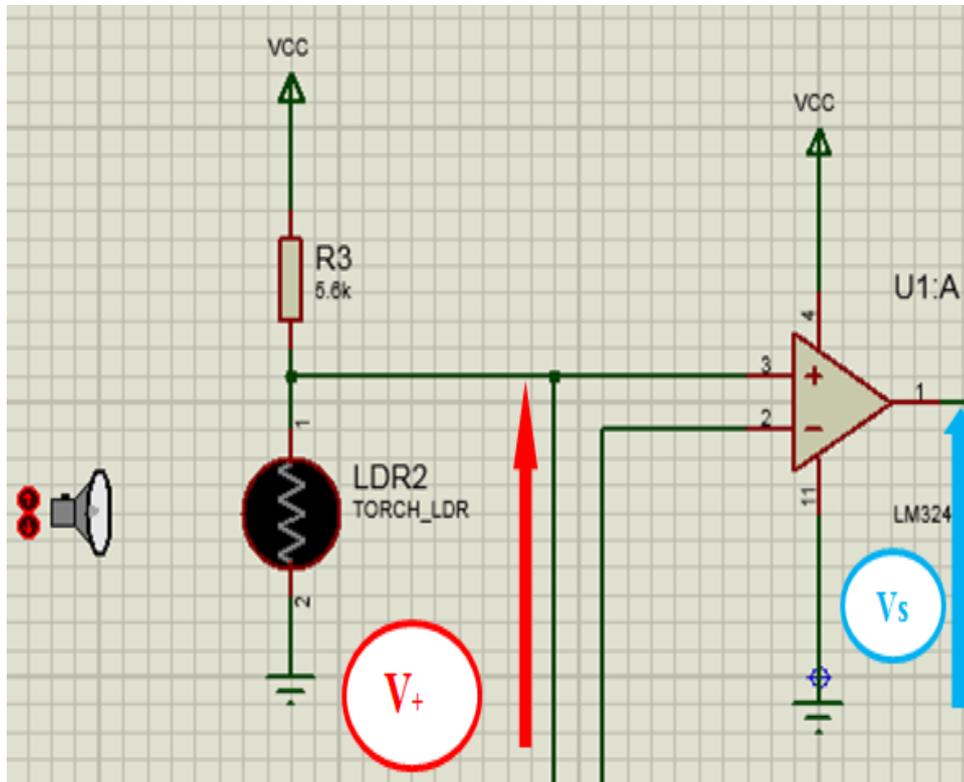


Figure III.4 Circuit électrique de la deuxième partie (circuit de commande)(a).

$$V_+ = V_{CC} \frac{R(\text{LDR})}{R(\text{LDR}) + R_3}$$

Dans ce cas

$$R_3 = \frac{V_{CC} - V_+}{V_+} R(\text{LDR})$$

R(LDR) 10 fois plus lumineux → $V_+ = 4.54$ Volts $10\text{k}\Omega \leq R(\text{LDR}) \leq 60\text{k}\Omega$

R(LDR) 10 fois moins lumineux → $V_+ = 0.54$ Volts $R(\text{LDR}) \geq 0.1\text{ M}\Omega$

Exemple de calculs

$$V_{CC} = 9 \text{ volts } R_3 = \frac{9 - 4.5}{4.5} \times 5.6\text{k}\Omega$$

$$\rightarrow R_3 = 5.6\text{k}\Omega$$

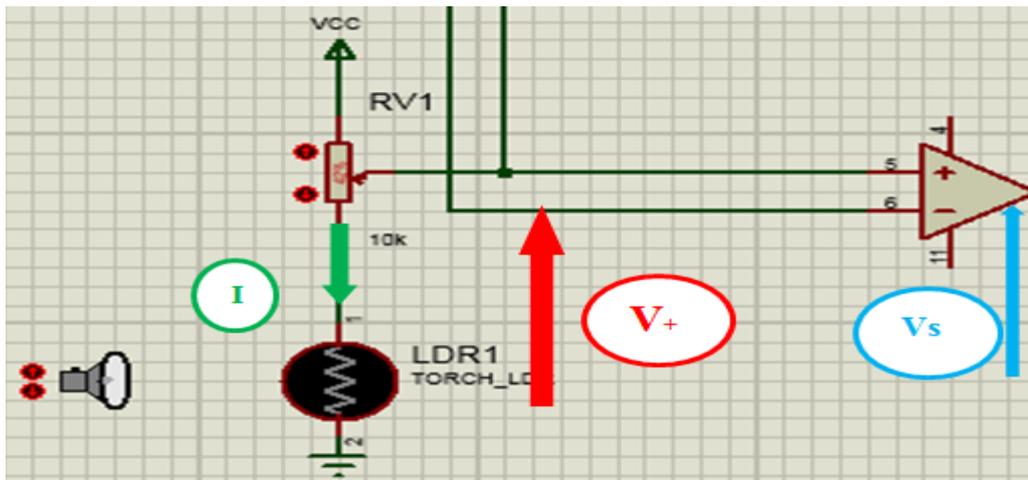


Figure III.5 Circuit électrique de la deuxième partie (b).

$$V+ = V_{cc} \frac{R(LDR)}{R(LDR)+RV1}$$

Dans ce cas

$$RV1 = \frac{V_{cc}-V+}{V+} R(LDR)$$

R(LDR) 10 fois plus lumineux $\rightarrow V+ = 4.54$ Volts $10k\Omega \leq R(LDR) \leq 60k\Omega$

R(LDR) 10 fois moins lumineux $\rightarrow V+ = 0.54$ Volts $R(LDR) \geq 0.1$ M Ω

Exemple de calculs

$$V_{cc} = 9 \text{ volts } RV1 = \frac{9-4.5}{4.5} \times 10k\Omega$$

$\rightarrow RV1 = 10k\Omega$

Partie 3 : Circuit puissance

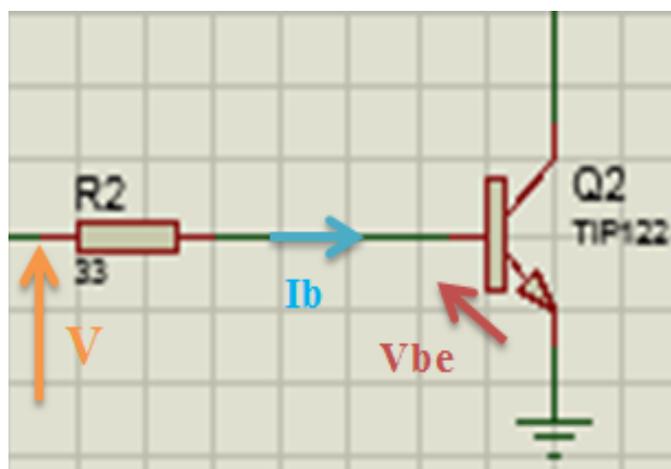


Figure III.6 Circuit électrique de la troisième partie (circuit de puissance).

I_b peut aller jusqu'à $I_b=120\text{mA}$

$$R = \frac{9 - V_{be}}{I_b} = \frac{9 - 1.2}{0.12 \text{ A}} = \frac{7.8}{0.12} = 65 \Omega.$$

Remarque au lieu de travaillée avec 65Ω on a travaillé avec 33Ω .

➤ Partie 4 : Actionneurs



Figure III.7 Moteur à courant continu.

Nous avons un seul actionneur dans ce système, le moteur à courant continu. Comme le dit son nom, ce moteur marche en courant continu et il convertit l'énergie électrique en énergie mécanique.

Les caractéristiques du moteur courant continu (DC) sont caractérisées par une constante de vitesse, et une pente vitesse/couple. Le courant est proportionnel à la charge ; et la vitesse est proportionnelle à la tension d'alimentation [21].

III.2.2 Avec carte Arduino

III.2.2.1 Schéma synoptique



Figure III.8 Schéma synoptique avec carte Arduino.

III.2.2.2 Schéma électrique

Ici, je représente le circuit électronique global du système concerné **Figure III.9** Ainsi, il est scindé en trois parties principales, ce qui permettra leur étude séparément. Nous remarquons l'utilisation d'une carte à microcontrôleur « Arduino », Donc, l'aspect logiciel est fondamental dans ce cas.

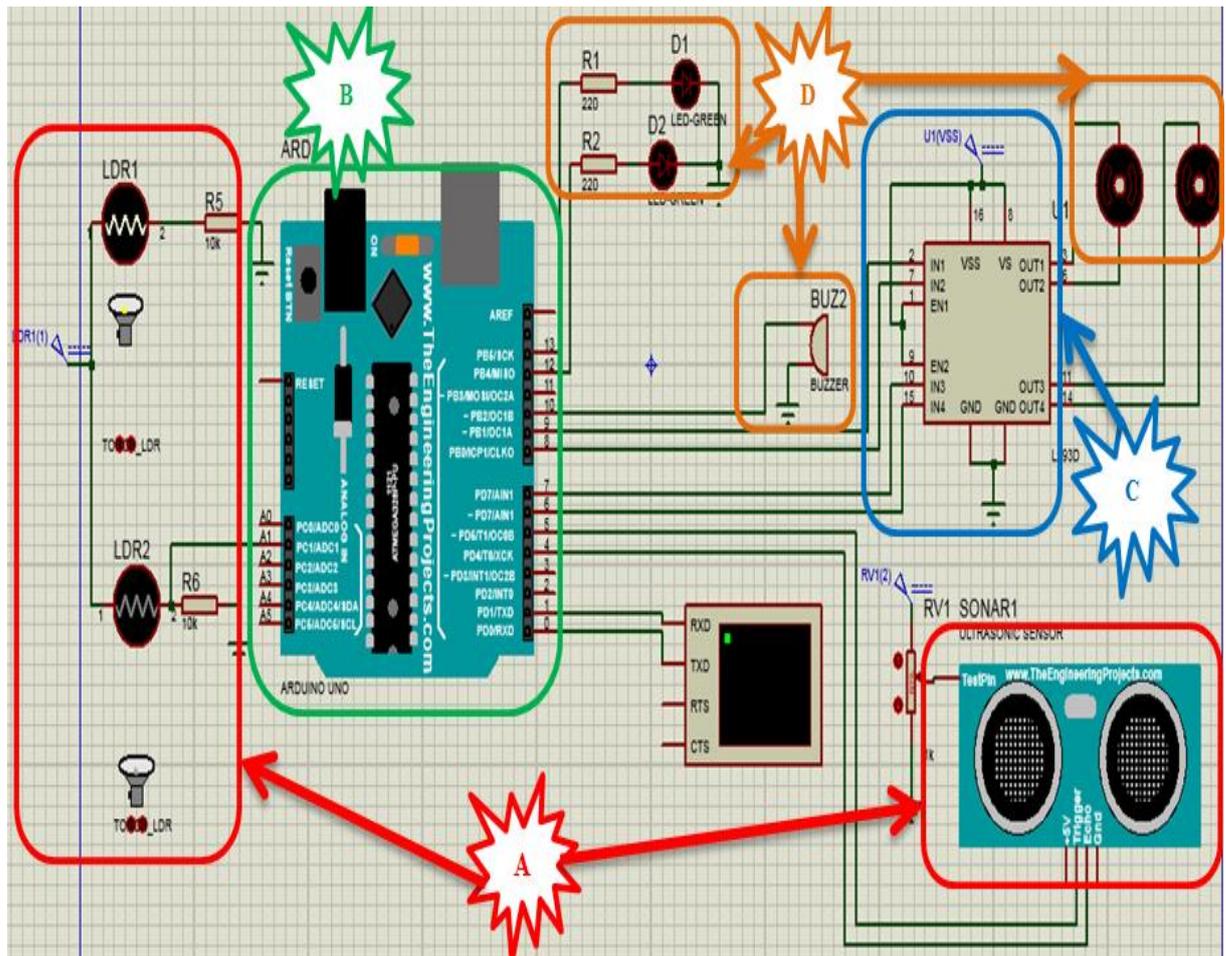


Figure III.9 Circuit électrique générale du robot avec carte Arduino.

III.2.2.3 Principe de fonctionnement

Le principe de fonctionnement du robot c'est d'être dans ce cas sur un sol clair avec une ligne noire, Si le robot a tendance à dévier trop à gauche il faut que les moteurs compensent pour le faire aller plus à droite et inversement.

Et le but c'est de garder la ligne noire bien au centre du robot entre les deux photorésistances au-dessous du robot,

- Quand le robot dévie, la ligne noire se rapproche de la photorésistance de droite donc la photorésistance devient peu éclairée et plus résistante et la tension a ses bornes va augmenter, donc les roues droites s'arrêtent et les roues gauches continuent de se rouler donc le robot vire à droite.
- Et quand le robot détecte un obstacle à moins de 30 cm devant lui les deux roues s'arrêtent et la sonnerie du robot se déclenche.

III.2.2.4 Calcul des circuits

➤ Partie 1 : Circuit d'entrées

La partie « A » dans le circuit électrique **Figure III.9**, elle se compose de LDR et des résistances et une alimentation plus le détecteur d'obstacle.

$$V_+ = V_{CC} \frac{R_5}{(R_5 + R(\text{LDR}))}$$

Dans ce cas

$$R_5 = \frac{V_+}{V_{CC} - V_+} R(\text{LDR})$$

R(LDR) 10 fois plus lumineux → $V_+ = 4.54$ Volts $10\text{k}\Omega \leq R(\text{LDR}) \leq 60\text{k}\Omega$

R(LDR) 10 fois moins lumineux → $V_+ = 0.54$ Volts $R(\text{LDR}) \geq 0.1 \text{ M}\Omega$

Exemple de calculs

$$V_{CC} = 9 \text{ volts } R_5 = \frac{4.5}{9 - 4.5} \times 10\text{k}\Omega$$

→ $R_5 = 10\text{k}\Omega$

➤ Partie 2 : Circuit de commande

La partie « B » dans le circuit électrique **Figure III.9**, elle se compose d'une carte de développement de type Arduino. Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA [22].

➤ Partie 3 : Circuit puissance

La partie « C » dans le circuit électrique **Figure III.9**, elle se compose d'un L293D.

Le L293D est un double pont-H, ses caractéristiques sont : 2 interfaces pour servomoteurs 5V, connectés au timer haute résolution de l'Arduino. Peut piloter 4 moteurs à courant continu DC, ou 2 moteurs pas à pas, ou 2 servomoteurs à la fois. Jusqu'à 4 moteurs DC bidirectionnels avec sélection de la vitesse individuelle (sur 8 bit), mais dans notre cas on va juste commander deux moteurs à courant continu.

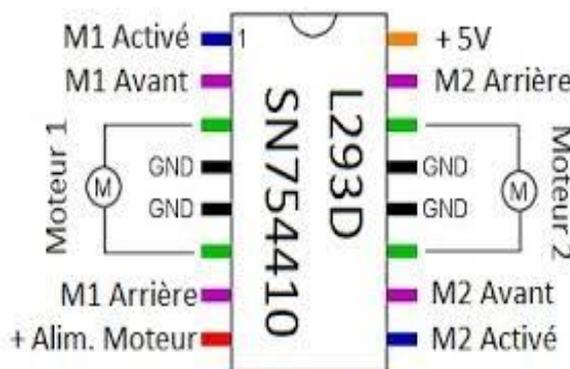


Figure III.10 L293D.

➤ **Partie 4 : Actionneurs**

La partie « D » dans le circuit électrique Figure III.9, partie actionneurs elle se compose de 2 moteurs à courant continu et de 2LED Green et d’un buzzer

III.3 Etude comparative entre les cas sans et avec Arduino

Comme dans le cas de notre étude nous sommes basés sur deux cas de situations, circuits sans et avec carte Arduino. le tableau suivant résume les principaux points de la comparaison entre les deux cas de circuits concernés. Nous remarquons que le circuit à carte Arduino comme il a ses inconvénients, il présente plus d’avantages par rapport au circuit sans carte Arduino.

| | Circuit sans carteA | Circuit avec carte Arduino |
|---|---|-----------------------------------|
| Coût d’utilisation en composants | Utilisation requiert plus de composants | Utilisation appropriée |
| Maintenance | Difficile (changement des composant) | Facile (correction de code) |
| Espace réservée | Plus espace | Moins d’espace |
| Complexité | complexe | Moins complexe |
| Communication | Pas support | support |
| Consommation | Moins économique | économique |

Tableau III.1 Comparaison entre les cas de circuits utilisés sans et avec carte Arduino

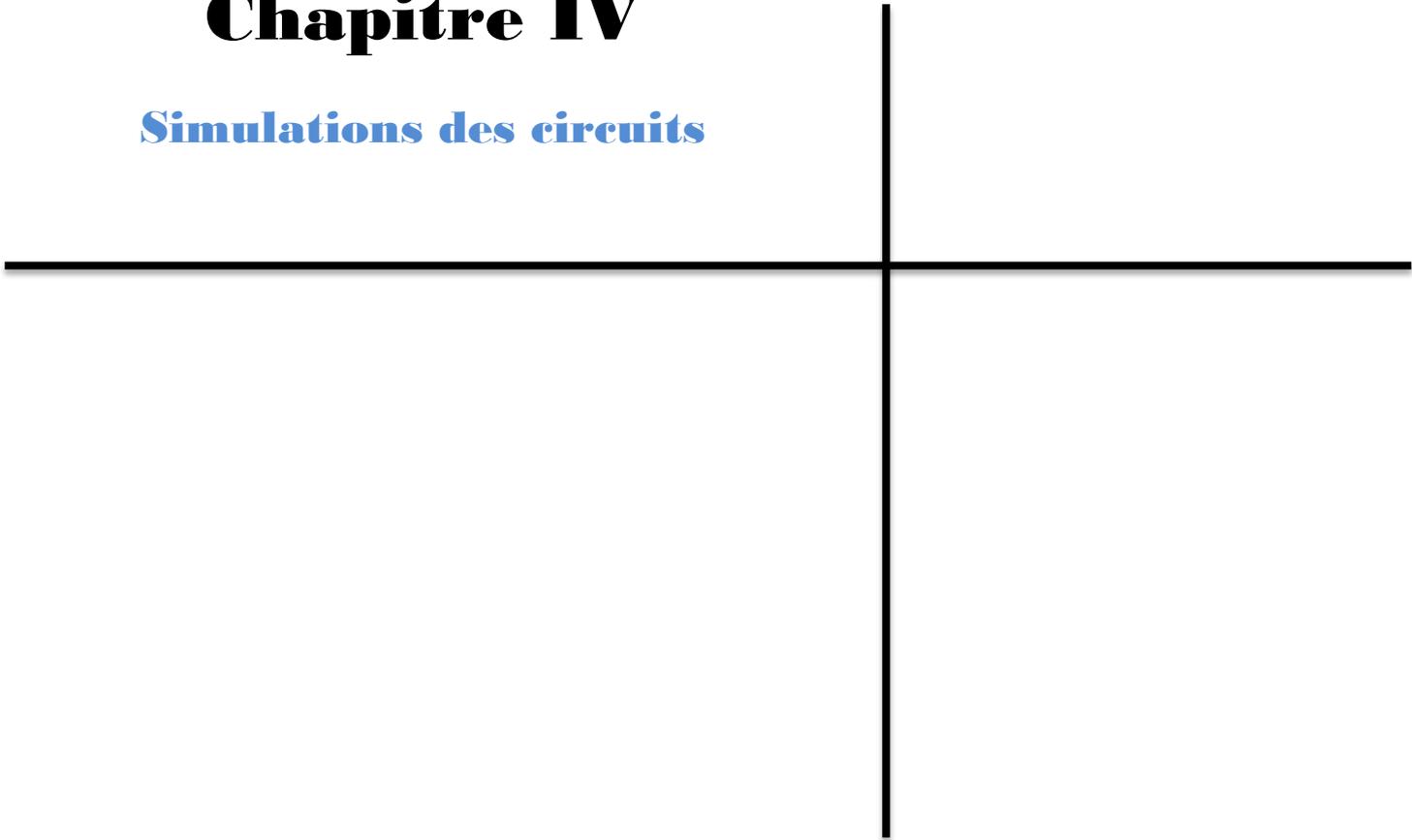
III.4 Conclusion

Dans ce chapitre j'ai étudié les différents circuits électriques sans et avec carte Arduino qui entrent dans le cadre de la conception du système du suiveur de ligne. Donc, deux types de systèmes ont été analysés, sans et avec carte ARDUINO. Ainsi, nous avons eu la possibilité de choisir certains composants selon leurs caractéristiques et de déterminer les valeurs pour d'autres.

Le chapitre suivant, sera consacré aux simulations des circuits électroniques, donc l'utilisation des résultats obtenus sur les différents composants est nécessaire

Chapitre IV

Simulations des circuits



IV.1 Introduction

Ce chapitre sera consacré aux différents tests de simulation de différents circuits électroniques étudiés de notre système dans le troisième chapitre. Donc, différents organigrammes correspondant aux deux situations, sans et avec carte Arduino seront établis. De plus, les programmes concernés seront présentés, ce qui permettra de constater leurs différences.

IV.2 Partie logicielle

Cette partie est dédiée à la représentation des informatiques utilisées dans le développement de notre projet. Ainsi, nous allons exploiter deux plateformes principales :

IV.2.1. Plateforme de programmation Arduino

Dans ce cas, il y a une interface de l'IDE ARDUINO. Ce qui nous offre la possibilité pour développer nos programmes sur les cartes Arduino. La **Figure IV.1** montre la forme de la structure de programmes ARDUINO.

Nous remarquons deux fonctions principales que nous devons respecter pour l'écriture et l'exécution de nos programmes :

✓ **Setup** : contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.).

✓ **Loop** : elle est exécutée en boucle après l'exécution de la fonction setup. Elle continuera de boucler tant que la carte n'est pas mise hors tension. Cette boucle est absolument nécessaire sur les microcontrôleurs étant donné qu'ils n'ont pas de système d'exploitation.

En effet, si l'on omettait cette boucle, à la fin du code produit, il sera impossible de reprendre la main sur la carte Arduino qui exécuterait alors du code aléatoire.

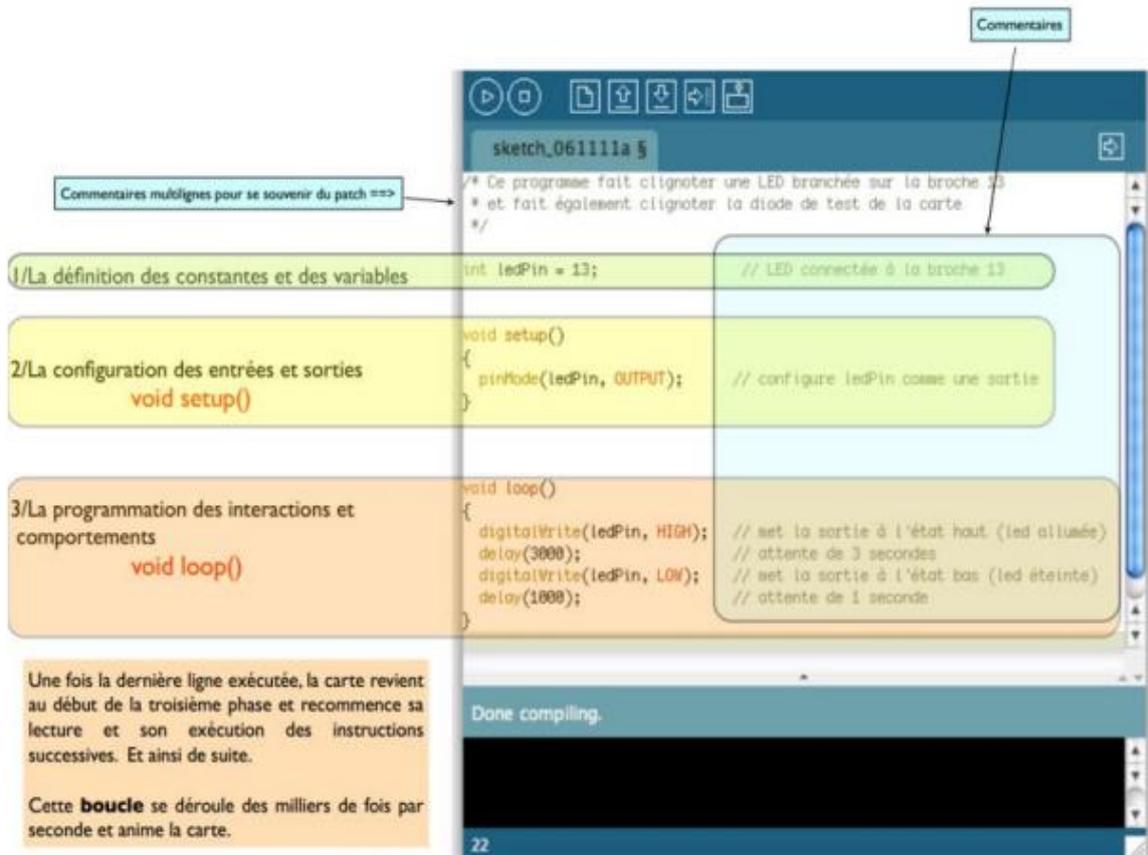


Figure IV. 1 Structure de programme ARDUINO.

IV.2.2. Plateforme de développement et de simulation Proteus

Avant de passer à la réalisation pratique de notre système, nous avons eu recours à la l'établissement des différentes parties électroniques du système déjà étudiés dans le chapitre3, pour cela on utilise le logiciel Proteus. Ce qui permettra d'ajuster, de modifier le circuit et de vérifier les résultats obtenus correspondant aux différentes fonctions du système suiveur de ligne. La figure IV.2

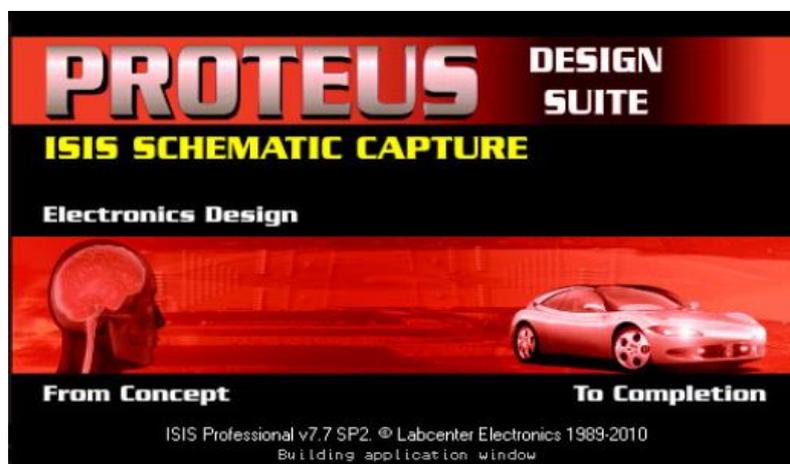


Figure IV.2 Représentation par une photo du logiciel Proteus

Deux logiciels principaux composent cette suite logicielle: ISIS, ARES :

ISIS : pour éditer nos schémas électriques.

ARES : pour réaliser le PCB (**de** l'anglais printed circuit board ou circuit imprimé) de la carte électronique.

IV.3. Simulations des circuits étudiés

IV.3.1.Sans carte Arduino

IV.3.1.1 Organigramme fonctionnel

L'Organigramme suivant représente le fonctionnement de notre projet dans le cas où la carte Arduino n'est pas considérée.

CD : Capteur Droit

CG : Capteur Gauche

MD : Moteur Droit

MG : Moteur Gauche.

1 : Le démarrage du système.

2 : La condition $(CD \& CG) \neq 1$.

3 : La condition $CD=0 \& CG=1$. (« 1 » Signifie que le capteur est illuminé et « 0 » Signifie que le capteur non illuminé).

4 : L'état des deux capteurs (CD, CG).

5 :L'état des deux moteurs (MD, MG).

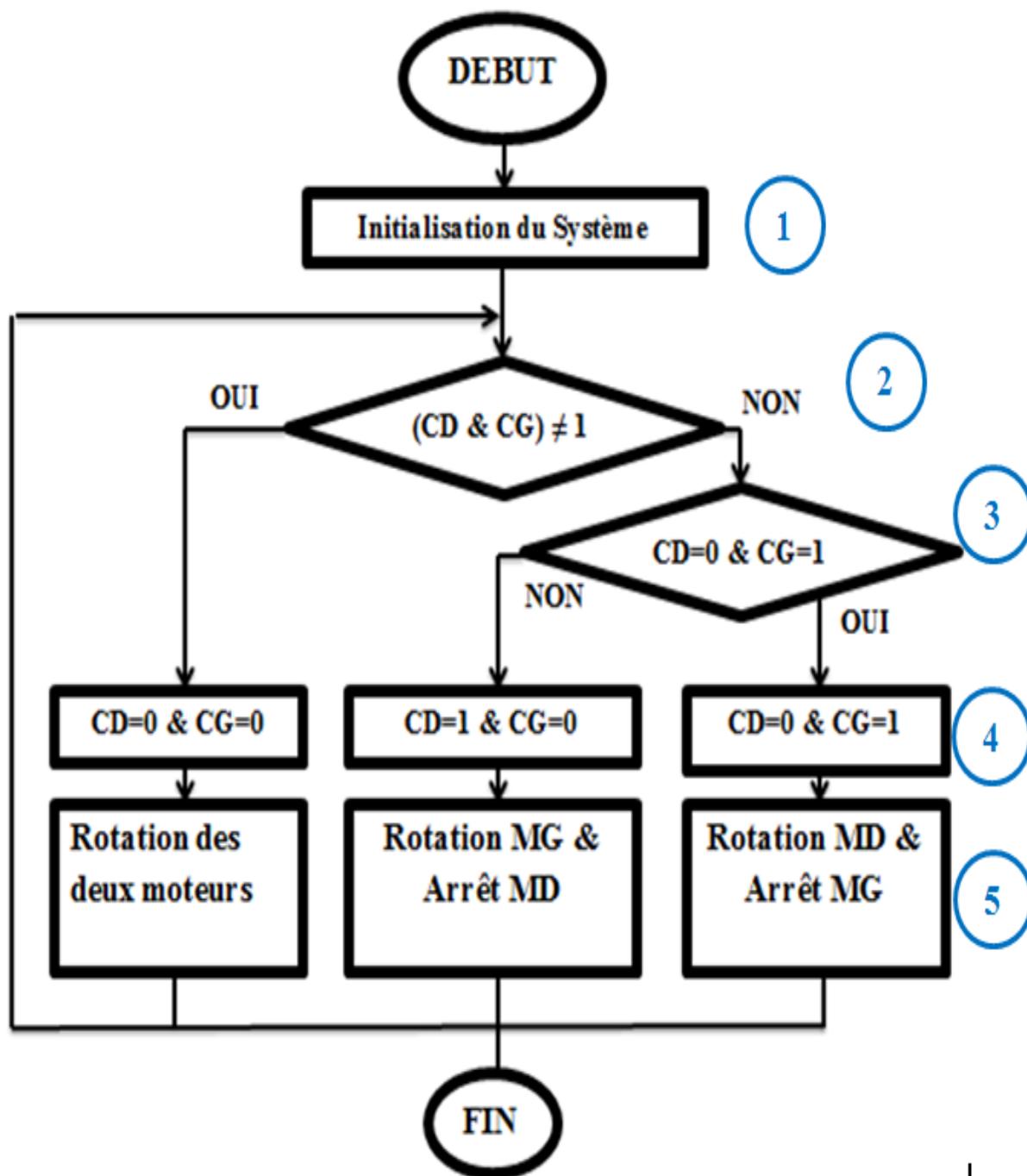


Figure IV.3 Organigramme fonctionnel du suiveur de ligne sans carte Arduino

IV.3.1.2 Simulation du schéma électronique

Les deux LDR sont éteintes (sur un fond noir) → Les deux moteurs marche → Le robot avance tout droit.

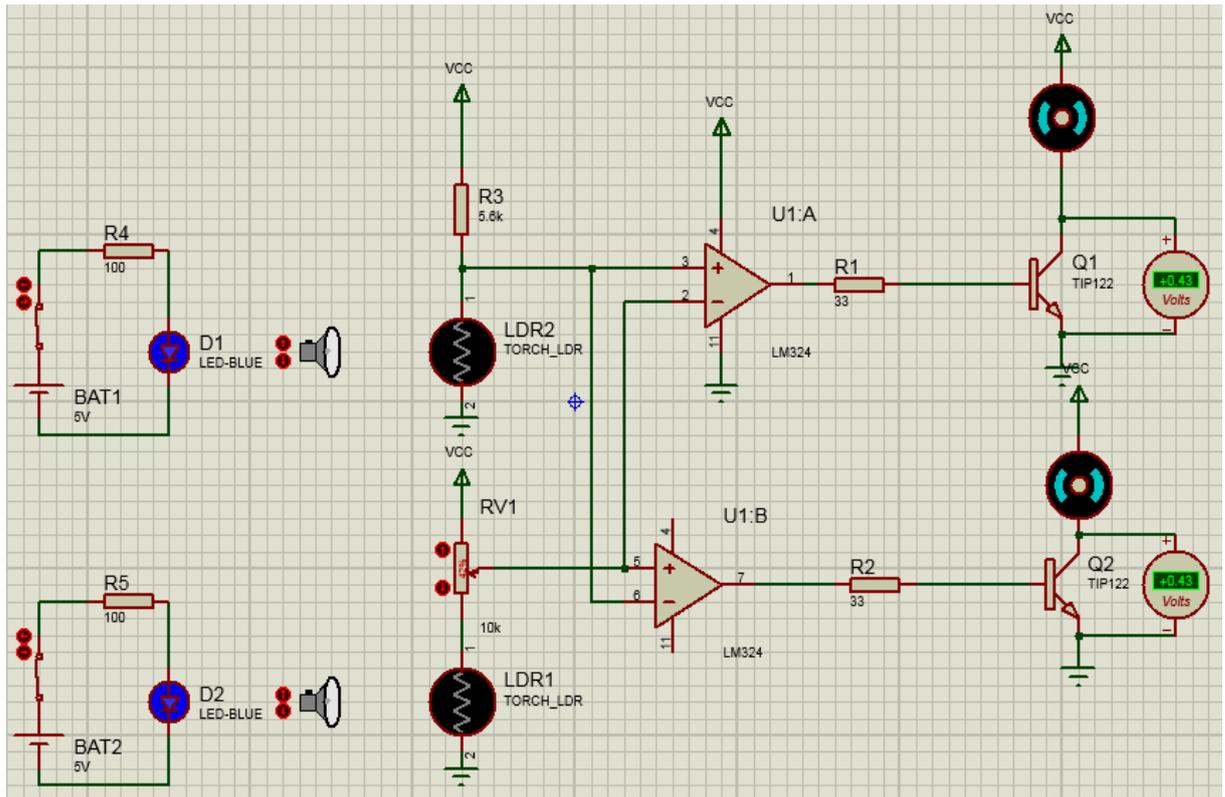


Figure IV.4 Simulation du circuit sans carte Arduino(1).

LDR2 (gauche) est allumée (La ligne blanche est détecté) → Le moteur droit est en marche et le moteur gauche s'arrête → Le robot marche à Gauche.

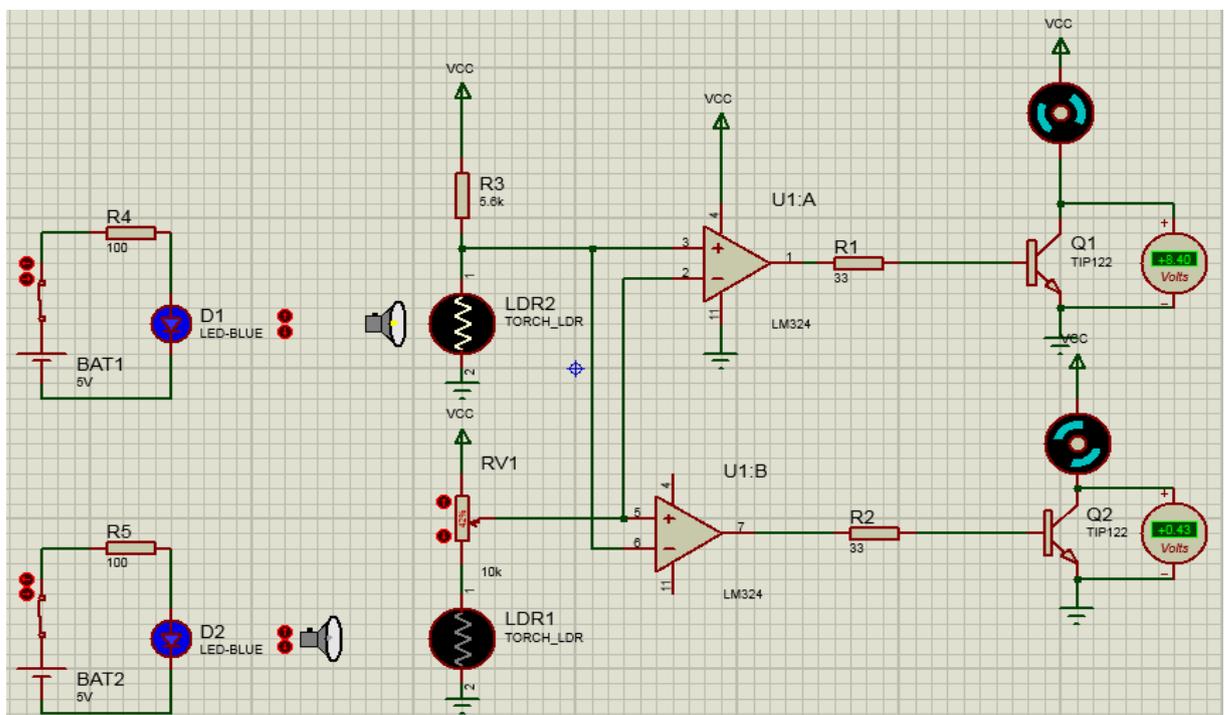


Figure IV.5 Simulation du circuit sans carte Arduino(2)

LDR1 (droite) est allumée (La ligne blanche est détectée) → Le moteur gauche est en marche et le moteur droit s'arrête → Le robot marche à Droite.

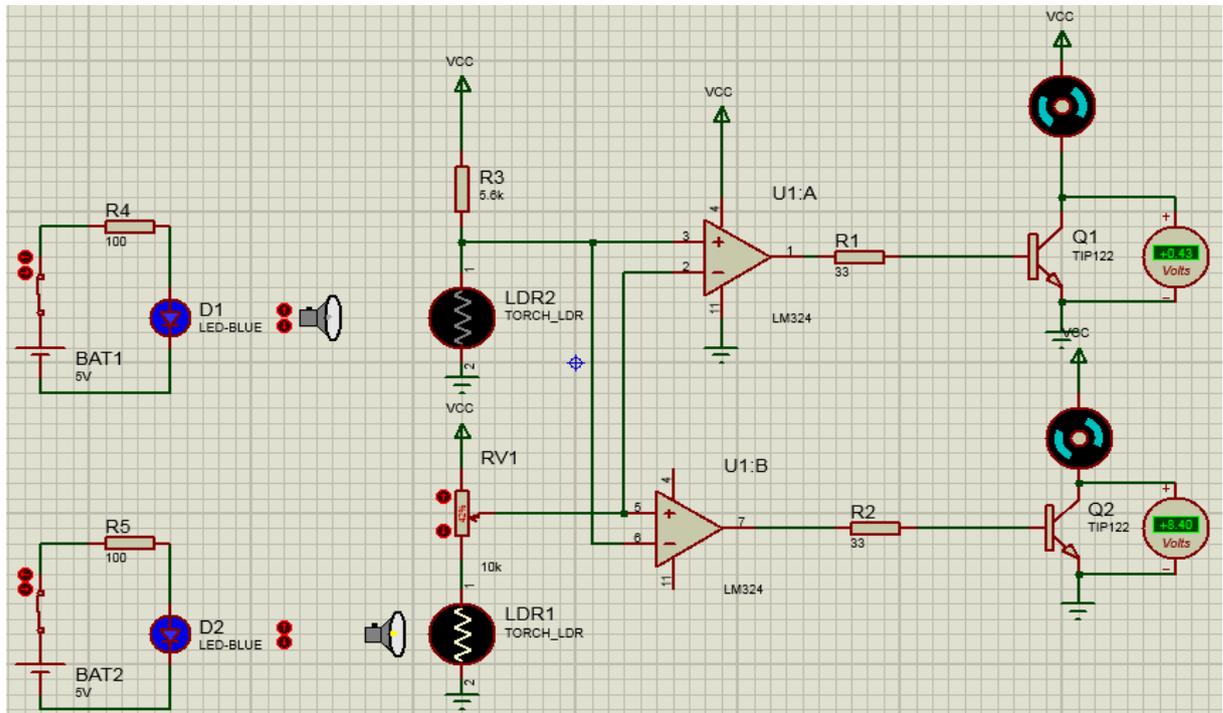


Figure IV.6 Simulation du circuit sans carte Arduino(3)

IV.3.2 Avec carte Arduino

IV.3.2.1 Organigramme fonctionnel

L'Organigramme suivant représente le fonctionnement de notre projet dans le cas où la carte Arduino est considérée.

- 1 : Le démarrage du système.
- 2 : La condition du détecteur d'obstacle (si il y a n obstacle moins de 30cm ou pas)
- 3 : La condition $(CD \& CG) \neq 0$.
- 4 : La condition $CD=0 \& CG=1$. (« 1 » Signifie que le capteur est illuminé et « 0 » Signifie que le capteur non illuminé).
- 5 : L'état des deux capteurs (CD, CG).
- 6 : L'état des deux moteurs (MD, MG).
- 7 : Affichage sur le virtuel terminal ou bien le PC.

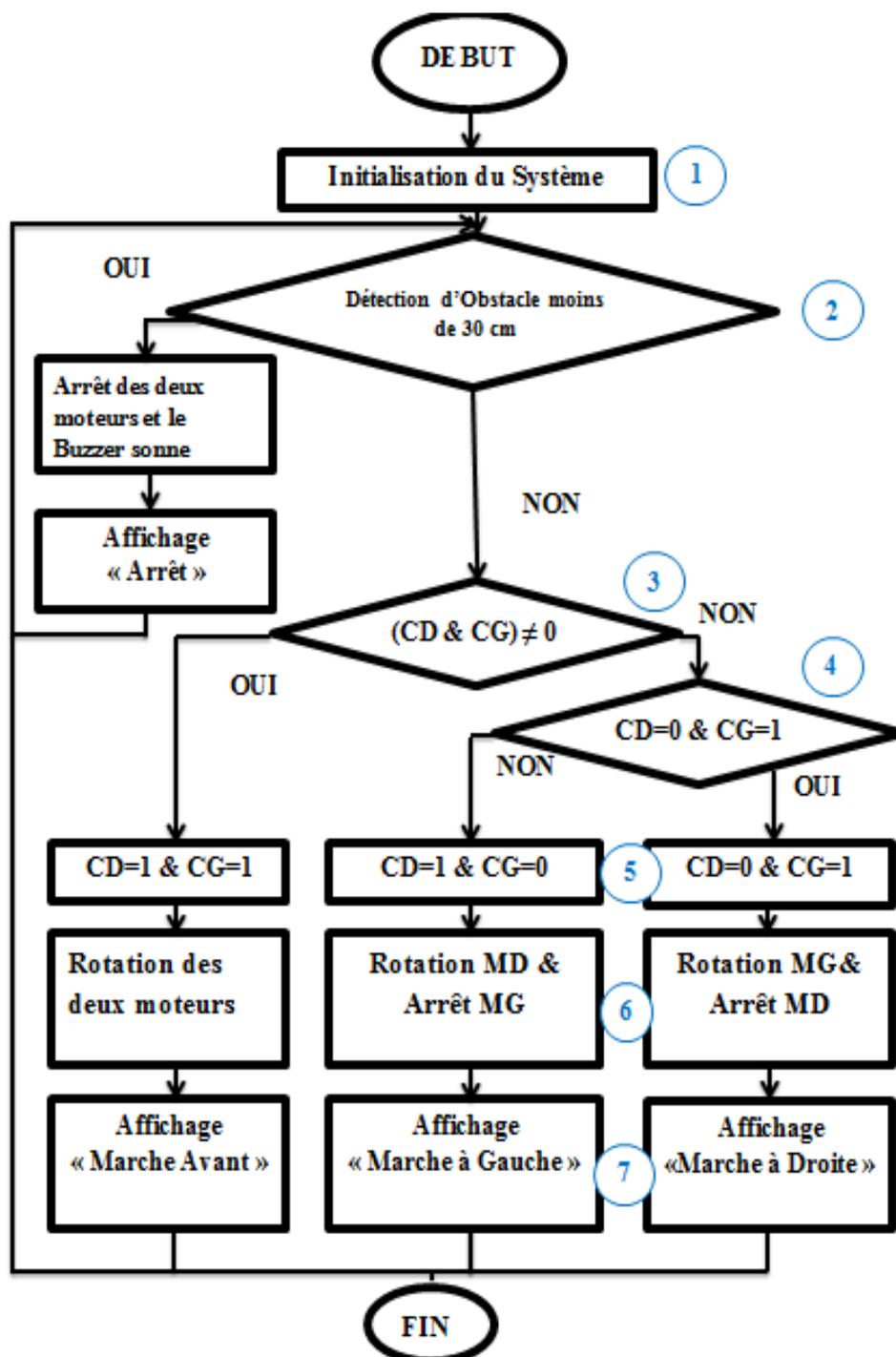


Figure IV.7 Organigramme fonctionnel du suiveur de ligne avec carte Arduino.

IV.3.2.2 Simulation du schéma électronique

Notre circuit avec la carte Arduino c'est un suiveur de ligne et au même temps détecteur d'obstacle, Dans la simulation on a plusieurs cas :

a) Quand il n'y a pas d'obstacle et les deux LDR sont allumées (la ligne noire n'est pas détectée ni à droite ni à gauche) donc les deux LED sont allumées → Passage Marche Avant, ça veut dire les moteurs sont en marche et le robot marche tout droit.

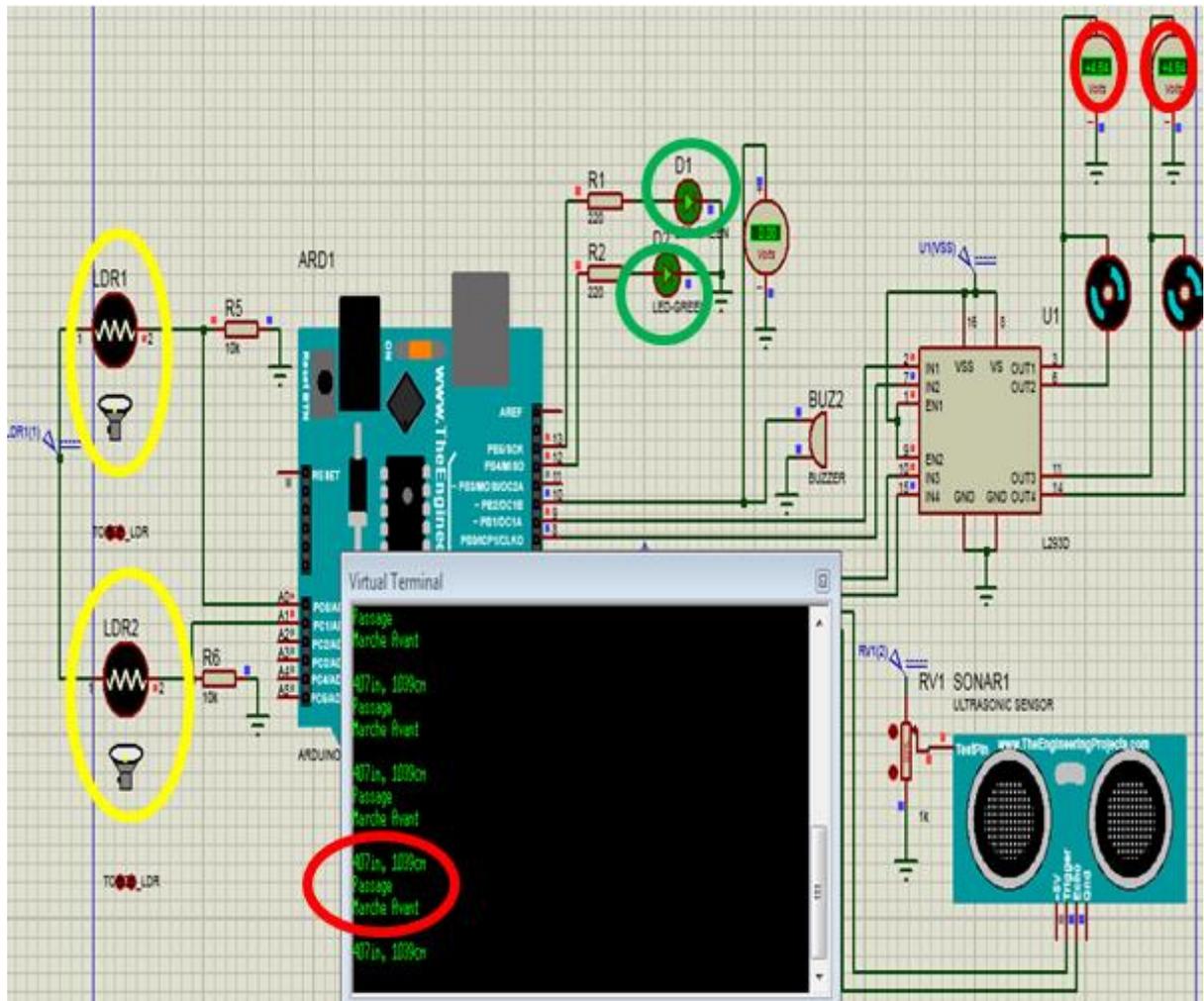


Figure IV.8 Simulation du circuit électrique avec la carte Arduino(a).

- b) Quand il n'y a pas d'obstacle et les deux LDR sont éteintes (sur un fond noir) donc les deux LED sont éteintes → les moteurs ne marche plus et le robot s'arrête.

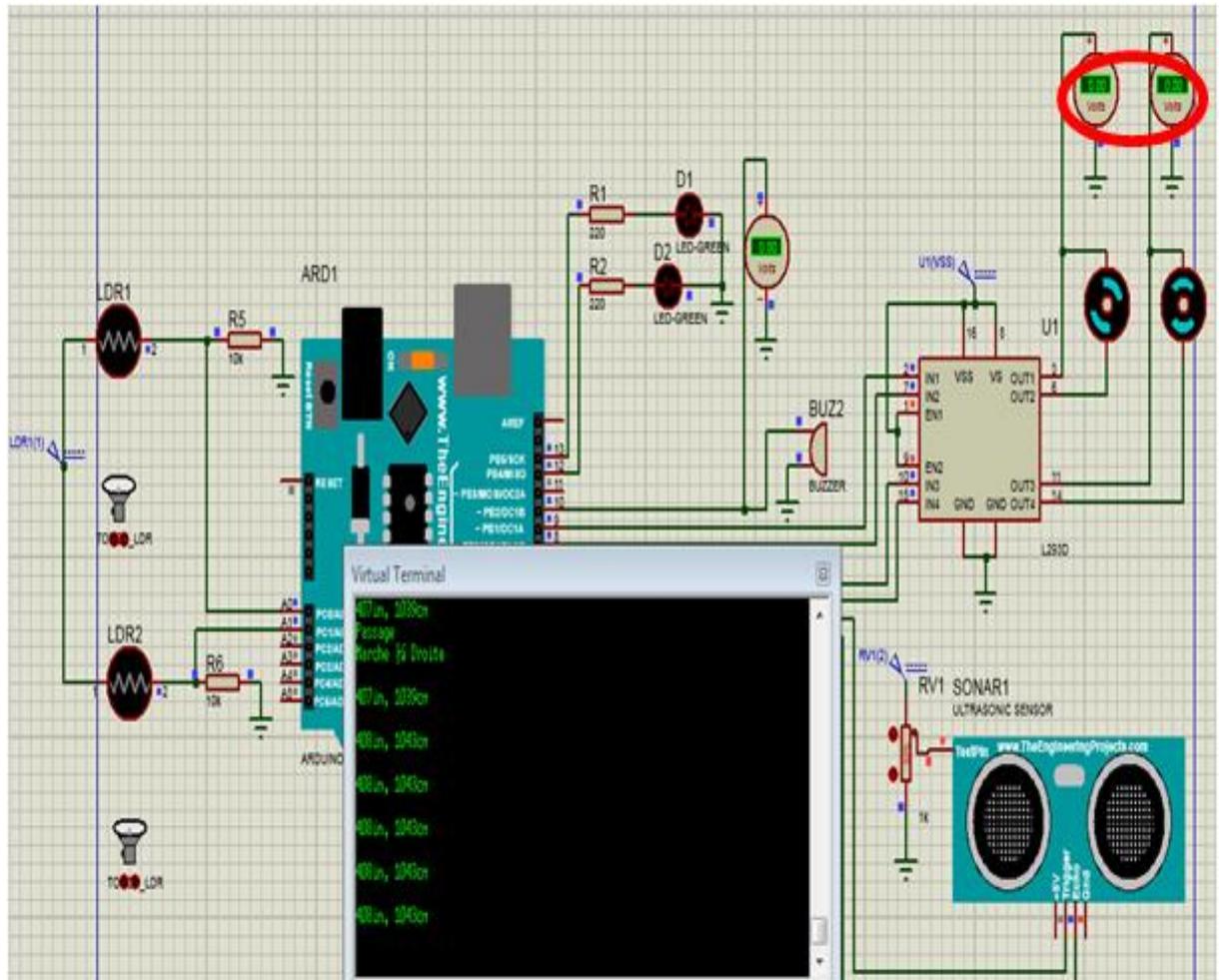


Figure IV.9 Simulation du circuit électrique avec la carte Arduino(b)

- c) Quand il n'y a pas d'obstacle et LDR1 (droit) est éteinte et LDR2 (gauche) est allumée (la ligne noire est détectée à droite (LDR1)) donc la LED D1 (droite) est éteinte et la LED D2 est allumée (gauche) → Passage Marche à Gauche, ça veut dire le moteur droit est en marche et le robot marche à Gauche.

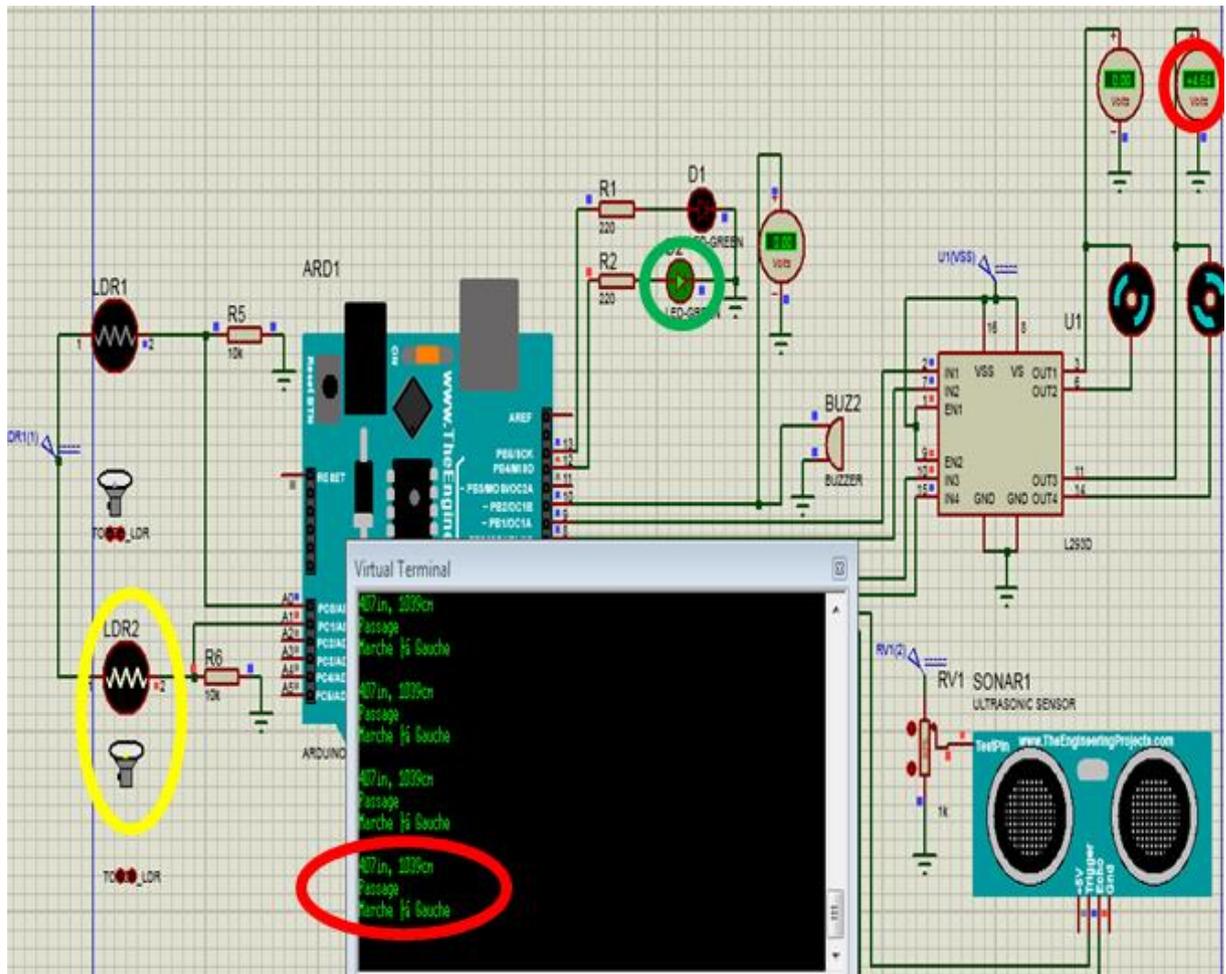


Figure IV.10 Simulation du circuit électrique avec la carte Arduino(c)

- d) Quand il n y a pas d'obstacle et LDR1 (droit) est allumée et LDR2 (gauche) est éteinte (la ligne noire est détecté à gauche (LDR2)) donc la LED D1 (droite) est allumée et la LED D2 est éteinte (gauche) → Passage Marche à Droite, ça veut dire le moteur gauche est en marche et le robot marche à Droite.

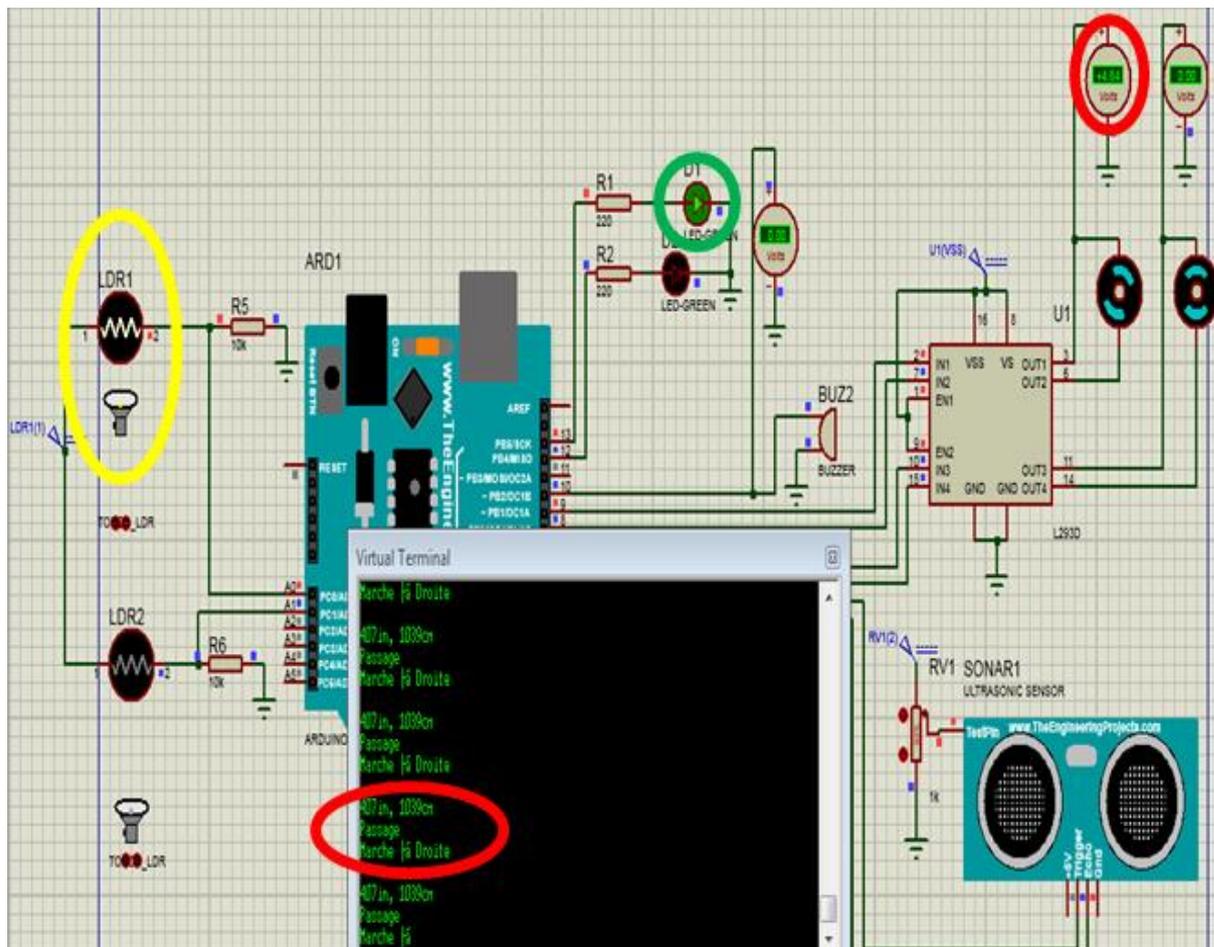


Figure IV.11 Simulation du circuit électrique avec la carte Arduino(d)

- e) Quand il y a d'obstacle près du robot (moins de 30 cm) → Obstacle STOP, les moteurs sont les deux arrêtés et le robot s'arrête, le buzzer sonne (Alerte).

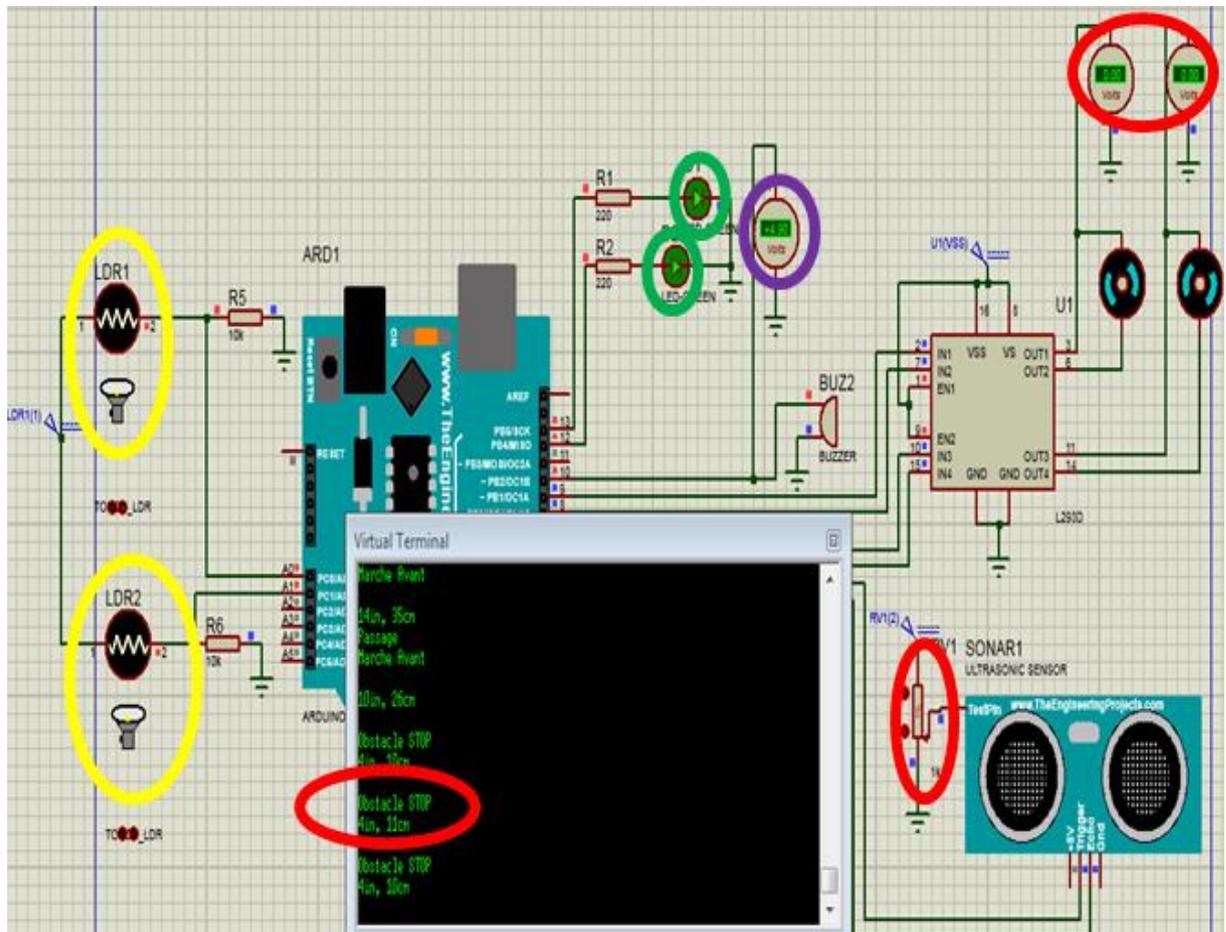


Figure IV.12 Simulation du circuit électrique avec la carte Arduino(e)

➤ Code du programme ARDUINO

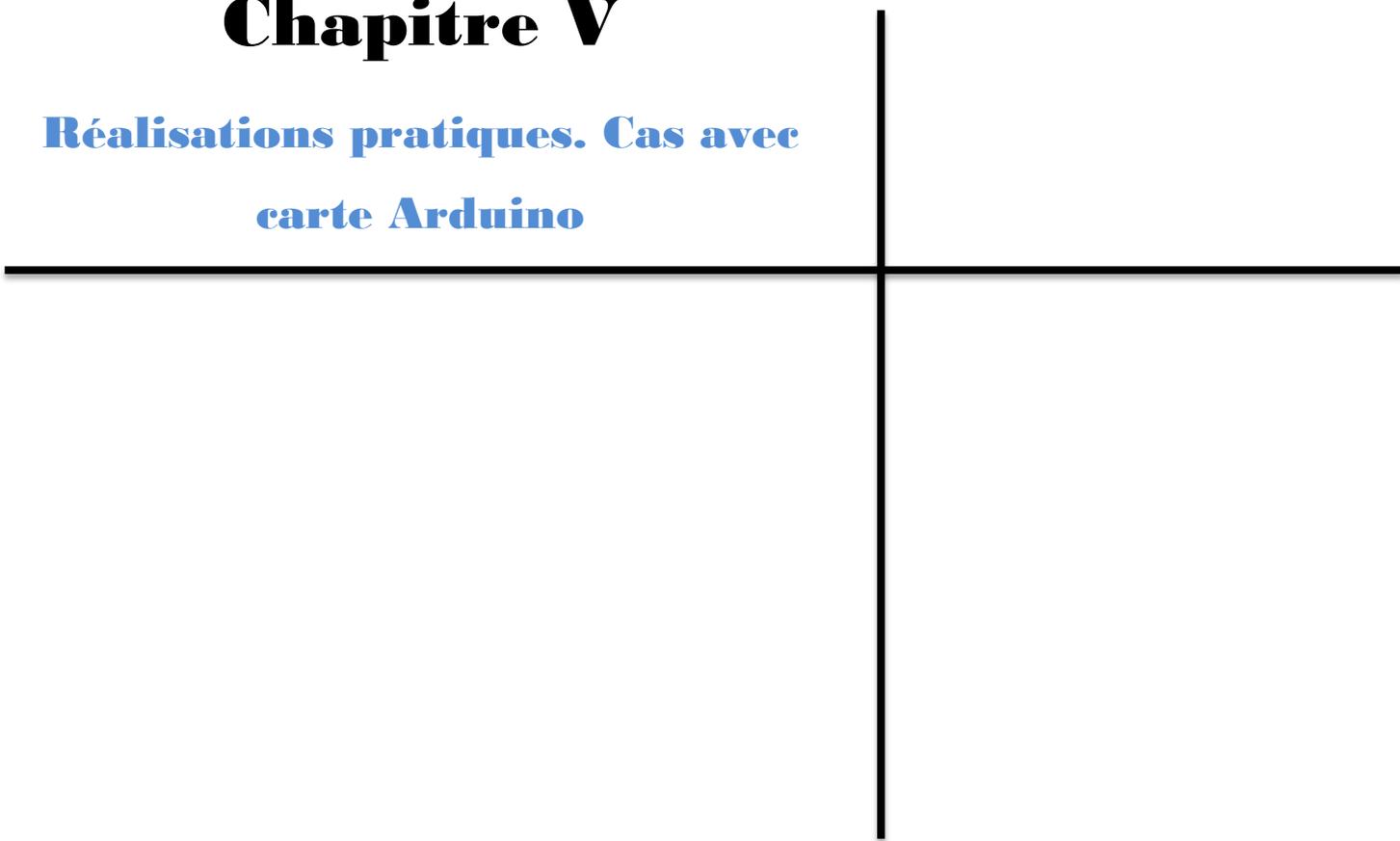
Le code du programme pour le fonctionnement du suiveur de ligne avec Arduino se trouve dans la partie Annexe.

IV.4 Conclusion

Dans ce chapitre nous avons effectué plusieurs simulations des différents circuits électroniques étudiés. Ainsi, les résultats obtenus sont satisfaisants, du fait que les sorties du système répondent aux différentes commandes selon les circuits d'entrées, et cela dans les deux cas de situations, circuits sans et avec carte Arduino. De plus, à travers cette étude, nous avons constaté l'avantage d'un système à base d'une carte à microcontrôleur par rapport son homologue constitué de circuiteries électroniques non programmables. Donc, l'étape de simulation est nécessaire avant toute réalisation pratique, ce qui fera l'objet de notre dernier chapitre.

Chapitre V

**Réalisations pratiques. Cas avec
carte Arduino**



V.1 Introduction

Notre étude a été vérifiée à l'aide de simulations effectuées dans le chapitre précédent. Dans ce qui suit, cela va être vérifié par la réalisation pratique du système pour montrer son fonctionnement. Donc, différentes étapes seront illustrées dans cette partie.

V.2 Organigramme de fonctionnement

L'Organigramme suivant représente le fonctionnement de notre robot.

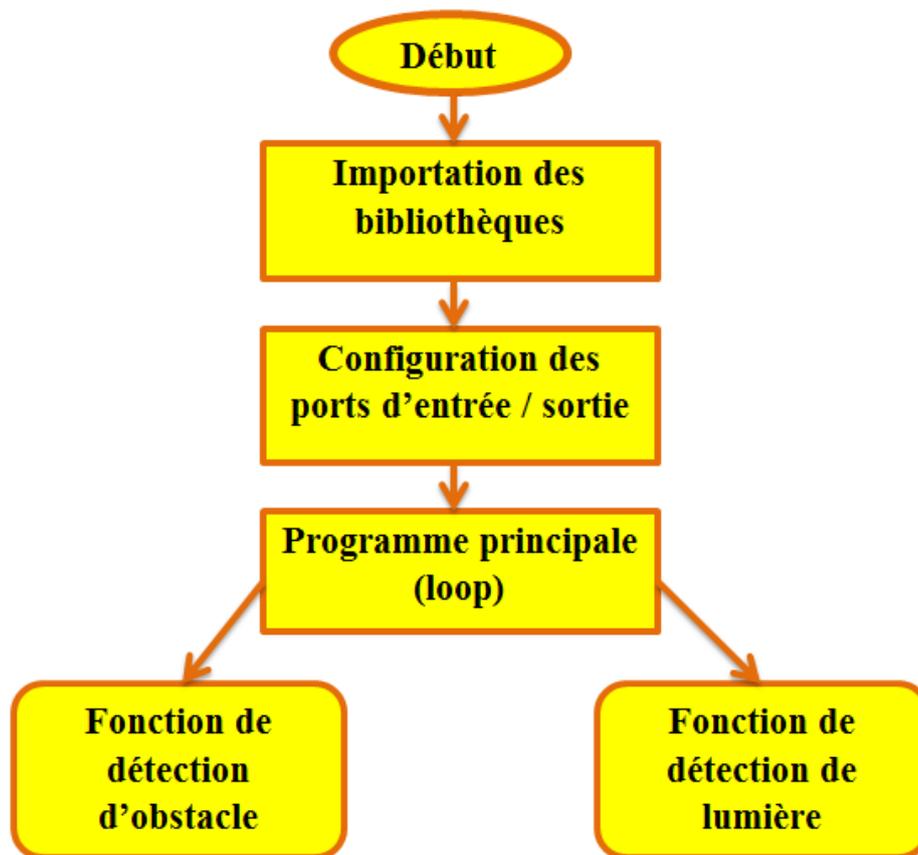


Figure V. 1 Organigramme représente le fonctionnement général du système.

V.3 Matériel et logiciel utilisés

Les besoins du système réalisé en termes de matériel et de logiciel sont comme suit :

- Le matériel utilisé
 - ✓ Une Arduino UNO.
 - ✓ Des résistances.
 - ✓ Des LEDs.
 - ✓ Capteurs photorésistance (LDR)
 - ✓ Capteur Ultrason.

- ✓ Des moteurs à courant continu avec des roues.
- ✓ Une plaque d'essai.
- ✓ Buzzer.
- ✓ L293D.
- Les logiciels utilisés
 - ✓ Logiciel ISIS/Proteus
 - ✓ Logiciel Arduino IDE

V.4 Réalisation pratique sur une maquette

V.4.1. Construction du Robot

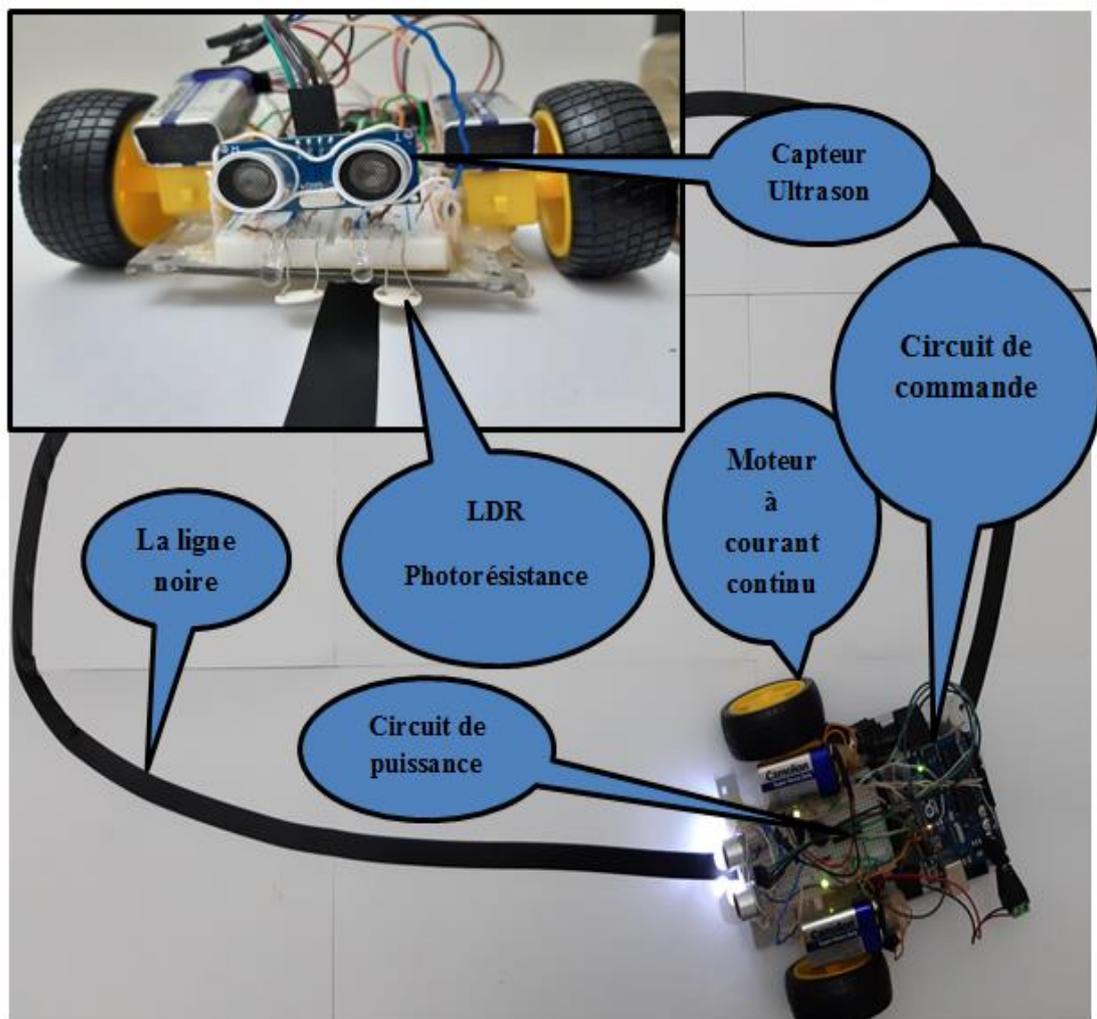


Figure V. 2 *La Construction du Robot.*

V.4.2. Les différents fonctions réalisées dans notre système

1. Fonction de la détection d'Obstacle

La fonction de la détection d'Obstacle est réalisée via le capteur Ultrason, quand il y a un obstacle a moins de 30 cm les deux moteurs à courant continu s'arrêtent.

2. Fonction de la détection de la lumière

La fonction de la détection de la lumière est réalisée via la photorésistance (capteur LDR), quand la photorésistance devient plus éclairée (présence de la lumière) et moins résistante et la tension a ses bornes vas descendre.

V.4.3. chargement du programme principal

Après l'écriture de code principal on a des étapes principales pour charger le programme sur Arduino :

- ✓ **Choisir le type de carte**

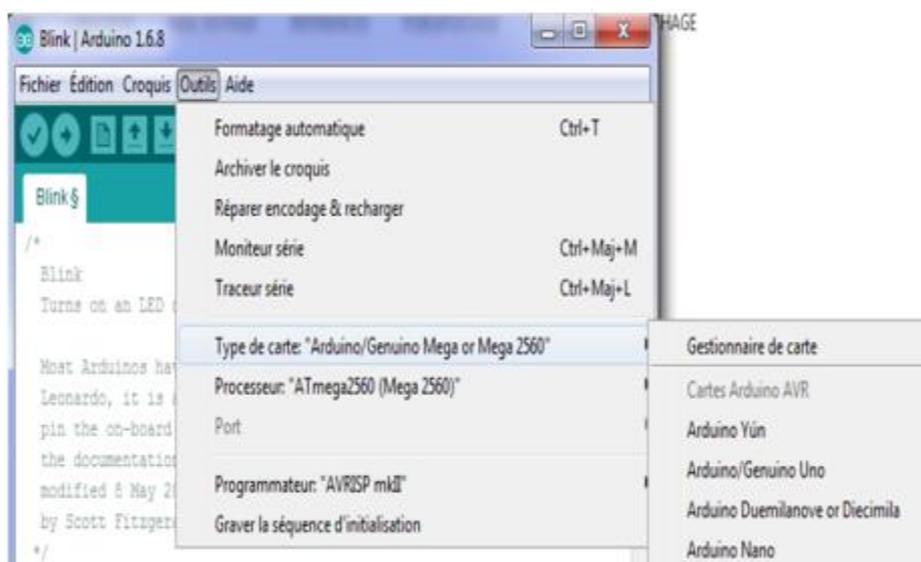


Figure V.3 Paramétrage de la carte étape1

- ✓ **Choisir le Serial Port (COM Port)**

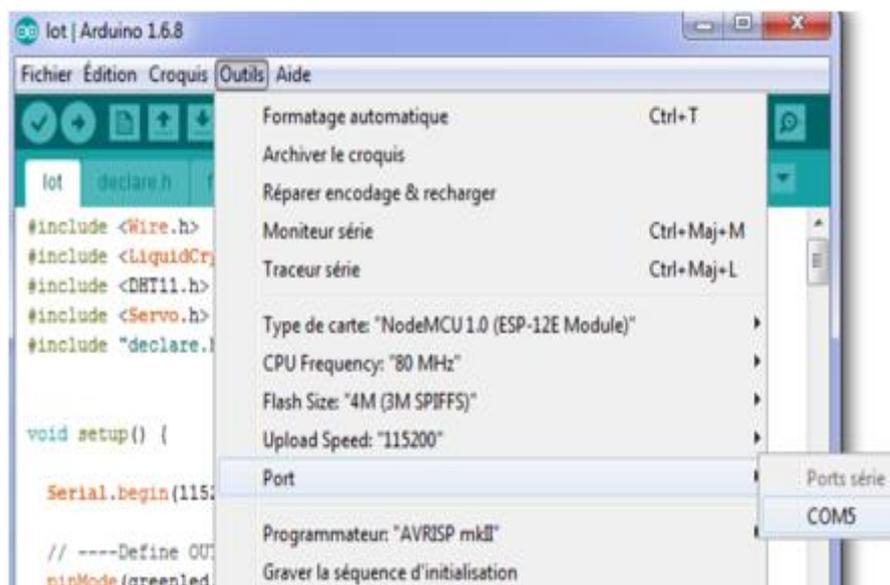


Figure V.4 Paramétrage de la carte étape2

✓ **Chargement du programme sur la carte Arduino**

On clique sur le Bouton Téléverser.



Figure V.5 Paramétrage de la carte étape3

Une simple manipulation enchaînée doit être suivie afin d’injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L’exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d’alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne [22].

La **Figure V.5**, illustre l’étape pour le téléversement du programme déjà écrit et sauvegardé au niveau du PC vers la carte Arduino afin d’assurer le bon fonctionnement système.



Figure V.6 Téléversement de programme vers la carte Arduino

V.5 Mise en œuvre

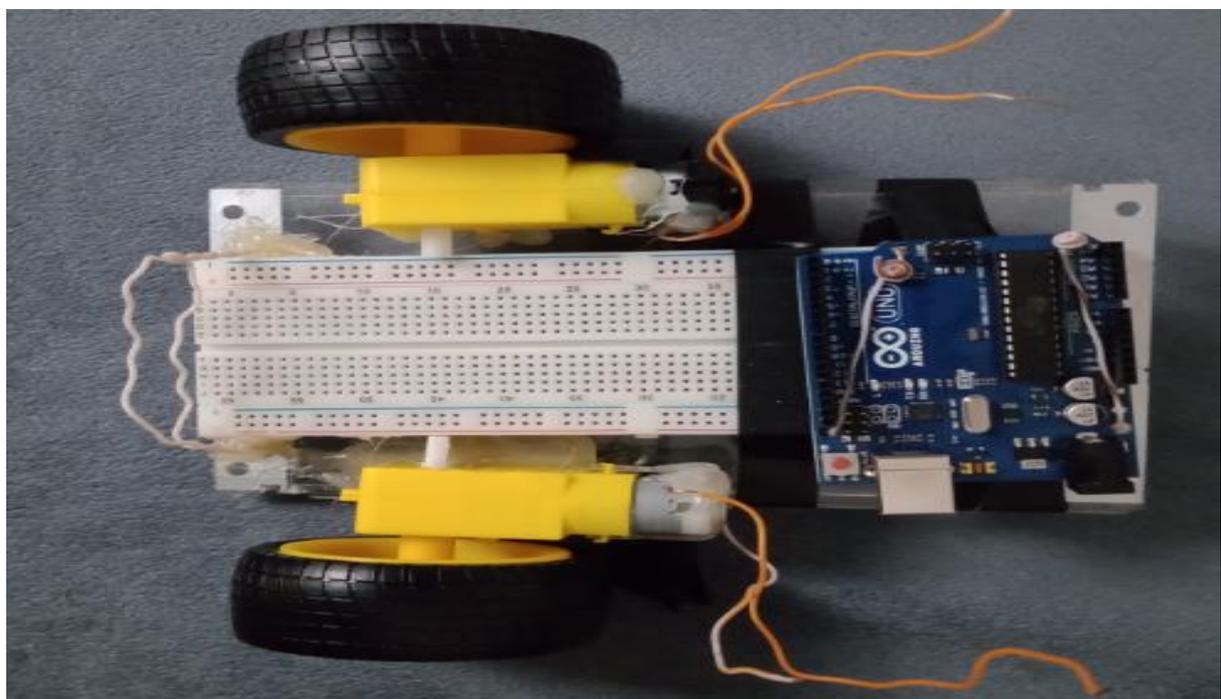


Figure V.7 Schéma de la plaque d'essai + moteurs à courant continu + Arduino.

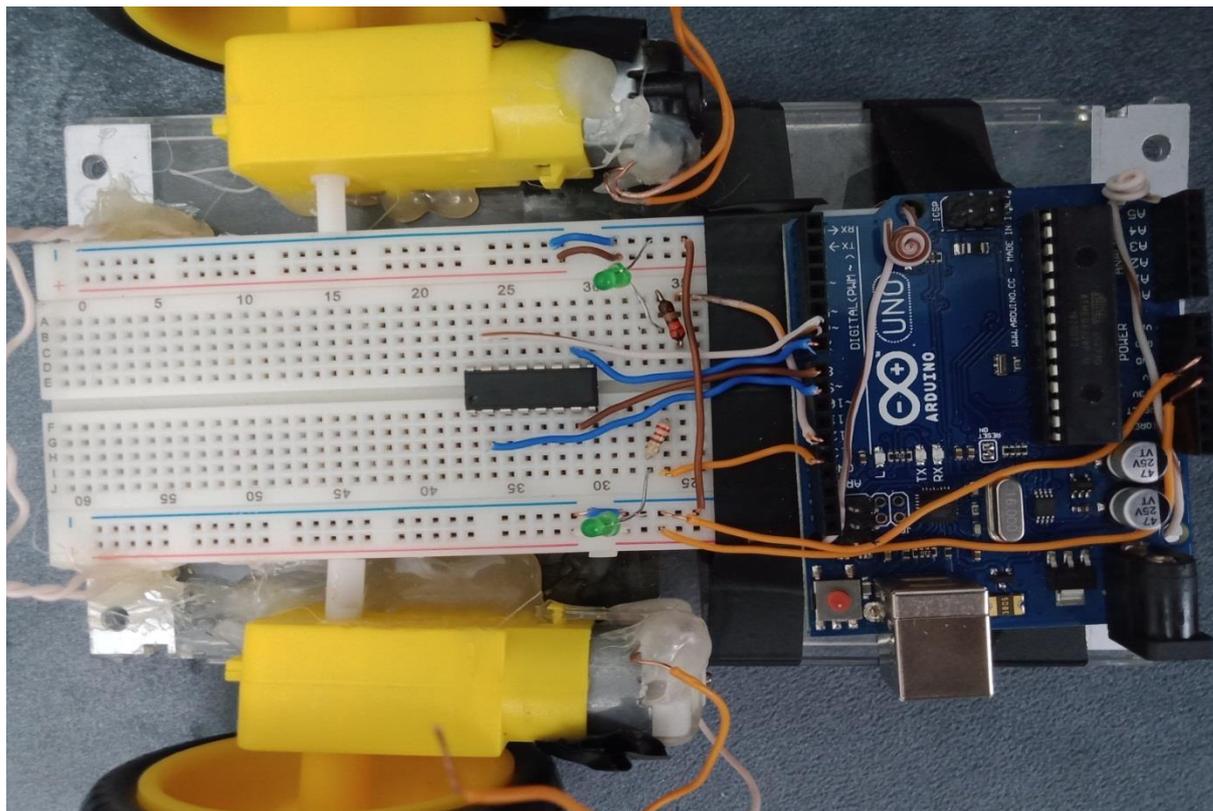


Figure V.8 Schéma de la plaque d'essai + moteurs à courant continu + Arduino + circuit de puissance.

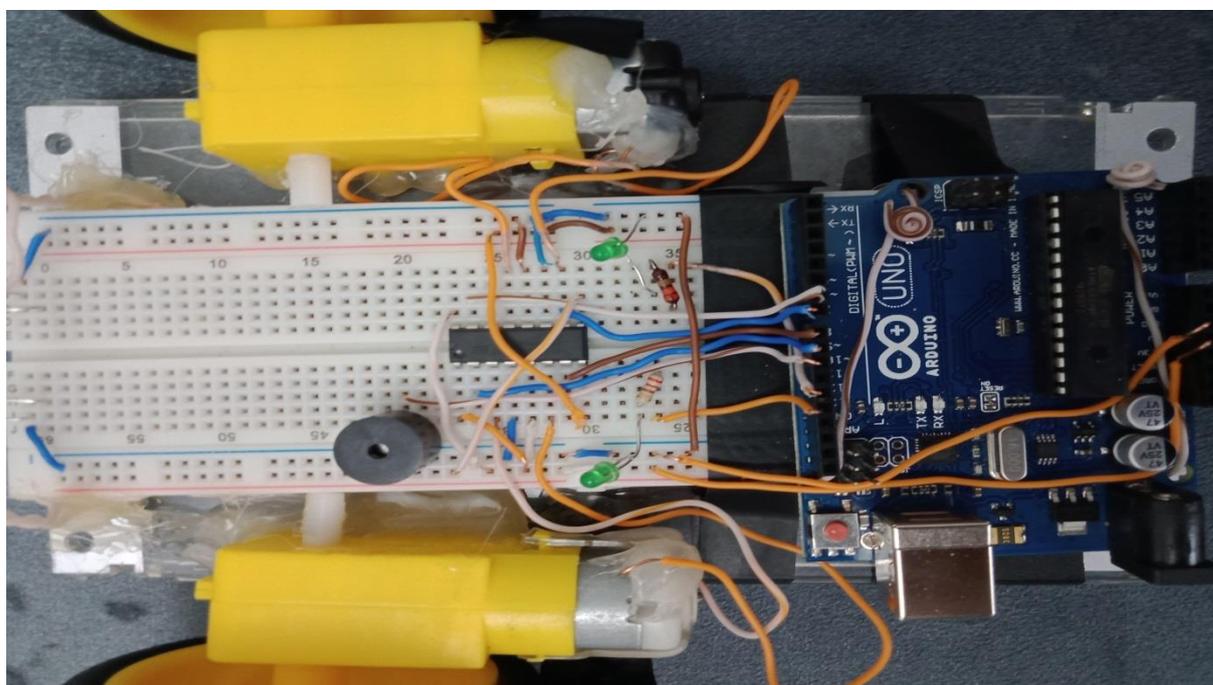


Figure V.9 Schéma de la plaque d'essai + moteurs à courant continu + Arduino + circuit de puissance + buzzer.

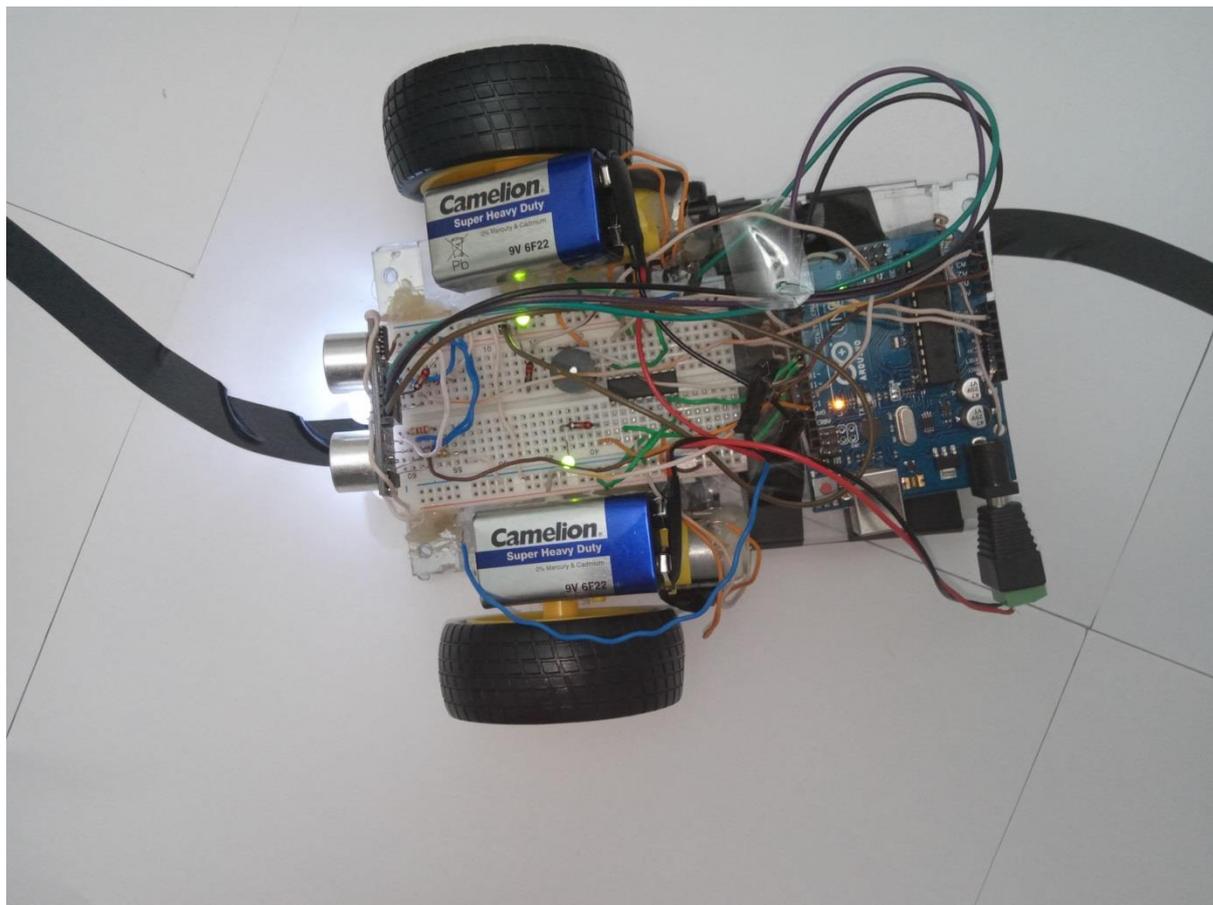


Figure V.10 Schéma du circuit complet(a).

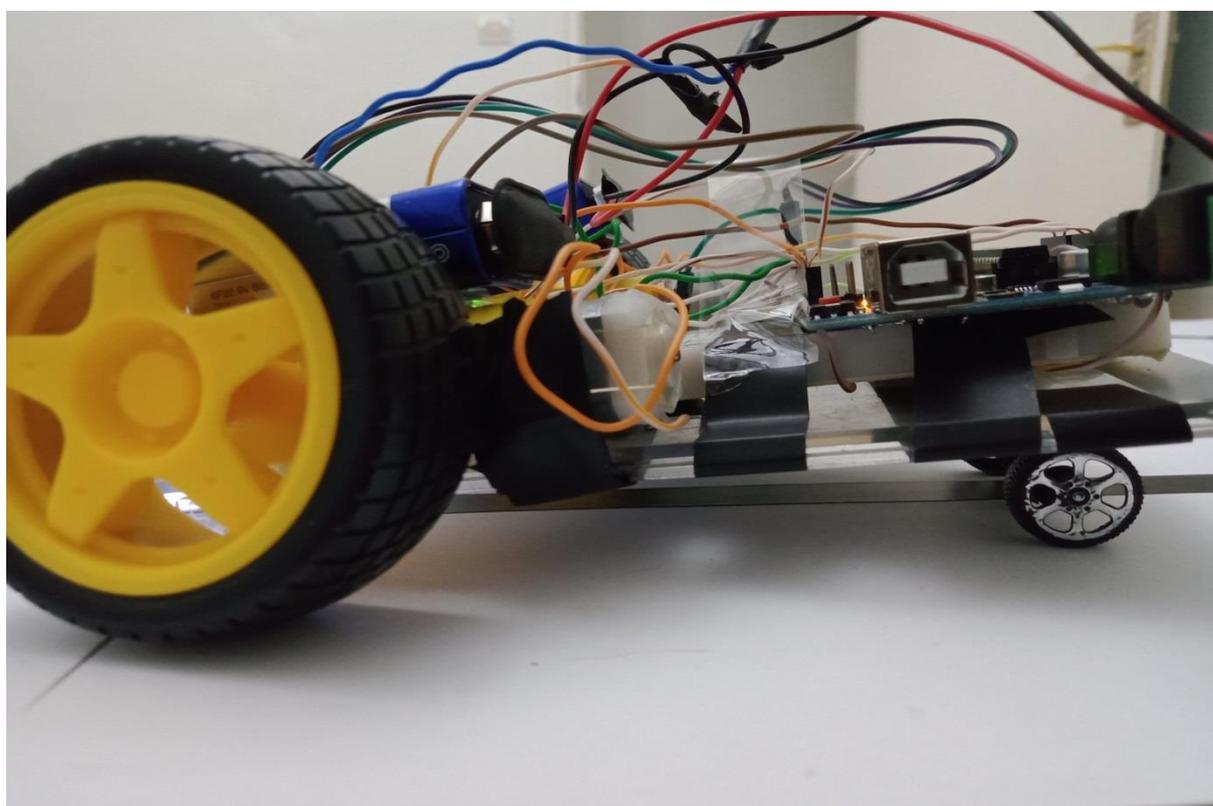


Figure V.11 Schéma du circuit complet(b)

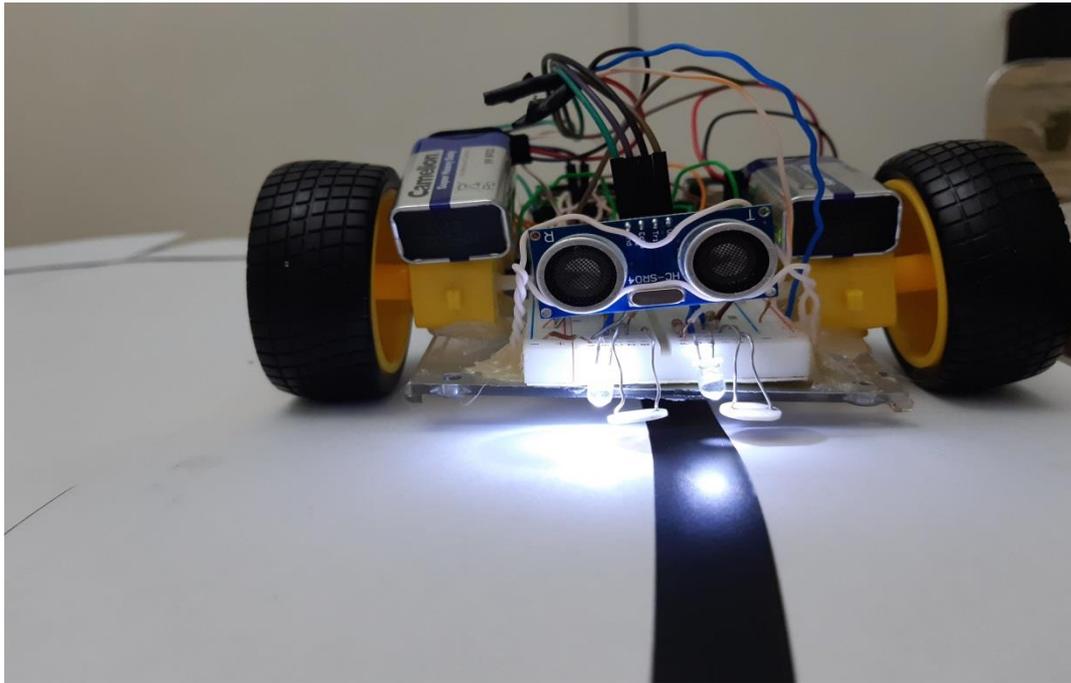


Figure V.12 Schéma du circuit complet(c)

V.6 Nomenclatures

- **Résistance :**
 - ✓ R1 et R2 : 220 Ω (rouge, rouge, marron, or).
 - ✓ R5 et R6 : 10 K Ω (marron, noir, orange, or).
- **Capteurs:**
 - ✓ LDR: Photorésistance.
 - ✓ Sonar1 : Capteur Ultrason.
- **Semi-conducteurs**
 - ✓ D1 et D2 : Diode lumineuse (LED Green).
- **Divers:**
 - ✓ ARDUINO UNO.
 - ✓ BUZZER.
 - ✓ U1 : L293D.

V.7 Etude économique des circuits

Le tableau suivant donne le coût des différents composants utilisés.

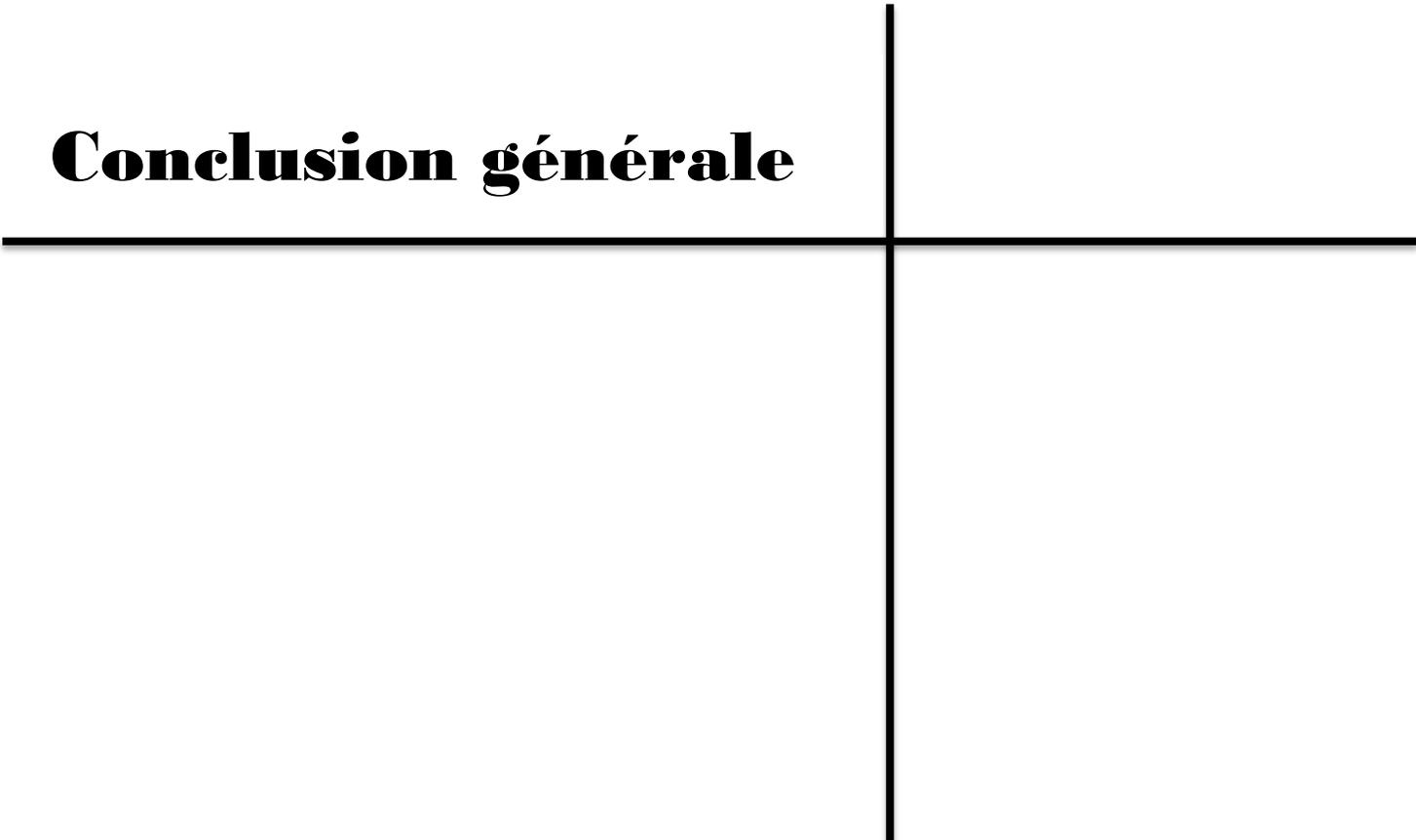
| Désignation | Prix unitaire (DA) | Quantité | Total ligne (DA) |
|--|--------------------|----------|------------------|
| Arduino UNO | 3000 DA | 1 | 3000 DA |
| BUZZER | 80 DA | 1 | 80 DA |
| Moteur à courant continu avec la roue | 850 DA | 2 | 1700 DA |
| L293D | 320 DA | 1 | 320 DA |
| Capteur Ultrason | 800 DA | 1 | 800 DA |
| Résistance (220 Ω) | 5 DA | 2 | 10 DA |
| Résistance (10 KΩ) | 5DA | 2 | 10 DA |
| LED GREEN | 10 DA | 2 | 20 DA |
| LDR | 80 DA | 2 | 160 DA |
| ALIMENTATION 9V avec coupleur | 150 DA | 2 | 300 DA |
| Plaque d'essai | 800 DA | 1 | 800 DA |
| Total | | | 7200 DA |

Tableau V. 2 *Tableau représentant le coût des composants du projet.*

V.8 Conclusion

Dans cette partie, les tests des différentes sorties du système réalisé pratiquement, nous ont donné la possibilité de montrer le bon fonctionnement des circuits en entrées formellement en relation avec le circuit de commande géré à partir d'une carte Arduino UNO. Donc, plusieurs éventualités peuvent être considérées pour améliorer ces circuits ainsi que les programmes.

Conclusion générale



Conclusion générale

L'objectif de mon projet était d'étudier, de simuler et de réaliser un robot à base d'une carte Arduino ayant la capacité de suivre une ligne bien précise.

D'abord, l'étude bibliographique menée a permis de se rendre compte de l'importance d'un système actif donné vis-à-vis du poids des nouveautés technologiques. Se passer de ce développement permettrait le fonctionnement des systèmes sans avoir recours à des améliorations.

Pour cela, plusieurs simulations ont été effectuées pour tester le fonctionnement des différents circuits étudiés. Dans ce cas, j'ai exploité le logiciel de conception assistée par ordinateur pour l'électronique « Proteus ». Ce qui m'a permis de saisir la réalisation de nos schémas électroniques, donc de mieux se rattraper pour renforcer mes prérequis dans ce domaine.

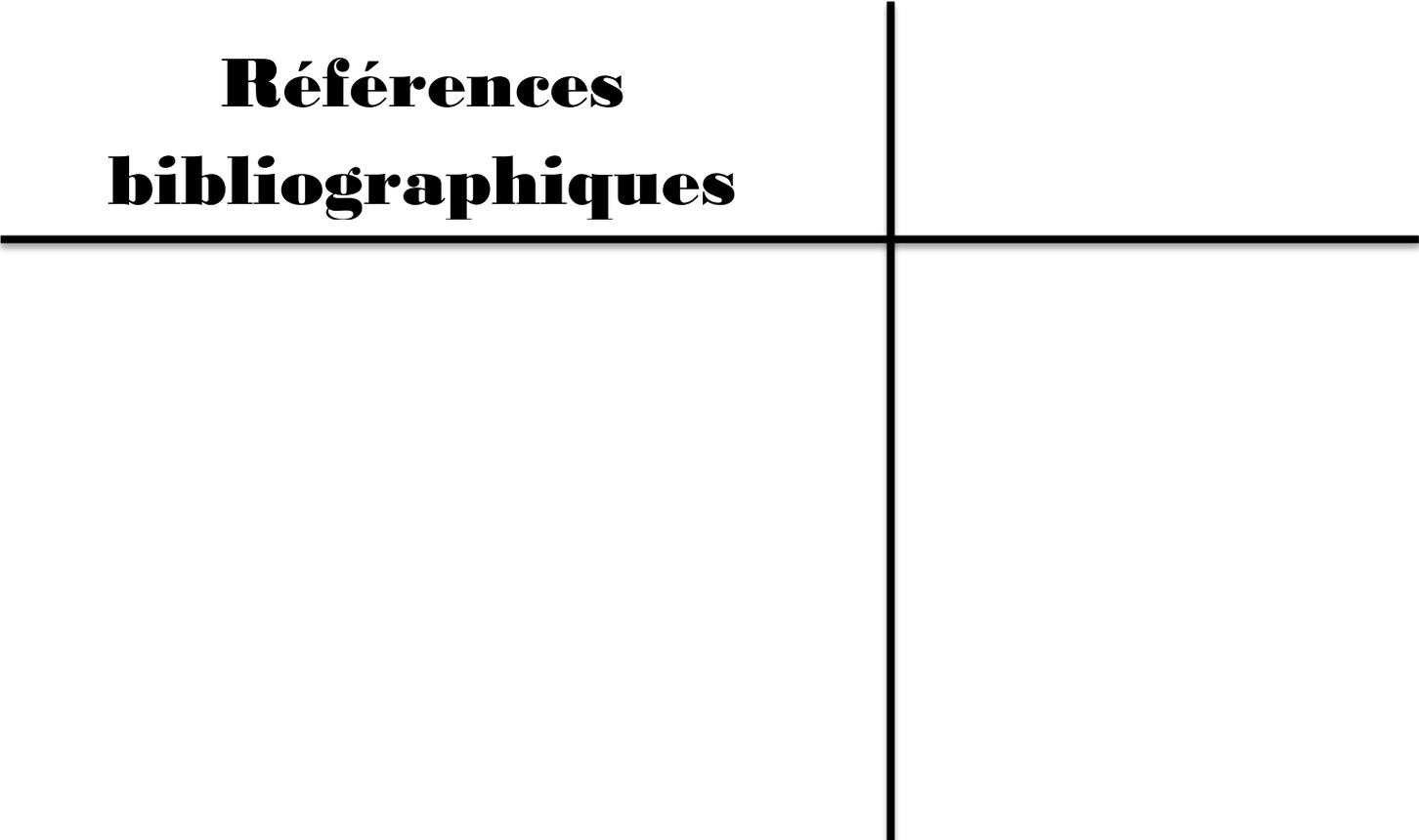
De plus, j'ai saisi l'occasion de réaliser plusieurs programmes correspondant à diverses fonctions d'usage données dans le but de suivre bien déterminée. Donc, chacune de ces fonctions a été programmée dans les cas sans et avec Arduino, Ce qui m'a permis d'obtenir un même résultat, mais pour deux situations différentes. Cela représente un avantage pour mieux comprendre la différence.

Par la suite, le robot suiveur a été réalisé. Cela, m'a donné un point avantageux et motivant, et qui m'a poussé à avoir un certain savoir-faire dans le domaine de notre spécialité. Bien sûr, durant le processus de réalisation du système, j'ai rencontré pas mal d'obstacles, voire l'indisponibilité de quelques composants, ainsi que la contrainte liée au manque de mes compétences sur l'application de certains composants rencontrés très nouveaux pour moi, mais à l'issue de ce projet, j'ai pu comprendre leur importance et leur utilisation pratique.

Le travail développé dans le cadre de ce mémoire ouvre des voies d'amélioration que l'on peut classer comme suit :

- Application du système réalisé avec d'autres types de microcontrôleurs.
- Développement du système réalisé en se basant sur le principe des objets connectés
- Application du système réalisé avec des capteurs d'obstacle.

**Références
bibliographiques**



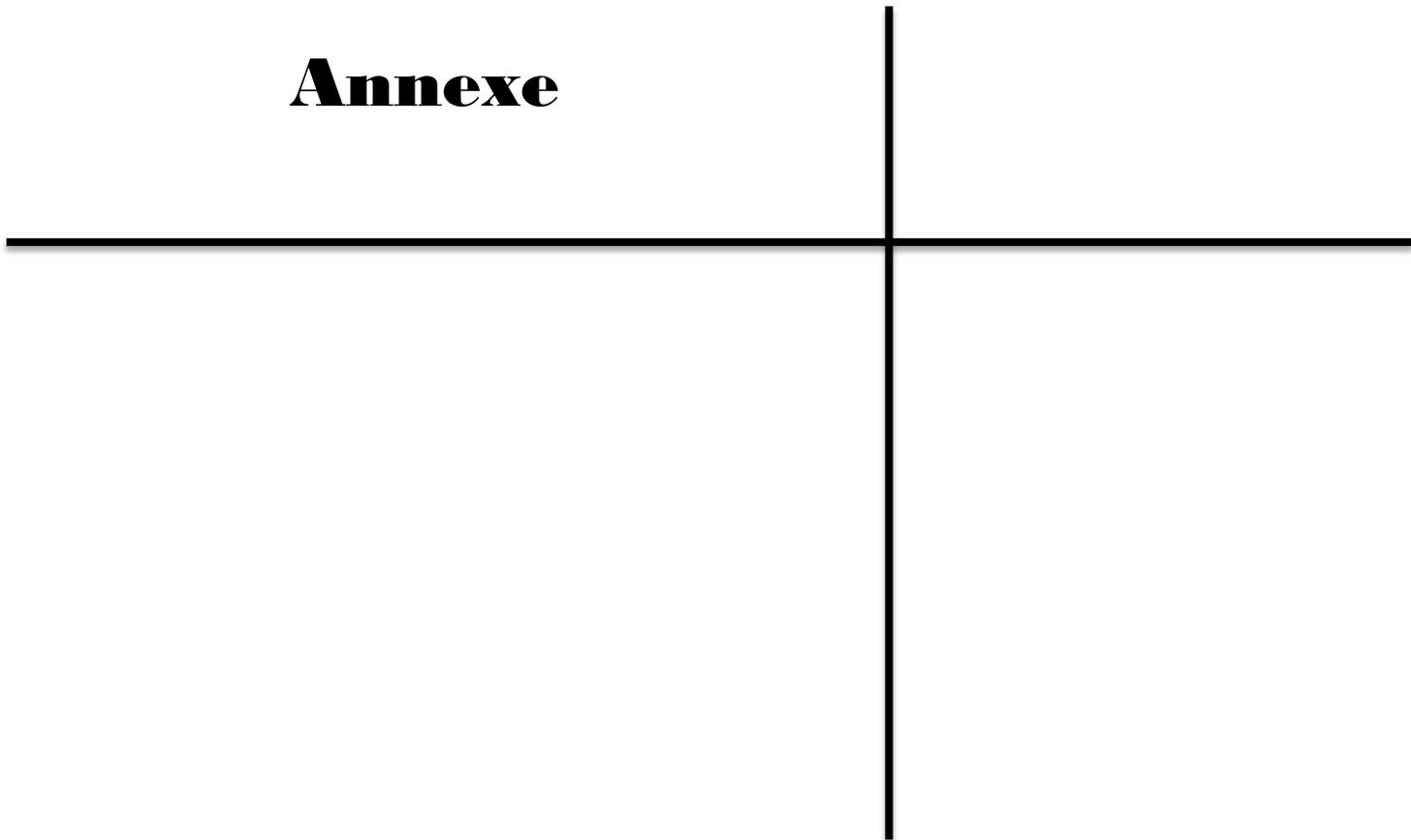
Conclusion générale

- [01] Emmanuel Simeu. Test et Surveillance Intégrés des Systèmes Embarqués. Micro et nanotechnologies/Microélectronique. Université Joseph-Fourier - Grenoble I, 2005.
- [02] https://fr.wikipedia.org/wiki/Robot_suiveur_de_ligne
- [03] <https://www.larousse.fr/dictionnaires/francais/actionneur/>
- [04] <https://fr.wikiversity.org/wiki/Capteur/>
- [05] http://www.recreomath.qc.ca/am_ligne.htm
- [06] http://www.recreomath.qc.ca/am_ligne.htm
- [07] ROBOTIQUE, ISTIA, Université Angers Jean-Louis Boimond.
- [08] <http://technologie.joueb.com/autres.shtml>
- [09] Jean ALLART. Quocduy NGUYEN. Hélène SOUDRY, 'Robot Suiveur de Ligne', Projet de Physique P6. STPI/P6/2013-003.
- [10] <http://technologie.joueb.com/autres.shtml>
- [11] <https://www.radioalgerie.dz/news/fr/article/20160220/69189.html>
- [12] Henri Nussbaumer, informatique industrielle, représentation et traitement de l'information, collection informatique, 1994.
- [13] <https://fr.wikipedia.org/wiki/Arduino>
- [14] <https://www.redohm.fr/2020/01/carte-microcontroleur-2/>
- [15] <https://www.generationrobots.com/fr/401684-carte-controleur-robotique-arbotix-m.html>
- [16] <https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>
- [17] <https://www.researchgate.net/publication/314674254> FORMATION ARDUINO MATLAB SIMULINK Commande d'un système thermique à l'aide de la carte Arduino UNO
- [18] Jean-François THULLIER, 'microcontrôleur arduino pour les plp mathématiques – physique chimie', 4^{ème} version, Mars 2020.
- [19] Leo Louis, 'Working principle of arduino and using it as a tool for study and research', International Journal of Control, Automation, Communication and Systems (IJCAACS), Vol.1, No.2, April 2016.
- [20] <https://www.technologuepro.com/microcontroleur-2/arduino/programmer.html>

Références Bibliographiques

[21] <https://www.mdp.fr/documentation/lexique/courant-continu/definition.html#:~:text=Les%20caract%C3%A9ristiques%20du%20moteur%20courant,%C3%A0%20la%20tension%20d'alimentation.>

Annexe



Annexe

Atmel : est un fabricant mondial de composants à semi-conducteur, créé en 1984 par George Perlegos et dont l'activité historique est le développement de mémoires non volatiles. Aujourd'hui, Atmel développe également des microcontrôleurs.

AVR : est le terme utilisé par **Atmel** pour désigner le cœur du processeur et la famille de microcontrôleurs qui le mettent en œuvre.

C++ : est un langage de programmation compilé permettant la programmation sous de multiples paradigmes (comme la programmation procédurale, orientée objet ou générique). Ses bonnes performances, et sa compatibilité avec le C en font un des langages de programmation les plus utilisés dans les applications où la performance est critique.

Application Java : est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable.

Application libre : est un logiciel dont l'utilisation, l'étude, la modification et la duplication par autrui en vue de sa diffusion sont permises, techniquement et légalement, ceci afin de garantir certaines libertés induites, dont le contrôle du programme par l'utilisateur et la possibilité de partage entre individus.

CVD : Contrôle de Vitesse Descendante.

PWM : La modulation de largeur d'impulsions (MLI ; en anglais : **Pulse Width Modulation**, soit **PWM**), est une technique couramment utilisée pour synthétiser des signaux pseudo analogiques à l'aide de circuits à fonctionnement tout ou rien, ou plus généralement à états discrets.

CODE ARDUINO

```
#define buzz 10
```

```
const int ledPin = 13;
```

```
const int ledPin1 = 12;
```

```
const int ldrPin = A0;
```

```
const int ldrPin1 = A1;
```

```
const int echoPin = 4;
```

```
const int pingPin = 5;
```

```
//moteur1 gauche
int motor1=9;
int motor2=8;
//moteur2 droite
int motor3=7;
int motor4=6;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  pinMode(ledPin1, OUTPUT);
  pinMode(ldrPin, INPUT);
  pinMode(ldrPin1, INPUT);
  pinMode(pingPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  pinMode(motor3, OUTPUT);
  pinMode(motor4, OUTPUT);
  pinMode(buzz, OUTPUT);
}
void loop() {
  int ldrStatus = analogRead(ldrPin);
  int ldrStatus1 = analogRead(ldrPin1);
  int ledPinState = digitalRead(ledPin);
  int ledPin1State = digitalRead(ledPin1);
```

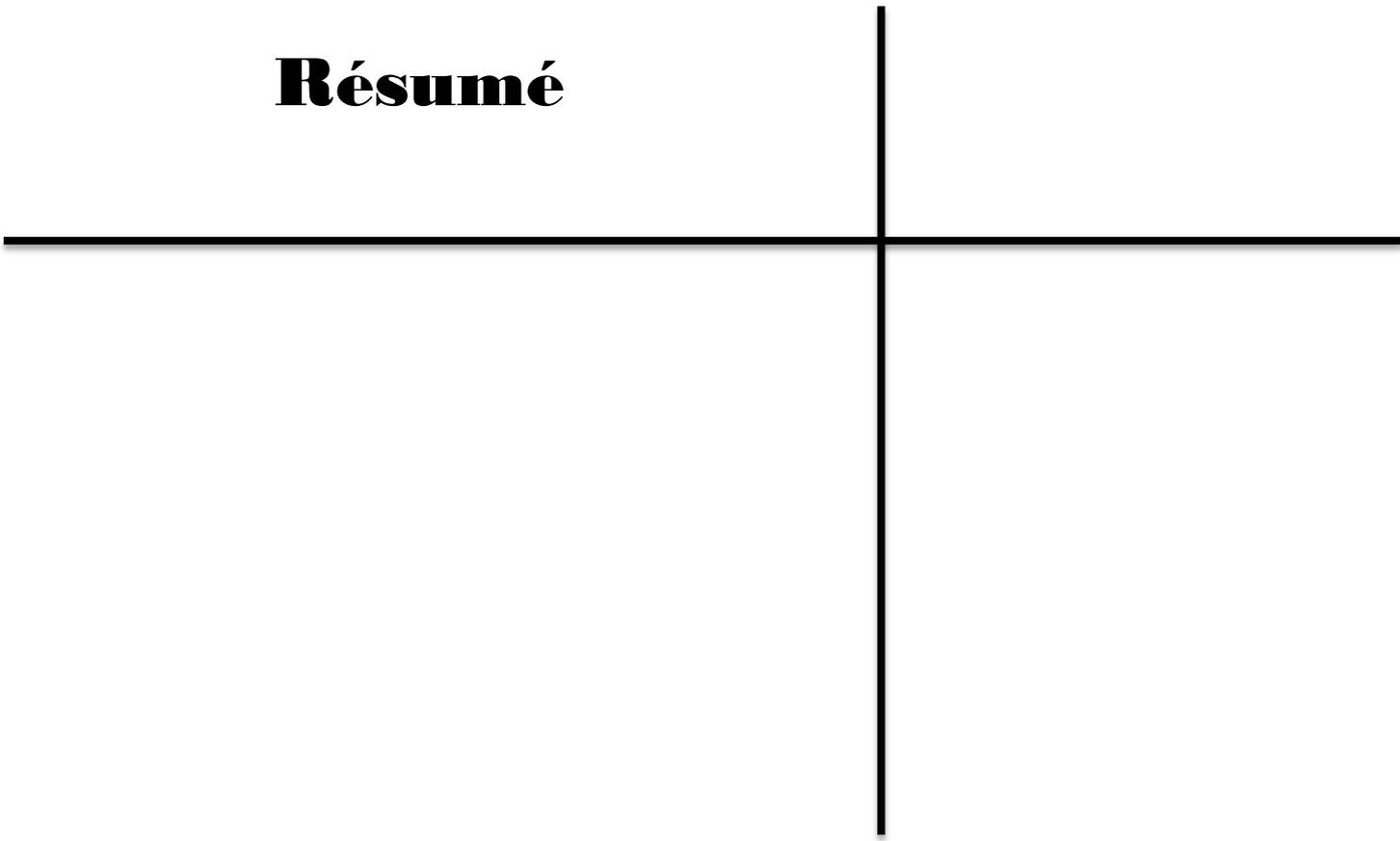
```
long duration, inches, cm;
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
delayMicroseconds(10);
    digitalWrite(pingPin, LOW);
duration = pulseIn(echoPin, HIGH); // using pulsIn function to determine total time
inches = microsecondsToInches(duration); // calling method
cm = microsecondsToCentimeters(duration); // calling method
Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
if (ldrStatus >=300){
digitalWrite(ledPin, HIGH );
}
else{
digitalWrite(ledPin, LOW);
}
if (ldrStatus1 >=300){
digitalWrite(ledPin1, HIGH );
}
else{
digitalWrite(ledPin1, LOW);
}
```

```
{  
  if(cm <= 30){  
digitalWrite(buzz,HIGH);  
  Serial.println("");  
  Serial.println("Obstacle STOP");  
    digitalWrite(motor1, LOW);  
digitalWrite(motor2, LOW);  
digitalWrite(motor3,LOW );  
digitalWrite(motor4,LOW );  
  }  
  else {  
digitalWrite(buzz,LOW);  
if((ldrStatus < 300)&& (ldrStatus1 < 300)){  
  Serial.println("");  
  digitalWrite(motor1,LOW );  
  digitalWrite(motor2,LOW );  
  digitalWrite(motor3,LOW );  
  digitalWrite(motor4,LOW);  
  }  
  if((ldrStatus >=300) && (ldrStatus1 >=300)){  
    Serial.println("Passage");  
    Serial.println("Marche Avant");  
    Serial.println("");  
    digitalWrite(motor1, HIGH);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3,HIGH );  
    digitalWrite(motor4,LOW );
```

```
}  
  
if((ldrStatus >=300) && (ldrStatus1 < 300)){  
    Serial.println("Passage");  
    Serial.println("Marche à Droite");  
    Serial.println("");  
    digitalWrite(motor1, HIGH);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3,LOW );  
    digitalWrite(motor4,LOW );  
}  
  
if((ldrStatus < 300) && (ldrStatus1 >=300)){  
    Serial.println("Passage");  
    Serial.println("Marche à Gauche");  
    Serial.println("");  
    digitalWrite(motor1, LOW);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3,HIGH );  
    digitalWrite(motor4,LOW ); }  
}  
}  
}  
  
long microsecondsToInches(long microseconds) // method to covert microsec to inches  
{  
    return microseconds / 74 / 2;  
}  
  
long microsecondsToCentimeters(long microseconds) // method to covert microsec to  
centimeters
```

```
{  
    return microseconds / 29 / 2;  
}
```

Résumé



Résumé

دراسة و انجاز روبوت تابع للخطوط يتم التحكم فيه بواسطة بطاقة أردوينو

الملخص في السنوات الاخيرة شهد النظام الداخلي تطورا ملحوظا بفضل التقنيات الإلكترونية المتقدمة، في هذا العمل يقترح إنجاز نظام مستقل (روبوت) يسمح باتباع خط أسود مرسوم على سطح أبيض. وهذا الروبوت يعتمد على أجهزة الاستشعار من نوع مقاوم ضوئي لتحديد الخط ، وبالتالي يساعده على البقاء في المسار الصحيح وهو مدفوع بمحركات التيار المستمر. كما يتم استخدام واجهة الأردوينو لتنفيذ خوارزميات لتوجيه الروبوت للتحرك على طول الخط..بالإضافة إلى ذلك، يستطيع ان يتحسس المعوقات باستخدام مسبار الموجات فوق الصوتية وإذا لم يكن هناك عائق على مسافة معينة أمامه يواصل التقدم في المسار.

الكلمات الجوهرية متابع، مسار، مستشعر، الأردوينو، روبوت، محرك، عائق، الموجات فوق الصوتية

Study and realization of a line follower robot controlled by a card ARDUINO

Abstract The embedded system has evolved considerably over recent years thanks to advanced electronic technologies. In this work, it's proposed to achieve a standalone system (robot) that's able to follow a black run on a white surface. This robot is based on sensors of photoresistor type to identify the line helping the robot to stay on the track and is driven by DC motors. The ARDUINO UNO interface is used to implement algorithms that can guide the robot to move along the line. In addition, an ultrasound probe is used to detect the presence of an obstacle in front of the follower. If no obstacle is detected at a specific distance in front of it, it continues to move forward.

Keywords Follower, Track, Sensor, Arduino, Robot, Motor, Obstacle, Ultrasound.

Etude et réalisation d'un robot suiveur de ligne contrôlé par une carte ARDUINO

Résumé Le système embarqué a considérablement évolué ces dernières années grâce aux techniques avancées de l'électronique. Dans ce travail de mémoire il est proposé la réalisation d'un système autonome permettant de suivre une ligne noire qui est tracée sur une surface de couleur blanche. Ce robot est basé sur des capteurs de type photorésistance pour identifier la ligne, aidant ainsi le robot à rester sur la piste et il est entraîné par des moteurs à courant continu. L'interface ARDUINO UNO est utilisée pour implémenter des algorithmes permettant de guider le robot pour se déplacer le long de la ligne. De plus, une sonde à ultrasons est utilisée pour détecter la présence d'un obstacle devant le suiveur. Si aucun obstacle n'est détecté à moins d'une distance donnée devant lui, il continue d'avancer en ligne.

Mots clés Suiveur, Ligne, Capteur, Arduino, Robot, Moteur. Obstacle, Ultrason.