

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE AKLI MOHAND OULHADJ-BOUIRA



Faculté des Sciences et Sciences Appliquées
Département de génie électrique

Mémoire de fin d'étude

Présenté par :

MORSLI Abdellah

ZERKI Sami

En vue de l'obtention du diplôme de **Master en :**

Filière : ELECTRONIQUE

Option : électronique des systèmes embarqués

Thème :

Commande gestuelle à distance d'un robot mobile

Devant le jury composé de :

ASRADJ Zahir	MCA	UAMOB	Président
AIT ABBAS Hamo	MCB	UAMOB	Encadreur
HAROUN Smail	MCB	UAMOB	Encadreur
BOUZIDA Ahcene	MCA	UAMOB	Examineur

Année Universitaire 2019/2020

Remerciement

Nous tenons d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

C'est avec une profonde reconnaissance et considération particulière que nous remercions :

✓ **Nos encadreurs Mr : AIT ABBAS Samou et Mr : SAROUIN Smail, pour leurs précieux conseils, leurs grandes bienveillances et leurs aides durant toute la période du travail.**

✓ **Tous les professeurs, on voudrait aussi les remercier qui ont contribué à notre formation.**

✓ **Tous nos amis et collègues Nous remercions aussi pour leurs aides, leurs patiences, leurs compréhensions et leurs encouragements.**

Enfin, nous exprimons toute notre gratitude à toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

MERCI A TOUS

Dédicace

C'est grâce à Allah seul que j'ai pu achever ce travail. Je le dédie à :

Mes très chers parents, que je ne remercierai jamais assez pour tout ce qu'ils ont fait pour moi et sans eux je ne serai pas arrivé là où j'en suis aujourd'hui.

A mes chers frères et sœurs Djamel, Fatiha, Kamel, Malika, Madjid, Rachid et Abdennour sans oublier les petits anges Zahra, Fahima, Aymen et Abd El Djalil.

A Toute la famille Morsli.

Mon cher ami, mon binôme Zekri Sami avec qui j'ai partagé des moments difficiles ainsi que des moments de joie tout au long de mon cursus.

A mes amis, Massi, Ali, Arezki, Makhlouf, Akli, Rachid, Abdou, Kahina, Saida, Amel et mes collègues de la promotion de génie électrique 2019-2020 nominativement.

A tous mes professeurs à qui je témoigne gratitude pour leur soutien, leur générosité à qui je dois un profond respect et ma loyale considération.

Et en dernier, un grand merci à toutes et celles et tous ceux qui d'une manière ou d'une autre m'ont aidé et soutenu.

MORSLI Abdellah.

Dédicace

C'est grâce à Allah seul que j'ai pu achever ce travail. Je le dédie à :

Mes très chers parents, que je ne remercierai jamais assez pour tout ce qu'ils ont fait pour moi et sans eux je ne serai pas arrivé là où j'en suis aujourd'hui.

À mes chères sœurs

À Toute la famille ZEKRI.

Mon cher ami, mon binôme MORSLI Abdellah avec qui j'ai partagé des moments difficiles ainsi que des moments de joie tout au long de mon cursus.

À mes amis, Massi, Ali, Arezki, Makhlouf, Akli, Rachid, Abdou et mes collègues de la promotion de génie électrique 2019-2020 nominativement.

À tous mes professeurs à qui je témoigne gratitude pour leur soutien, leur générosité à qui je dois un profond respect et ma loyale considération.

Et en dernier, un grand merci à toutes et celles et tous ceux qui d'une manière ou d'une autre m'ont aidé et soutenu.

ZEKRI Sami.

Sommaire

Remerciement

Dédicaces

Sommaire

Liste des figures	i
Liste des tableaux	iii
Liste des Acronymes	iv
Abréviations.....	iv
Liste des symboles	iv
Résumé	v
Introduction générale.....	1

Chapitre I : Etats de l'art sur les Robots

I.1.	Introduction.....	3
I.2.	Historique.....	3
I.3.	Définition de Robot.....	6
I.4.	Les composants d'un robot	7
I.5.	Types de robots	8
	I.5.1. Robots manipulateurs	8
	I.5.2. Robots mobiles.....	8
I.6.	Domaine d'application de la robotique.....	13
	I.6.1. Industriel	13
	I.6.2. Le domaine militaire	13
	I.6.3. La santé.....	14
	I.6.4. Utilisation civile.....	14
I.7.	Avantages et inconvénients	15
	I.7.1. Avantages.....	15
	I.7.2. Inconvénients	15
I.8.	Conclusion	16

Chapitre II : Description de la partie Soft et Hard utilisées

II.1. Introduction	17
II.2. Définition du module Arduino	17
II.2.1. Pour quoi l'Arduino	17
II.2.2. Les gammes d'Arduino.....	18
II.3. Arduino UNO	19
II.3.1. Présentation de la carte	19
II.4. Le module Accéléromètre ADXL345-EP	29
II.4.1. Description Générale	29
II.4.2. Schéma fonctionnel	30
II.4.3. Communication	31
II.4.4. Configuration et le calibrage	32
II.4.5. Traitement des données	32
II.5. Modules Bluetooth	32
II.5.1. Module Bluetooth HC-05	33
II.6. Pilote de moteurs L293 D.....	34
II.6.1. Fonctionnement de L293 D	35
II.6.2. Description des broches.....	36
II.7. Moteurs à courant continu	36
II.7.1. Constitution	37
II.7.2. Principe de fonctionnement	37
II.8. Conclusion.....	38

Chapitre III : Réalisation d'un prototype expérimental du Robot gestuel

III.1. Introduction	39
III.2. Principe de fonctionnement d'un Robot à commande gestuelle	39
III.3. Choix technique des composants utilisés.....	40
III.4. Etapes de la réalisation	43

III.5. Schéma synoptique général	43
III.5.1. Bloc d'alimentation	45
III.5.2. Bloc de commande	46
III.5.3. Bloc de communication.....	46
III.5.4. Bloc opératif	47
III.6. Mise en œuvre	47
III.6.1. Partie émettrice.....	47
III.6.2. Partie réceptrice.....	52
III.7. Conclusion.....	55
Conclusion générale	56
Bibliographie	
Annexes	

Liste des figures

Figure I.1 : Canard de jacques de Vaucanson 4

Figure I.2 : La clepsydre 4

Figure I.3 : UNIMATE : le premier robot industriel..... 4

Figure I.4 : Flex picker..... 5

Figure I.5 : Robonaut (R2B) 6

Figure I.6 : robot mobile a chenille 9

Figure I.7 : robot mobile à pattes 10

Figure I.8 : Robot mobile de type unicycle 10

Figure I.9 : Robot mobile de type tricycle 11

Figure I.10 : Robot mobile de type omnidirectionnel 11

Figure I.11 : Robot mobile de type voiture 12

Figure I.12: le robot dans l'industrie automobile 13

Figure I.13: Le robot militaire..... 14

Figure I.14: le robot dans la médecine 14

Figure I.15: yukitaro le robot déneigeure 15

Figure II.1 : module Arduino UNO. 19

Figure II.2 : Constitution de la carte Arduino Uno 20

Figure II.3 : Microcontrôleur ATmega328..... 21

Figure II.4 : Entrées-sorties numériques de l'Arduino Uno..... 23

Figure II.5 : Entrées analogiques de l'Arduino Uno 24

Figure II.6 : Interface IDE Arduino 27

Figure II.7 : Paramétrage de la carte 28

Figure II.8 : Les étapes de téléchargement du code 29

Figure II.9 : Le module Accéléromètre ADXL345 30

Figure II.10 : Schéma fonctionnel de ADXL 345[7] 31

Figure II.11 : Photo et schéma qui figurer les différentes broches de module Bluetooth..... 34

Figure II.12 : Pilote de moteurs L293D 35

Figure II.13 : D'éférénts broches du module L293 D 35

Figure II.14 : Moteur à courant continu (DC 12V)..... 36

Figure III.1 : chaine de commande de robot..... 40

Figure III.2 : Schéma synoptique générale de projet 44

Figure III.3 : schéma électrique de la partie émettrice..... 45

Figure III.4 : Schéma électrique de circuit récepteur	46
Figure III.5 : présentation Fritzing de la partie émettrice.....	48
Figure III.6 : Organigramme de la commande émettrice.....	49
Figure III.7 : position de la main pour la commande stop.....	50
Figure III.8 : position de la main pour la commande marche en avant	50
Figure III.9 : position de la main pour la commande marche arrière	51
Figure III.10 : position de la main pour la commande tourne à gauche	51
Figure III.11 : position de la main pour la commande tourne à droite	52
Figure III.12 : Schéma de circuit récepteur sur Fritzing	52
Figure III.13 : organigramme de la partie réceptrice	53

Liste des tableaux

Tableau I.1: Comparaison des différents types de robots mobiles à roues 12
Tableau II.1 : Caractéristiques de la carte Arduino Uno. 20
Tableau III.1 : liste des composants utilisés 41
Tableau III.2 : entrée activée par geste 54

Liste des Acronymes

Abréviations :

JC : Jésus Christ

EEPROM : Electrically Erasable Programmable Read-Only Memory

Ko : Kilo octet

RAM : Read only memory

V : volt

MCU : microcontroler unit.

LED : light emitting diode

LCD : Liquid cristal

display

PWM : Pulse Width Modulation.

WIFI : Wireless Fidelity.

SPI : Serial Peripheral Interface.

GFSK : Gaussian Frequency-Shift Keying.

GHz : Giga hertz.

DC : Direct Current.

GND : Ground

PC : Personal computer

C : commande.

1- Liste des symboles :

Symbols	Description	Unité
V	Tension	Volte
R	Resistance	Ohm(Ω)
I	Courant	Milliampere(mA)
F	Frequence	Hertz (Hz)
M	Distance	Mètre(m)

Résumé :

Dans notre projet, nous avons des fonctionnalités spéciales grâce auxquelles notre robot surmonter tant de problèmes s'il est développé plus avant il peut être utilise pour diverses applications. Un robot a commandé gestuelle avec microcontrôleur Arduino Uno a été conçu dans ce travail qui peut être Controller par des gestes de la main humain. Cela nécessite de porter un petit appareil émetteur sur notre main inclut un accéléromètre qui transmet des commandes particulières au robot pour se déplacer selon les gestes de la main de l'utilisateur et sur le récepteur au robot.

Les mots clés : Robot, gestes, Arduino, ADXL345, PWM, commande, IDE, Actionneurs, Bluetooth, programmation, microcontrôleur, émetteur, récepteur, contrôle,

Abstract :

In our project we have added special features by which our robot can overcome so many problems in industry. If it is further developed then it can be used for military application. A Gesture Controlled robot with Arduino Uno microcontroller has been designed in this work, which can be controlled by human hand gestures. This requires to wear a small transmitting device of our hand included an accelerometer, which transmits particular commands to the robot to move according to the users hand gesture and one receiver at the robot.

ملخص :

لقد أضفنا في مشروعنا مميزات خاصة يستطيع الروبوت من خلالها التغلب على العديد من المشكلات في الصناعة. إذا تم تطويره بشكل أكبر ، نتمكن استخدامه للتطبيقات العسكرية. تم تصميم روبوت يتم التحكم فيه بالإيماءات مع متحكم Arduino Uno في هذا العمل ، والذي يمكن التحكم فيه عن طريق إيماءات اليد البشرية يتطلب ذلك ارتداء جهاز إرسال Arduino Uno صغير في يدينا ينضم من مؤناس نسرار ، والذي يوزل أوامر معينة إلى الروبوت للتحرك وننقل الإيماءة يد المتحكم من وجهاز استقبال واحد عند الروبوت

Introduction

générale

Introduction générale

Aujourd'hui, l'électronique classique 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 est 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. On va en effet mettre en pratique nos connaissances en électronique embarquée qui est un sous - domaine de l'électronique à celles de la programmation [8].

L'objectif de ce travail est la réalisation d'un robot mobile à commande gestuelle à distance. Il est constitué de deux parties, un robot mobile à roues, et son unité de commande gestuelle à distance. La communication entre les deux parties est assurée par des modules de communication Bluetooth. Les deux unités embarquent deux cartes Arduino qui permettent de contrôler l'ensemble du système [8].

Le code à implémenter sur la carte Arduino de l'unité de commande doit détecter les gestes de la main en recevant les informations venant de l'unité de mesure inertielle (accéléromètre ADXL). Par la suite, ces informations sont traduites en commandes d'avance, de recule et de rotation qui seront envoyer au robot a traves la communication Bluetooth. De l'autre côté, le rebot reçoit les commandes et les convertis en signaux et les envois aux drivers des moteurs afin de réaliser la commande requise [8].

Dans ce manuscrit, nous présentons l'essentiel de ce travail qui sera réparti comme suit :

Le premier chapitre sera une brève présentation sur les robots mobiles et la robotique en général.

Le deuxième chapitre sera consacré à une étude approfondie de la partie matérielle du système à réaliser en présentant ces différents éléments constitutants.

Dans le troisième chapitre, nous abordons la réalisation pratique du robot et sa commande, en présentant les circuits, les organigrammes des codes, et l'implémentation pratique de l'ensemble du système.

Enfin, on terminera avec une conclusion générale qui résumera l'intérêt de notre étude.

Chapitre I

Etats de l'art sur

les Robots

I.1. Introduction

L'homme a depuis toujours été fasciné par la possibilité de pouvoir recréer la vie. Désir profond de jouer les créateurs, ou bien tentation de la mystification, ou encore simple jeu de copie de l'existant, les divers hommes d'airain, androïdes ou robots, créés dans la littérature, au cinéma ou dans la réalité, ont depuis la haute antiquité répondu à nos fantasmes.

Il est cependant nécessaire de faire la distinction entre deux types d'automates : les automates cycliques qui exécutent un même mouvement de façon répétitive, et les automates à programme dont le mouvement est beaucoup plus riche. Dans le premier cas, le mouvement est obtenu par une mécanique figée qu'il est très difficile de modifier. En revanche, dans le deuxième cas les mouvements sont programmés à partir d'un dispositif central qui décrit leur séquence et éventuellement leur amplitude [1].

I.2. Historique

Depuis la nuit des temps, l'homme a cherché à se faire remplacer pour des tâches spécifiques pouvant être dangereuses, fastidieuses ou longues. Le concept du robot a été établi par de nombreuses réalisations historiques créatives tel que : l'horloge à l'eau de clepsydre introduite par les babyloniens (1400 avant JC), le théâtre de l'automate du héron d'Alexandrie (une centaine d'années après JC) et les automates de l'arabe al-djazari (1200 après JC).

Un de ces spécimens très évolués fut présenté par Jacques de Vaucanson en 1738 : il représentait un homme jouant d'un instrument de musique à vent. Jacques créa également un automate représentant un canard mangeant et refoulant sa nourriture après ingestion de cette dernière.

Cependant, l'apparition du robot physique a dû attendre l'arrivée de technologies avancées et plus adaptées au cours du 20^{ème} siècle [2].

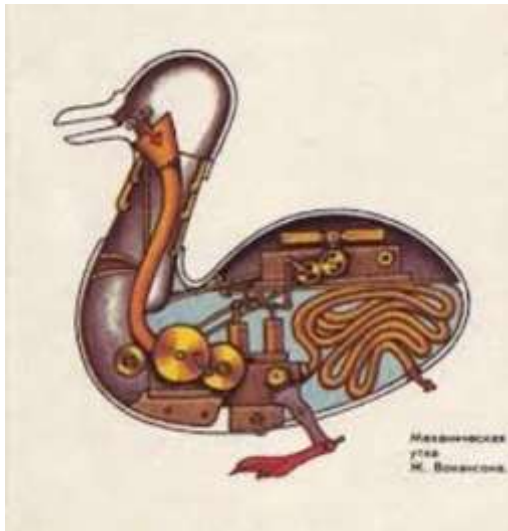


Figure I.1 : Canard de Jacques de Vaucanson [14].

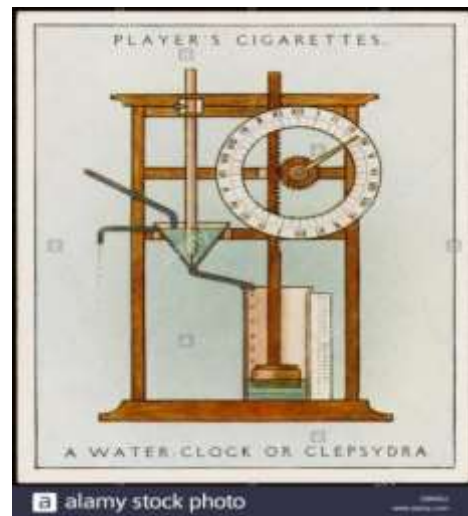


Figure I.2 : La clepsydre [13].

- **1920 : l'origine** du mot robot provient de la langue tchèque dans laquelle son ancêtre "robota" signifie travail forcé. Il a été introduit en 1920 par l'écrivain tchèque Karel Čapek dans la pièce de théâtre « Rossum's Universal Robots ».

En 1954 : George Charles Devol invente le premier robot industriel programmable. Il fonde en 1956 la première entreprise de robots, unimation, avec l'ingénieur Joseph Engelberger. Leur robot, appelé unimate, était capable de manipuler des objets avec des actionneurs hydrauliques [3].



Figure I.3 : UNIMATE : le premier robot industriel [15].

- **En 1969** : victore scheinman de l'université de stanford inventa un bras articulé sur 6 axes. scheinman vendra son concept a unimation plus tard .
- **1973** : kuka une entreprise allemande créé le premier robot industriel avec 6 axes contrôlés electromécaniquement : le famulus.
- **En 1978** : unimation propose le puma, un robot d'assemblage encore utiliser de nos jours dans plusieurs laboratoires de recherche.
- **1981** : takeo kanade construit le premier robot muni de moteurs installés directement aux articulations du bras. Cela a permis des robots plus rapides et plus précis qu'auparavant.
- **1998** : ABB, suède, a développé le flexpicker, le robot de préparation de commandes le plus rapide au monde basé sur le robot delta développé par reymond clavel, institut fédérale de technologie de Lausanne(epfl) [4].



Figure I.4 : Flex picker .

- **2003** : première mission d'exploration de la planète mars impliquant deux robots mobiles, spirit et opportunité.
- **2011** : Robonaut (R2B) premier robot humanoïde envoyer dans l'espace construit et conçu par la NASA au Johnson Space Center (JSC) à Houston (Texas), en collaboration avec General Motors (GM) et Oceaneering[3].



Figure I.5 : Robonaut (R2B) .

I.3. Définition de Robot

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) possédant des capteurs, un système logique et des actionneurs. Il exécute une ou plusieurs tâches de manière autonome conformément à un programme préalablement établi et modifiable [6].

Un robot est un appareil dirigé à l'aide d'un système de commande automatique à base de microprocesseur et qui est conçu pour accomplir une tâche précise, dans le domaine industriel, scientifique ou encore domestique.

Par ailleurs, la robotique fait appel à plusieurs spécialités à savoir :

- **Mécanique** : conception, réalisation, modélisation des robots.
- **Electronique** : mise en place de composants pour les robots en vue de les commander à distance en utilisant des concepts de télécommunication.
- **Informatique** : création de programmes destinés à la gestion du transfert d'informations entre les différents composants du robot.
- **Automatique** : commande, calibration des capteurs, des effecteurs, identification des paramètres.
- **Traitement du signal** : analyse des informations enregistrées par les capteurs du robot.
- **Mathématique** : modèles mathématiques pour la prise de décision ou/et l'apprentissage, calcul de trajectoire, localisation, planification. [5]

I.4. Les composants d'un robot

Un robot, en tant que système, se compose des éléments, qui sont intégrés ensemble pour former un ensemble, la plupart des robots contiennent les éléments suivants :

- **Manipulateur** : c'est le corps principal du robot qui comprend les jonctions, les articulations, et d'autres éléments de structure du robot.
- **Actionneur** : les actionneurs sont les muscles de manipulateur. Le contrôleur envoie des signaux aux actionneurs, qui à son tour, déplacent les articulations du robot et des jonctions.
- **Les capteurs** : Les capteurs sont utilisés pour recueillir des informations sur l'état interne du robot ou pour communiquer avec l'environnement extérieur. Comme chez l'humain, les dispositifs de commande de robot doivent connaître l'emplacement de chaque lien du robot afin de connaître la configuration du robot. Les robots sont équipés de dispositifs sensoriels externes comme un système de vision, le toucher est les capteurs tactiles, synthétiseur de parole, et grâce à eux le robot peut communiquer avec le monde extérieur [7].
- **Le contrôleur** : Le contrôleur est plutôt proche de votre cervelet, même si elle n'a pas la puissance du cerveau, il contrôle toujours vos mouvements. Le contrôleur reçoit les données de l'ordinateur (le cerveau du système), commande les mouvements des actionneurs, et coordonne les mouvements avec les informations envoyées par les capteurs [7].
- **Processeur** : Le processeur est le cerveau du robot. Il calcule les mouvements des articulations du robot, détermine combien et à quelle vitesse, chaque joint doit se déplacer pour atteindre l'emplacement et la vitesse souhaitée, et supervise les actions coordonnées du contrôleur et des capteurs. Dans certains **systèmes**, le contrôleur et le processeur sont intégrés ensemble en une seule unité, et dans d'autres cas se sont les unités séparées [7].
- **Logiciel** : Trois groupes de logiciels sont regroupés dans un robot. L'un est le système d'exploitation qui exploite le processeur. Le second est le logiciel robotique qui calcule la motion nécessaire de chaque joint du robot basée sur des équations cinématiques. Ces informations sont envoyées au dispositif de commande. Ce logiciel peut être à différents niveaux, de la langue de la machine aux langues sophistiquées utilisées par les robots modernes. Le troisième groupe est la collection d'applications orientées vers les routines et les

programmes développés pour utiliser le robot ou ses périphériques pour des tâches spécifiques telles que l'assemblage, le chargement des machines, la maintenance et les routines de vision.

I.5. Types de robots

On peut classer les robots en deux grandes catégories selon leurs types de tâche :

I.5.1. Robots manipulateurs

Robots ancrés physiquement à leur place de travail et généralement mis en place pour réaliser une tâche précise ou répétitive. Les manipulateurs sont des systèmes mécaniques destinés à exécuter de manière autonome des tâches répétitives dans un environnement fixe et ordonné et dont les mouvements enregistrés dans une mémoire se répètent de manière cyclique, aucune fonction de décision ne lui permet de réagir dans le cas d'un événement inattendu, c'est pour ça qu'ils sont appelés (automate). Les domaines d'application des manipulateurs sont : la peinture, le soudage par position, la manipulation des objets...etc [8].

I.5.2. Robots mobiles

Un robot mobile est celui qui peut se déplacer dans son environnement de façon indépendante. Pour ce faire, le robot doit pouvoir naviguer, et la portée et la précision de navigation requise varient en fonction de la taille du robot et du type de sa tâche.

I.5.2.1. L'architecture des robots mobiles

L'architecture des robots mobiles est principalement basée sur trois structures comme suit :

A. Structure mécanique :

Elle assure le mouvement du robot par des roues motrices placées selon le type de mouvement et la précision de la tâche voulue

B. Structure instrumentale

Un robot est équipé d'un certain nombre de capteurs de sécurité afin de leur donner une certaine connaissance de l'environnement. Selon l'application, les capteurs peuvent être : (capteurs ultrasons, capteurs infrarouges, capteurs lasers, les gyroscopes...).

C. Structure informatique

Une commande numérique est impérative, afin de bien analyser les différentes informations, soit du système de perception ou de localisation. Cette commande peut être à base d'un microprocesseur ou microcontrôleur [9].

I.5.2.2. Classes de robots mobiles**I.5.2.2.1. Robots mobiles à chenilles**

Lorsque le terrain est accidenté, les roues perdent leur efficacité de locomotion. Ceci limite la capacité d'évolution du robot mobile équipé de ce type de système de locomotion. Dans ces conditions, les chenilles sont plus intéressantes, car elles permettent d'augmenter l'adhérence au sol et de franchir des obstacles plus importants. [9]

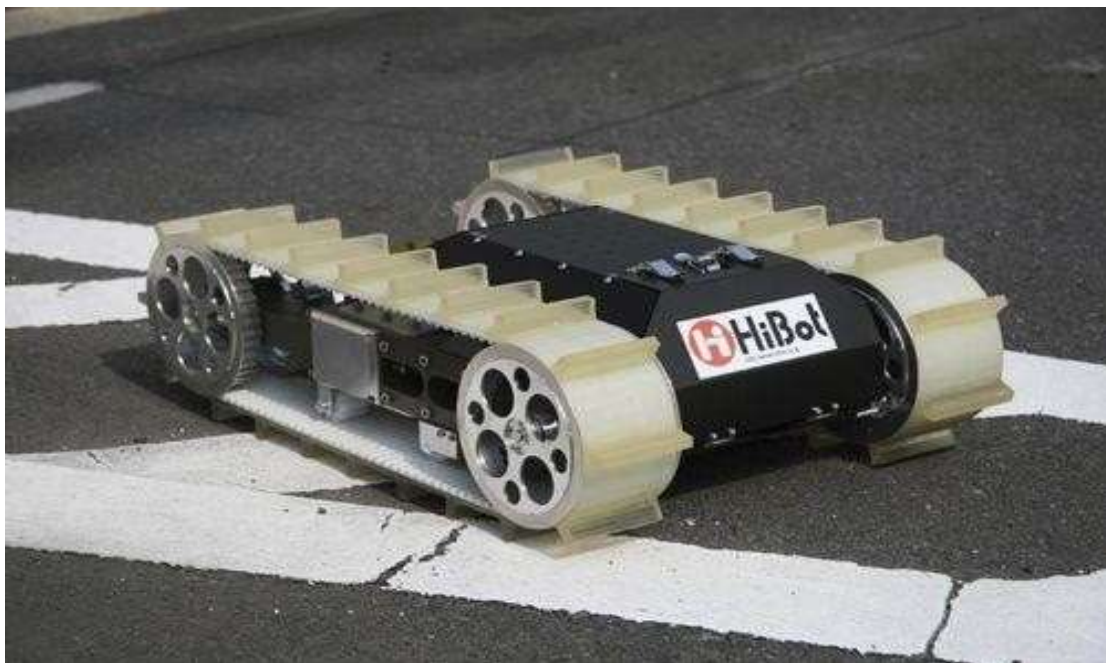


Figure I.6 : robot mobile a chenille

I.5.2.2.2. Robots mobiles à pattes

Dans la situation où le terrain est encore plus incertain, avec des grandes Différences de hauteur comme un escalier ou un terrain très accidenté, les Robots à roues ou à chenilles ne sont plus efficaces. Pour surmonter ces problèmes les Chercheurs en Robotique mobile ont fait recours aux robots mobiles à pattes ayant des Points d'appui discrets sur le terrain. [9]

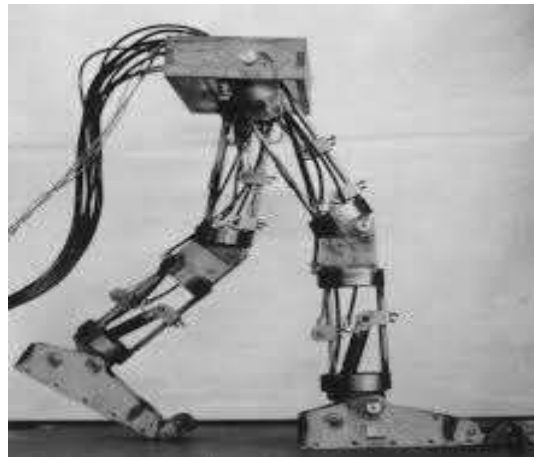


Figure I.7 : robot mobile à pattes.

Robots mobiles à roues

Compte tenu de la simplicité du mécanisme de locomotion, ce type de robot est le plus répandu actuellement. La plupart des robots mobiles à roues opèrent dans des sites aménagés : des sites industriels ou des environnements intérieurs, mais il existe également des applications en environnement extérieur.

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées.

a) Robot mobile de type unicycle

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles (roue libre) pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites [10].

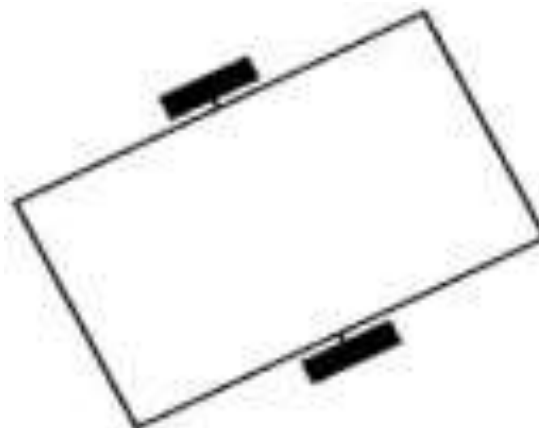


Figure I.8 : Robot mobile de type unicycle.

b) Robot mobile de type tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable, C'est un robot non-holonyme. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable [10].

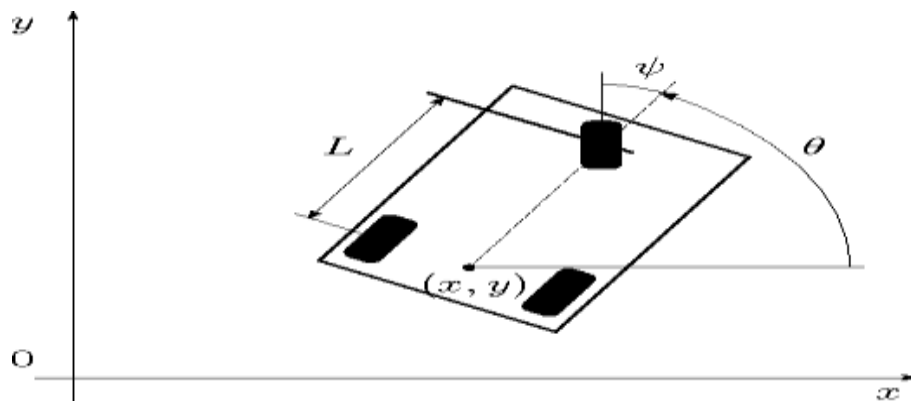


Figure I.9 : Robot mobile de type tricycle.

c) Robot mobile de type omnidirectionnel

Un robot mobile omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral. L'énorme avantage du robot omnidirectionnel est qu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande [10].

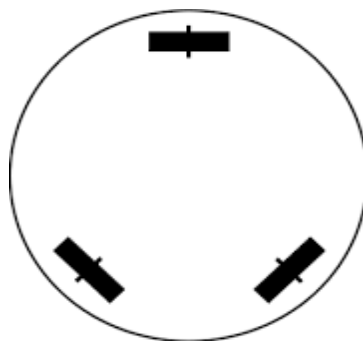


Figure I.10 : Robot mobile de type omnidirectionnel.

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe. Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé [10].



Figure I.11 : Robot mobile de type voiture.

A. Comparaison des différents types de robots à roues

Nous pouvons observer dans le tableau ci-dessous un récapitulatif des avantages et des inconvénients des différents types de robots à roues.

Tableau I.1: Comparaison des différents types de robots mobiles à roues.

Type du robot	Avantage	Inconvénient
Unicycle	Stable. Rotation sur lui-même. Complexité mécanique faible	Non-holonome
Tricycle	Complexité mécanique modérée	Non-holonome Peu stable Pas de rotation sur soi-même
Voiture	Stable Complexité mécanique modérée	Non-holonome Pas de rotation sur soi-même
Omnidirectionnel	Holonome. Stable. Rotation sur soi-même.	Complexité mécanique importante

I.6. Domaine d'application de la robotique

La robotique est un domaine en plein essor depuis quelques années, les évolutions technologiques, dépassent sans cesse nos espérances, permettent maintenant de réaliser des solutions technologiques s'adaptent au moindre problème.

Par conséquence, la robotique est utilisée dans des domaines extrêmement rigoureux et exigeantes. Nous allons explorer ces différents domaines.

I.6.1. Industriel

Le but premier des robots est de remplacer l'homme dans des activités fastidieuses ou onéreuses pour employer : les robots ont donc commencé à être utiliser dans les chaines d'assemblage industrielles. Donc ces chaines d'assemblages, en retrouve des robots soudeurs, manipulateurs, peintre...etc [11].



Figure I.12: le robot dans l'industrie automobile.

I.6.2. Le domaine militaire

Les robots sont de plus en plus utilisés dans le domaine militaire. En effet, la miniaturisation permet aujourd'hui de créer des robots discrets mais dotés de nombreux capteurs, ce qui idéale pour des missions d'espionnage ou d'éclairage [11].



Figure I.13: Le robot militaire.

I.6.3. La santé

Les robots commencent à être de plus en plus dans le domaine médical, qu'il s'agisse de simples échographies ou d'opérations chirurgicales plus délicates. En fait ces robots ne sont pas complètement autonomes mais ils assistent les médecins ou chirurgiens, jusqu'à permettre des opérations médicales à distance. On parle de chirurgie assistée (chirurgie) c'est-à-dire tous ce qui consiste à introduire les derniers outils des technologies informatiques et robotiques dans la partie médico-chirurgicale. Cette pratique de chirurgie assistée est émergente donc bien que peu répandue, elle est en phase de devenir la chirurgie de la future [11].



Figure I.14: le robot dans la médecine.

I.6.4. Utilisation civile

De leur polyvalence, de plus en plus de tâches sont confiées aux robots. Voici par exemple « yukitaro », un robot japonais conçu pour déneiger : il est équipé de deux caméras et d'un GPS, il circule dans la rue de manière autonome en engouffrant la neige avec sa bouche pelleuse, puis il la rejette sous forme de blocs de glace [11].



Figure I.15: yukitaro le robot déneigeure.

I.7. Avantages et inconvénients

I.7.1. Avantages

- L'utilisation de robot réduit les opérations à cœur ouvert grâce aux techniques mini-invasives
- Elle permet aussi de fonctionner par des incisions plus petites, car la miniaturisation des éléments chirurgicaux des robots et l'articulation des bras engendrent une réduction de la taille des incisions
- De plus, l'utilisation de robot élimine les tremblements de la main grâce à un filtre électronique.
- Pour les entreprises, les robots industriels apparaissent comme l'outil incontournable pour développer leurs entreprises.
- Ces robots, sont exceptionnels en termes de polyvalence, de fiabilité, de sécurité ou encore de rentabilité.
- Ces machines travaillent rapidement et peuvent réaliser des tâches répétitives tout au long de la journée mais aussi pendant la nuit [12].

I.7.2. Inconvénients

- Les mouvements ne sont pas complètement autonomes, car ils sont sous l'ordre de l'action du chirurgien.
- Il peut y avoir des risques tels qu'une panne ou une difficulté technique.

- Acheter un robot pour certaines entreprises peut s'avérer être un très gros investissement si bien que les banques ont du mal leur prêter de l'argent même si les robots assurent très souvent une bonne rentabilité au bout de plusieurs années.
- Les robots sont l'origine du chômage.
- Certains métiers risquent de disparaître [12].

I.8. Conclusion

Dans ce chapitre, on a parlé sur les racines de la robotique les différents types de robots en commençant par les robots manipulateurs jusqu'au robots mobiles, ou on a comparé entre ces différents types et spécialement les robots de type voiture qui sera notre thème de réalisation dans les prochains chapitres.

Enfin, on a abordé les différents domaines d'application de la robotique et ces multiples avantages pour l'être humain ainsi que ces inconvénients.

Chapitre II

Description de la

partie Soft et

Hard utilisée

II.1. Introduction :

L'objectif de ce chapitre est de mettre l'accent sur la description des différents matériels et logiciels qu'on a utilisés pour la réalisation, programmation d'un robot mobile à commande gestuelle.

Ce dispositif est basé principalement sur une carte Arduino UNO ATmega 328, cette dernière permet de commander un robot communiquant (le gant) via une connexion Bluetooth. Elle est aussi constituée d'une alimentation (pile 9v), module de contrôle de moteurs L293D, un accéléromètre ADXL345 et des moteurs a courant continue.

II.2. Définition du module Arduino

Arduino est la marque d'une plateforme de prototypage open-source qui permet aux utilisateurs de créer des objets électroniques interactifs à partir de cartes électroniques matériellement libres sur lesquelles se trouve un microcontrôleur (d'architecture Atmel AVR comme l'Atmega328p, et d'architecture ARM comme le Cortex-M3 pour l'Arduino Due). Sans tout ni connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe à l'aide de nombreuses ressources disponibles en ligne. Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations humain/machine ou environnement/machine. En outre, Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions [28].

L'Arduino se compose à la fois d'un composant électronique programmable (souvent appeler microcontrôleur) et d'un logiciel IDE (environnement de développement intégré) qui s'exécute sur ordinateur, utilisé pour écrire et télécharger le code informatique sur la carte physique.

II.2.1. Pour quoi l'Arduino

Il existe pourtant dans le commerce, une multitude de plateformes qui permettent de faire la même chose.

Notamment les microcontrôleurs « PIC » du fabricant Micro-chip. Nous allons voir pourquoi choisir l'Arduino.

a. Le prix

En vue des performances qu'elles offrent, les cartes Arduino sont relativement peu coûteuses, ce qui est un critère majeur pour le débutant.

b. La liberté

C'est un bien grand mot, mais elle définit de façon assez concise l'esprit de l'Arduino. Elle constitue en elle-même deux choses :

Le logiciel : gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.

Le matériel : cartes électroniques dont les schémas sont en libre circulation sur internet. Cette liberté a une condition : le nom « Arduino » ne doit être employé que pour les cartes « officielles ». En somme, vous ne pouvez pas fabriquer votre propre carte sur le modèle Arduino et lui assigner le nom « Arduino ».

Les cartes non officielles, on peut les trouver et les acheter sur Internet et sont pour la quasi-totalité compatibles avec les cartes officielles Arduino.

c. La compatibilité

Le logiciel, tout comme la carte, est compatible sous les plateformes les plus courantes (Windows, Linux et Mac), contrairement aux autres outils de programmation du commerce qui ne sont, en général, compatibles qu'avec Windows.

d. La communauté

La communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet. De plus, on trouve les références du langage Arduino ainsi qu'une page complète de tutoriels sur le site arduino.cc (en anglais) et arduino.cc (en français).

[16]

II.2.2. Les gammes d'Arduino [17]

Bien qu'il existe de nombreuses variétés de cartes Arduino sur le marché, il en existe une poignée qui sont les plus couramment vues. Les cartes Arduino les plus couramment utilisées par les ingénieurs et les amateurs comprennent :

- Arduino UNO
- Arduino Nano
- Arduino Due
- Arduino Méga

II.3. Arduino UNO

II.3.1. Présentation de la carte

L'UNO est sans doute l'Arduino le plus populaire. Il est alimenté par un processeur Atmega328 fonctionnant à 16 MHz, comprend 32 Ko de mémoire programme, 1 Ko d'EEPROM, 2 Ko de RAM, 14 E / S numériques, 6 entrées analogiques et des rails d'alimentation 5 V et 3,3 V.

L'Arduino UNO a un arrangement d'en-tête de broche qui devient rapidement la norme de l'industrie pour les cartes de développement, le rendant compatible avec la plupart des blindages de cartes de développement sur le marché.

Une prise d'alimentation est incluse sur l'UNO, lui permettant d'être alimenté par une verree murale externe. Une option VIN est également disponible pour connecter l'UNO aux batteries.

Les dimensions physiques de l'UNO (69 mm x 54 mm) en font une petite carte de développement qui peut facilement s'intégrer dans de nombreux projets et les quatre trous de vis permettent aux concepteurs de les mettre en place en toute sécurité [2].



Figure II.1 : module Arduino UNO.

II.2.2. Caractéristiques de la carte Arduino UNO

Le tableau II.1 décrit les différentes caractéristiques de la carte Arduino Uno (électriques, stockage et horloge...).

Tableau II.1 : Caractéristiques de la carte Arduino Uno.

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12 V
Tension d'alimentation (limite)	6-20 V
Broches E/S numériques	14(dont 6disposent d'une sortie (PWM)
Broches d'entrées analogiques	06 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (Attention : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction d'alimentation utilisée – 500mA max si port USB utilisé seul
Mémoire programme flash	32 Ko dont 0.5 Ko sont utilisé par le Bluetooth
Mémoire RAM	8 Ko
Mémoire EEPROM (mémoire volatile)	32KB
Vitesse d'horloge	16 MHz

II.2.3. Description de la carte Arduino UNO



Figure II.2 : Constitution de la carte Arduino Uno [20].

Les cartes Arduino Uno généralement se compose de deux partie une partie matérielle (hardware) et une partie programme (software) :

II.3.1.1. Partie matérielle

Généralement tous modules électroniques qui possèdent une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

A- Le microcontrôleur ATmega 328 :



Figure II.3 : Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure-II.3 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino [3].

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. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **Mémoire Flash** : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont boot loader de 0.5 ko).
- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme [19].

B- Les entrées et sorties

Le nombre d'entrées-sorties est variable selon les cartes Arduino. Toutefois leurs principes de fonctionnement se retrouvent quasiment à l'identique. Si la plupart des broches d'entrées/sorties disponibles à la périphérie des Arduino permettent d'émettre ou de recevoir une valeur numérique binaire (0 ou 1), elles offrent aussi des fonctions spécialisées et on peut, par programme, décider de la fonction allouée à une broche particulière. Passons en revue la nature des entrées-sorties d'un Arduino [33].

Quasiment toutes les broches d'un Arduino peuvent être programmées en entrée ou sortie numérique (et non les deux en même temps). Par exemple, sur l'Arduino Uno, il s'agit d'une part des broches numérotées de 0 à 13 mais également des broches A0 à A5. Une broche programmée ainsi peut être :

- **Une entrée :** Le programme peut lire une tension présente sur cette broche en utilisant `digitalRead(...)`. Comme cette tension est interprétée comme un chiffre binaire (0 ou 1), la datasheet du MCU de l'Arduino Uno garantit que toute tension inférieure à $0,3 \times V_{cc}$, V_{cc} étant égale à 5V, soit 1,5V sera comprise comme un 0 et que toute tension supérieure à $0,6 \times V_{cc}$, soit 3V, sera comprise comme un 1. Entre les deux, c'est flou. `digitalRead(...)` renverra de toutes façons un 0 ou un 1 mais de manière plus ou moins aléatoire et variable selon l'Arduino utilisé.
- **Une sortie :** Le programme peut écrire un chiffre binaire, au moyen de `digitalWrite(...)`, chiffre qui dans le programme sont nommées HIGH pour le 1 et LOW pour le 0, qui sera traduit en une tension de 5V pour le 1 et de 0V pour le 0. Attention toutefois, cette tension peut respectivement être plus basse ou plus haute si le courant qui est tiré de la broche commence à être important [33].

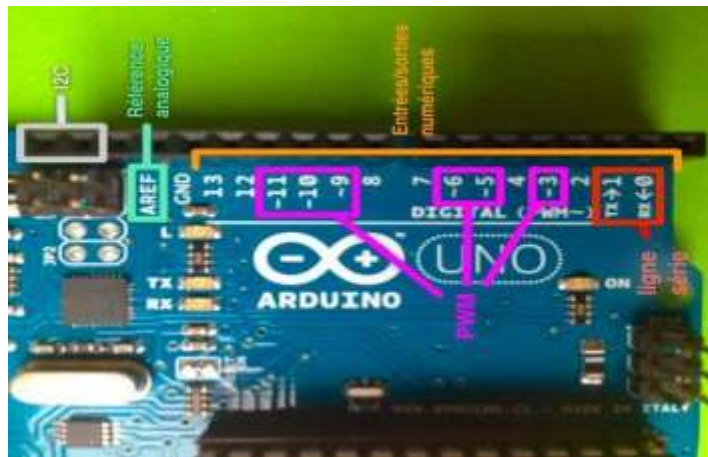


Figure II.4 : Entrées-sorties numériques de l'Arduino Uno

Il est préférable de ne pas dépasser une consommation de 20 mA sur une broche programmée en sortie et absolument nécessaire de ne pas dépasser 40 mA sous peine de risque la destruction de la sortie du MCU. Il s'agit aussi de ne pas dépasser au total 200 mA.

Les sorties numériques sont employées de nombreuses manières : alimentation d'une DEL pour un éclairage ou un témoin, commande d'un transistor, connexion à un afficheur LCD, etc.

Les entrées numériques permettent également de connecter de nombreuses choses : interrupteur, bouton poussoir, interrupteurs à lame souple REED mais aussi des capteurs de présence plus sophistiqués [21].

a. Les entrées analogiques

Les Arduino sont pourvus de 6, pour le module Uno, à 16 entrées analogiques, pour le Mega. Ce sont les broches étiquetées A suivie d'un nombre. Les tensions, toujours entre 0 et 5V, présentes sur ces broches, peuvent être numérisées via un convertisseur analogique-numérique ou ADC (Analog Digital Converter). La fonction `analogRead(...)` remplit ce rôle.

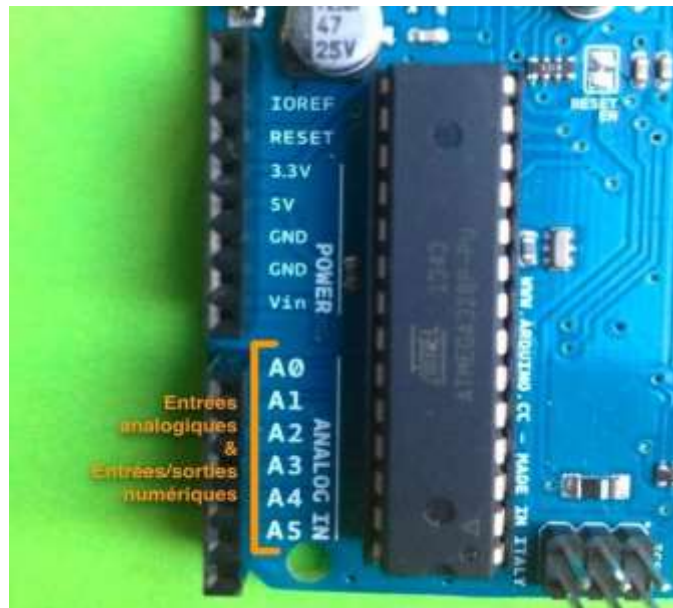


Figure II.5 : Entrées analogiques de l'Arduino Uno

Le convertisseur des Arduino à base d'AVR effectue une conversion sur 10 bits, c'est à dire qu'il convertit la tension en un nombre entier ayant une valeur de 0 à 1023. 0 correspond au 0V de la tension et 1023 au 5V. La résolution, c'est à dire la différence entre deux valeurs successives de la tension correspondant à une différence de 1 sur l'entier résultat, est donc d'environ 5mV [21].

Il est possible de changer la tension de référence de la conversion analogique-numérique, c'est à dire la tension comprise comme 1023. Le MCU des Arduino possède une référence de tension interne de 1,1V qui peut être sélectionné à la place de la tension d'alimentation. Cela permet une meilleure résolution, environ 1mV, sur les faibles valeurs de tension à convertir. Il est également possible de sélectionner la tension fournie sur la broche AREF comme tension de référence.

Plusieurs types de capteurs analogiques peuvent trouver place sur ces broches. On citera la mesure de courant traversant une résistance ou la mesure d'intensité lumineuse via une photo-résistance [5].

b. Les sorties PWM

Se diffèrent d'une carte Arduino à une autre comme dans la carte Arduino Uno, ont trouvent 6 broches qui peuvent être configurées comme sortie PWM et 15 broches pour le Méga. La configuration est effective dès que le programme utilise la fonction `analogWrite(...)` sur une des broches configurables en PWM.

Le rapport cyclique de la PWM est réglé via cette même fonction entre 0 (le signal est constamment à l'état bas, 0V) à 255 (le signal est constamment à l'état haut, 3,3V ou 5V selon l'Arduino, soit un rapport cyclique de 100%).

La PWM trouve son emploi lorsqu'il s'agit de commander un moteur électrique à courant continu, comme celui qui équipe nos locomotives, ou une intensité lumineuse, par exemple dans une DEL. Si le courant nécessaire à une DEL est compatible avec ce que peut fournir une sortie PWM, il n'est pas de même pour la commande d'un moteur. Dans ce dernier cas, il est nécessaire d'amplifier le signal via un transistor qui permettra également de passer des 3,3V ou 5V de l'Arduino aux 12V nécessaires pour nos engins [5].

C- La ligne série

Les 2 broches étiquetées TX et RX sont respectivement la ligne d'émission série (T pour Transmit, émettre) et la ligne de réception série (R pour recevoir). Cette ligne est principalement employée pour dialoguer avec l'ordinateur hôte, votre Mac ou votre PC. Des fonctions permettent d'afficher des messages dans une fenêtre de l'ordinateur hôte et de lire le clavier de l'ordinateur hôte.

Elle est également employée pour dialoguer avec certains modules comme les modules radio XBee ou certains modules WiFi ou encore Bluetooth [21].

a. Le bus I2C

L'I2C est l'un des bus de communication disponibles pour dialoguer soit avec un autre Arduino soit avec des circuits périphériques disposant de ce bus. 2 broches sont employées pour l'I2C, l'horloge et l'adresse/donnée. De nombreux circuits périphériques disposent d'une interface I2C. On peut citer les expenseurs d'entrées/sorties, circuits qui permettent d'ajouter des entrées/sorties numériques à l'Arduino et les générateurs de PWM qui permettent donc d'augmenter le nombre de PWM pilotables [32].

b. Le bus SPI

Enfin, le SPI est également un bus de communication destiné à dialoguer avec les circuits périphériques. À la différence de l'I2C, il est bidirectionnel. Il permet aussi un débit de communication plus important. 4 broches sont employées pour le SPI, l'horloge, les données en sortie, les données en entrée et la sélection du circuit avec lequel l'Arduino veut dialoguer. Il faudra autant de broches de sélection que de circuit branché sur le SPI [32].

D- Les broches d'alimentation

La carte Arduino Uno peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée. Une alimentation externe peut provenir soit d'un adaptateur AC-DC ou d'une batterie. L'adaptateur peut être connecté en branchant une prise 2.1mm dans la prise d'alimentation de la carte ou à partir d'une batterie connectée dans le pin (ou broche) GND et V-in (alimentation externe). Le processeur peut fonctionner sur une alimentation externe de 6 à 20 volts. Cependant, si la tension est inférieure à 7V, le pin 5V peut fournir moins de cinq volts et le processeur peut devenir instable.

Si la tension est supérieure à 12V, le régulateur de tension peut surchauffer et endommager la carte. La plage recommandée est de 7 à 12 volts.

Les pins (ou broches) d'alimentation sont les suivantes :

- V-in. Tension d'entrée à la carte Arduino à l'aide d'une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou une autre source d'alimentation régulée). Si l'alimentation en tension est faite par l'intermédiaire de la prise d'alimentation, on pourra y accéder via ce pin.
- 5V. Ce pin délivre un 5V régulé par la carte. Le processeur peut être alimenté soit à partir de la prise d'alimentation DC (7-12V), le connecteur USB (5V), ou le pin V-in de la carte (7-12). La fourniture d'une tension via les 5V ou 3,3V contourne le régulateur, et peut endommager votre processeur. A déconseiller !
- 3V 3. Une alimentation de 3,3 volts générée par le régulateur. La consommation de courant maximale est de 50 mA
- GND. masse
- IOREF. Ce pin sur la carte Arduino fournit la référence de tension avec laquelle le microcontrôleur fonctionne [34].

II.3.1.2. Partie programme**II.3.1.2.1. Environnement de programmation**

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C++). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information, ce programme appelé IDE Arduino [22].

II.3.1.2.2. Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C++ [22].



Figure II.6 : Interface IDE Arduino

II.3.1.2.3. Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino Uno) et le numéro de port USB (COM 3) comme à titre d'exemple cette figure suivante [22].

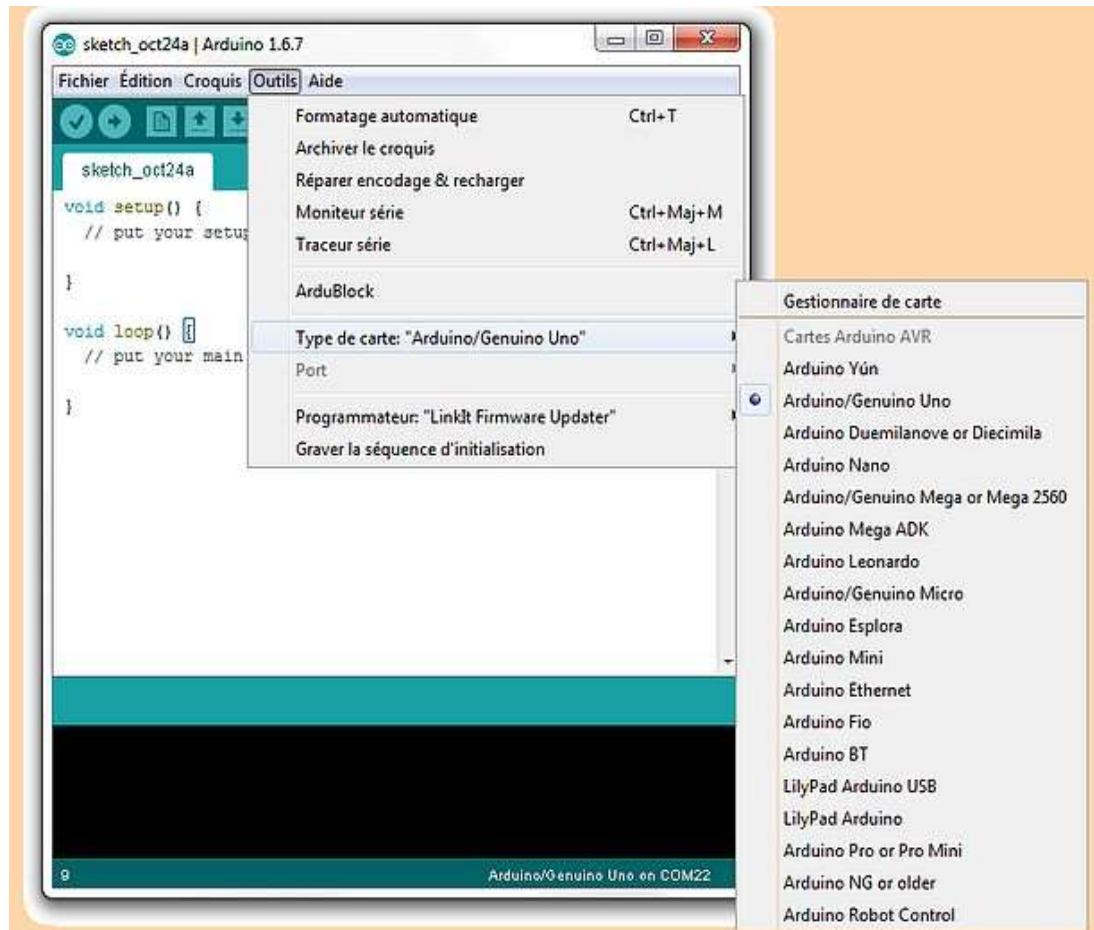


Figure II.7 : Paramétrage de la carte

II.3.1.2.4. Description du programme

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. Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme [22].

A- Les étapes de téléchargement du programme

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].

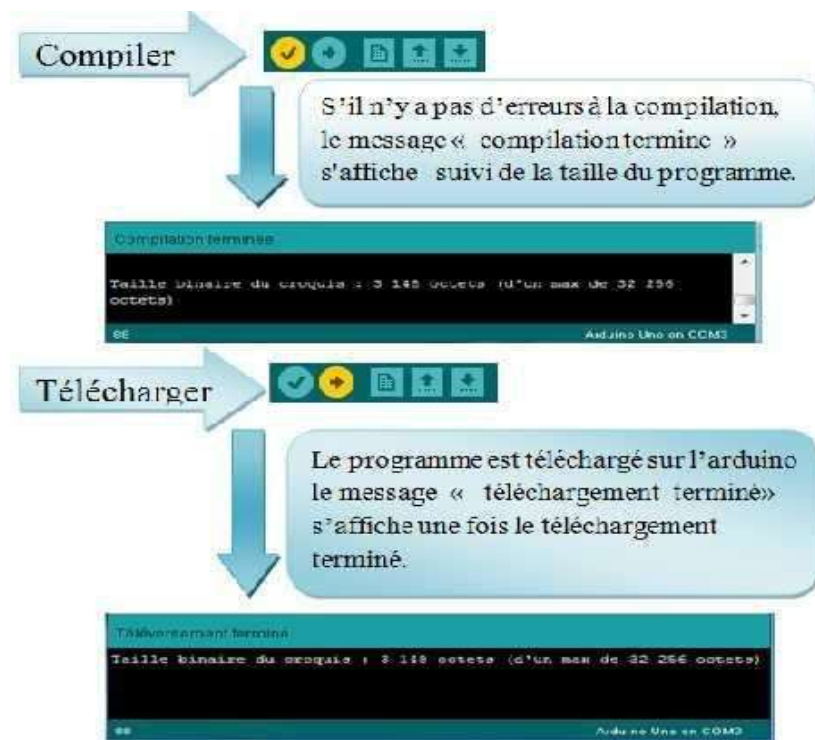


Figure II.8 : Les étapes de téléchargement du code

II.4. Le module Accéléromètre ADXL345-EP

II.4.1. Description Générale

L'ADXL345-EP est une version à performances étendues de l'ADXL345, qui est une petite machine à 3 axes, fine et de très faible puissance accéléromètre avec mesure à haute résolution (13 bits) jusqu'à ± 16 g. Les données de sortie numériques sont formatées en 16 bits, en complément de deux et est accessible soit par un SPI (3 ou 4 fils), soit par I2C numérique interface. L'ADXL345-EP est bien adapté à une plage de température étendue l'équipement industriel et aérospatial. Il mesure l'accélération de la gravité dans les applications de détection d'inclinaison, ainsi que l'accélération dynamique résultante d'un mouvement ou d'un choc. Sa haute résolution (3,9 mg/LSB) permet de mesurer le changement d'inclinaison (moins de $1,0^\circ$).

Plusieurs fonctions de détection spéciales sont prévues. L'activité et la détection de l'inactivité permet de détecter la présence ou l'absence de mouvement en comparant l'accélération sur un axe quelconque avec des seuils fixés par l'utilisateur.

La détection de tapotements détecte les tapotements simples et doubles dans n'importe quelle direction. La détection de chute libre détecte si l'appareil est en train de tomber. Ces fonctions peuvent être mis en correspondance individuellement avec l'une des deux broches de sortie d'interruption. Un système intégré de gestion de la mémoire avec un premier niveau de 32 entrées, la mémoire tampon FIFO (First Out) peut être utilisée pour stocker les données afin de minimiser les coûts de l'hôte. L'activité des processeurs et une consommation d'énergie globale du système plus faible.

Les modes de faible consommation permettent une alimentation intelligente basée sur le mouvement gestion avec détection de seuil et accélération active mesure à une dissipation de puissance extrêmement faible. L'ADXL345-EP est fourni dans un petit boîtier fin de 3 mm × 5 mm × 1 mm, 14 fils, emballage plastique amélioré [23].



Figure II.9 : Le module Accéléromètre ADXL345

II.4.2. Schéma fonctionnel

Le schéma fonctionnel de ADXL 345 est illustré dans la figure II.10.

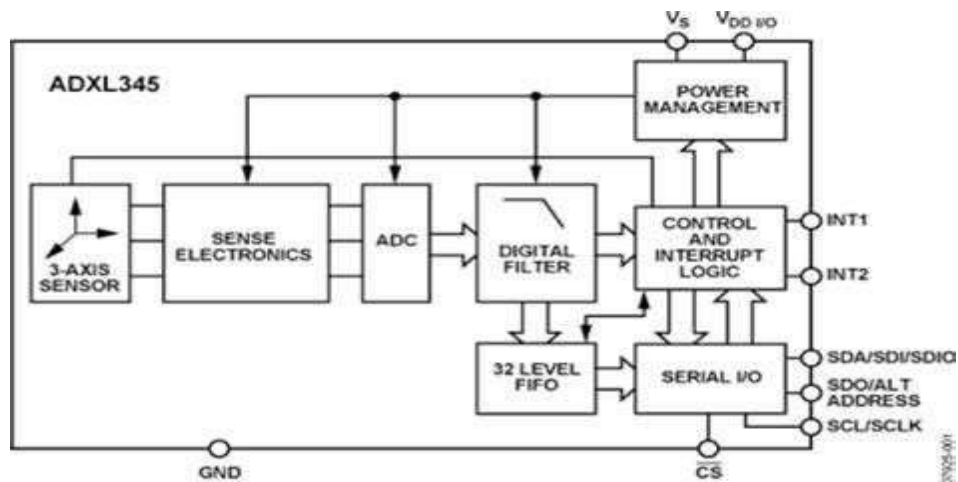


Figure II.10 : Schéma fonctionnel de ADXL 345[23]

II.4.3. Communication

L'ADXL345 est un accéléromètre qui communique soit en SPI soit en I2C. Ces deux bus sont régulièrement rencontrés dans des montages à base de microcontrôleurs et de capteurs, Ayant déjà utilisé le SPI auparavant, nous sommes naturellement orientés vers ce bus dans notre projet.

Dans ce cas (SPI), la communication entre le maître (ici, l'Arduino) et le ou les esclaves s'effectue au moyen de quatre liaisons minimums : l'horloge fourni par le maître pour la synchronisation, la réception de donnée, l'émission de donnée et enfin la sélection de l'esclave.

Pour pouvoir communiquer avec l'ADXL345, il faut mettre le pin de sélection de l'esclave associé à l'état haut. Une fois les messages reçus et/ou envoyés, il faut mettre ce pin à l'état bas.

Pour pouvoir configurer, il suffit d'écrire dans le registre voulu en envoyant d'abord le registre puis la valeur à y stocker. De la même façon, pour pouvoir récupérer une valeur, il faut écrire n'importe quelle valeur dans le registre que nous souhaitons lire et ensuite récupérer la valeur. Donc c'est au maître de demander toujours la donnée dont il a besoin.

Il est à noter que les valeurs des accélérations renvoyées par le capteur sont codées sur 10 bits mais envoyées sur deux octets (0x32 et 0x33 pour X, 0x34 et 0x35 pour Y et 0x36 et 0x37 pour Z) [24].

II.4.4. Configuration et le calibrage

L'ADXL345 peut-être paramétré sur une sensibilité de 2G, 4G, 8G ou bien encore 16G qui permet d'adapter la précision de la mesure en fonction du besoin. Dans notre cas, 4G fut le meilleur compromis après plusieurs tests. Pour cela, nous envoyons la valeur 0x01 dans le registre 0x31.

Avant toute utilisation des valeurs reçues, il faut d'abord effectuer un calibrage afin de déterminer la position initiale du capteur. En effet, les calculs que nous verrons par la suite ne sont valables que dans le cas où les trois axes valent 0 à l'état initial. Hors, le capteur ne sera pas forcément bien positionné sur la main et les valeurs ne seront alors pas exactes. On se retrouvera alors avec une erreur dans la position calculée. Il faut donc prendre en compte cette différence.

Pour pouvoir réaliser cette calibration, nous considérons qu'au démarrage de l'Arduino, la tête de l'utilisateur se trouve en position initiale. Il suffit alors de récupérer plusieurs mesures et d'en faire la moyenne après avoir lancé les mesures en envoyant 0x08 dans le registre 0x2D. La moyenne obtenue sera soustraite aux mesures effectuées par la suite [24].

II.4.5. Traitement des données

Pour pouvoir récupérer les angles qui nous sont utiles, il faut appliquer une règle de math : dans un plan 0xy, l'angle d'un vecteur AB par rapport à l'axe 0x vaut :

$$\theta_{AB} = \text{Arctan}((y_B - y_A) / (x_B - x_A)) \quad \text{Équation II.1}$$

Sachant que dans le cas où $x_B - x_A$ vaut 0, il faut appliquer une autre formule.

La bibliothèque "math" propose la fonction Arctan^2 qui prend en compte le cas du $x_B - x_A =$

0. Appliquer à notre programme, on a donc :

$$\text{- Pour l'inclinaison avant-arrière : } \theta_{ZX} = \text{Arctan}^2(z_{Raw}, x_{Raw}) \quad \text{Équation II.2}$$

$$\text{- Pour l'inclinaison gauche-droite : } \theta_{ZY} = \text{Arctan}^2(z_{Raw}, y_{Raw}) \quad \text{Équation II.3}$$

Où x_{Raw} , y_{Raw} et z_{Raw} sont les trois valeurs renvoyées par le capteur [24].

II.5. Modules Bluetooth

Il existe deux sortes de module Bluetooth, tous deux compatibles Arduino et utilisables sur un breadboard (plaque d'essai en français). On les distingue par le nombre de pattes d'entrées / sorties :

- HC-06 : 4 sorties. Ce module ne peut être qu'esclave.
- HC-05 : 6 sorties. Ce module peut être “**maître**” (il peut proposer à un autre élément Bluetooth de s'appairer avec lui) ou “**esclave**” (il ne peut que recevoir des demandes d'appairage). **C'est ce module que nous utilisons dans notre projet.**

Les deux modules peuvent être utilisés en mode COMMANDE, pour les programmer avec des “commandes AT ”, ou en mode DATA, pour échanger des données [25].

II.5.1. Module Bluetooth HC-05

Le module Bluetooth HC-05 est un module Bluetooth SPP (protocole de port série) facile à utiliser, conçu pour la configuration d'une connexion série sans fil transparente. Sa communication s'effectue via une communication série, ce qui facilite l'interface avec le contrôleur ou le PC. Le module Bluetooth HC-05 fournit un mode de commutation entre les modes maître et esclave, ce qui signifie qu'il ne peut utiliser ni la réception ni la transmission de données [26].

II.5.1.1. Caractéristique de module [27]

Ce module Bluetooth HC05 permet d'ajouter une liaison Bluetooth sur vos projets à microcontrôleur (communication via série TTL). Livré avec un cordon 6 broches de raccordement.

- Alimentation : 3,6 à 6 Vcc (régulateur 3 Vcc intégré)
- Bluetooth : 2.0
- Portée : jusqu'à 10 mètres
- Liaison série : 4800 à 1382400 bauds
- Antenne : 2,4 GHz intégrée
- Sorties : Key, Vcc, Gnd, TXD, RXD et State.
- Configuration via commandes AT
- Longueur du cordon : environ 20 cm
- Dimensions : 35 x 16 x 12 mm

II.5.1.2. Principe de module

Le module HC05 est un module simple d'emploi qui permet d'ajouter la fonctionnalité communication Bluetooth full duplex, c'est-à-dire bidirectionnelle, à vos projets Arduino.

Ci-dessous, une photo du module et un schéma qui fait figurer ses broches qui permettent de le connecter comme périphérique de la carte Arduino :

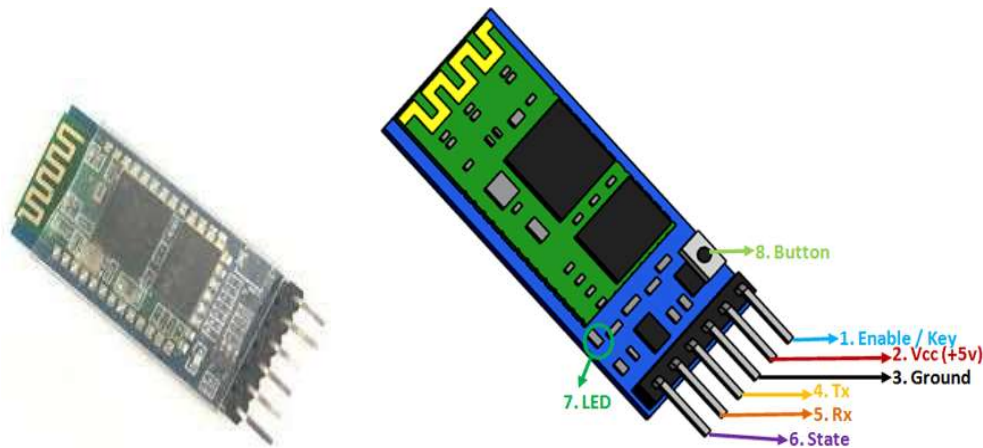


Figure II.11 : Photo et schéma qui figurent les différentes broches de module Bluetooth.

Nous allons expliquer chaque broche :

- **Enable/Key :** permet de fixer le mode Data, au niveau bas, ou le mode AT command (cf. plus bas), au niveau haut. Par défaut, si la broche n'est pas connectée, le module est en mode Data. L'état de cette broche est aussi contrôlé par le bouton (cf. plus bas) ;
- **Vcc :** c'est la tension d'alimentation, à priori 5V ;
- **Ground,** la masse ;
- **Tx (ou TXD) :** tout ce qui est reçu par le Bluetooth du module est sorti sur cette broche. C'est donc Transmit Data, vu du côté du module HC05 ;
- **Rx (ou RXD) :** tout ce qui est présenté sur cette broche (issu du microcontrôleur) est envoyé par le Bluetooth du module. C'est donc le Récepteur Data du point de vue du module HC05 ;
- **State :** la broche est connectée à la LED embarquée du module et indique l'état de fonctionnement du module avec les caractéristiques décrites ci-dessous [28].

II.6. Pilote de moteurs L293 D

Le L293D est un circuit intégré typique de pilote de moteur ou de pilote de moteur qui permet au moteur à courant continu de conduire dans les deux sens. Le L293D est un circuit intégré à 16 broches qui peut contrôler simultanément un ensemble de deux moteurs à courant continu dans n'importe quelle direction. Cela signifie que vous pouvez contrôler deux moteurs à courant continu avec un seul circuit intégré L293D. Dans une seule puce L293D, il y a deux circuits de pont en H à l'intérieur du circuit intégré qui peuvent faire tourner deux moteurs à courant continu indépendamment. Le pont en H est un circuit qui permet à la tension d'être Les

circuits intégrés à pont en H sont parfaits pour entraîner un moteur à courant continu. En raison de sa taille, il est très utilisé dans les applications robotiques pour contrôler les moteurs à courant continu [13]. La fiche technique de L293D se trouve dans l'annexe (A).

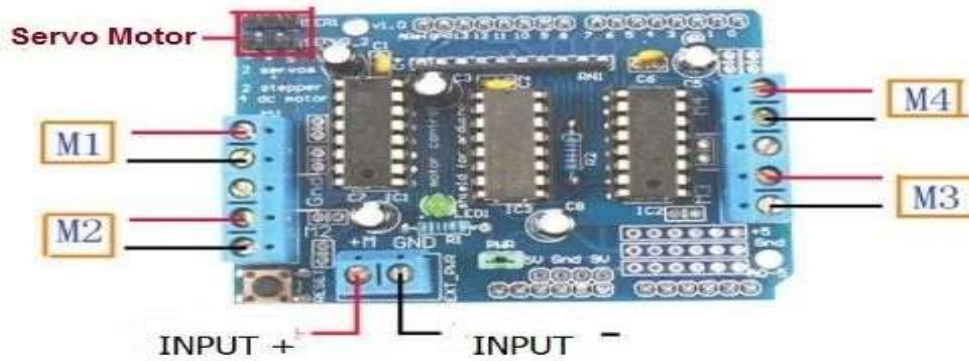


Figure II.12 : Pilote de moteurs L293D

II.6.1. Fonctionnement de L293 D :

Il y a 4 broches d'entrée pour l293d, les broches 2,7 à gauche et les broches 15, 10 à droite, comme indiqué sur le diagramme des broches. Les broches d'entrée de gauche réguleront la rotation du moteur connecté sur le côté gauche et l'entrée droite pour le moteur du côté droit. Les moteurs tournent sur la base des entrées fournies à travers les broches d'entrée comme LOGIC 0 ou LOGIC 1. Pour faire tourner le moteur dans le sens des aiguilles d'une montre, les broches d'entrée doivent être pourvues de Logic 1 et Logic 0. Activez les broches 1 et 9 (correspondant aux deux moteurs) doit être élevée pour que les moteurs commencent à fonctionner. Lorsqu'une entrée d'activation est élevée, le pilote associé est activé. En conséquence, les sorties deviennent actives et fonctionnent en phase avec leurs entrées. De même, lorsque l'entrée d'activation est basse, ce pilote est désactivé et leurs sorties sont désactivées et à l'état haute impédance [29].

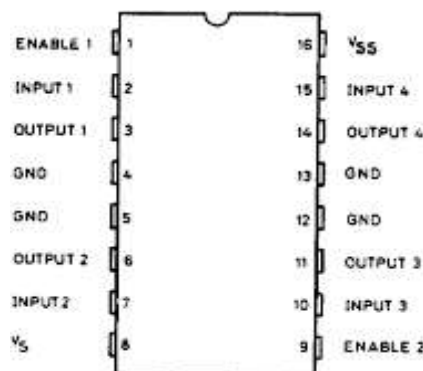


Figure II.13 : Différents broches du module L293 D

II.6.2. Description des broches :

- VSS - Alimentation de la logique de commande (5V). A raccorder à la borne +5V d'Arduino (donc sur le régulateur d'Arduino).
- VS - Alimentation de puissance des moteurs. Par exemple, s'il s'agit d'un ancien véhicule téléguidé que vous avez cannibalisé, il s'agira de la borne positive (+9.2v) de votre accumulateur.
- GND - Doit être raccorder à la masse (GND) de la source d'alimentation de puissance VS (donc le négatif de l'accumulateur) et à la masse de la source d'alimentation de VSS (donc GND Arduino).
- Si vous n'avez qu'une source d'alimentation pour le tout, c'est forcément plus simple.
- OUTPUT1, OUTPUT2 - Broches à raccorder à la charge (le moteur). C'est via ces broches de INPUT1, INPUT2 - Broche de commande du Pont-H. Se raccorde à Arduino.
- ENABLE1 - permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 & OUTPUT2.
- ENABLE1 commande l'activation du premier Pont-H.
 - Si ENABLE1 = GND, le pont-H est déconnecté et le moteur ne fonctionne pas.
 - Si ENABLE1 = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT1 & INPUT2 [30].

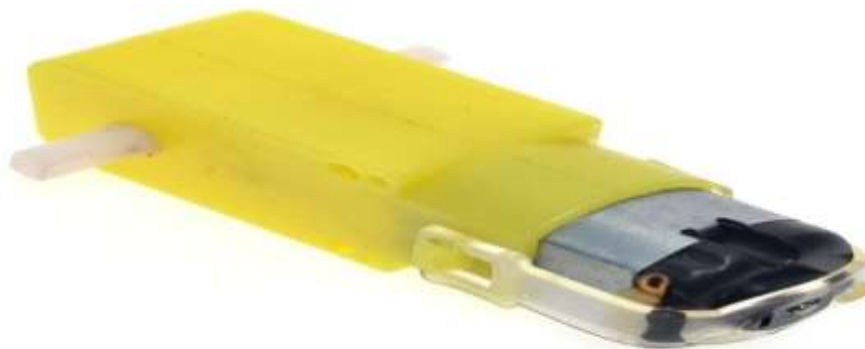
II.7. Moteurs à courant continu :

Figure II.14 : Moteur à courant continu (DC 12V)

Les moteurs à courant continu à excitation séparée sont encore quelquefois utilisés pour l'entraînement à vitesse variable des machines. Très faciles à miniaturiser, ils s'imposent dans les très faibles puissances et les faibles tensions. Ils se prêtent également fort bien, jusqu'à des puissances importantes (plusieurs mégawatts), à la variation de vitesse avec des technologies électroniques simples et peu onéreuses pour des performances élevées (plage de variation couramment exploitée de 1 à 100).

Leurs caractéristiques permettent également une régulation précise du couple, en moteur ou en générateur. Leur vitesse de rotation nominale, indépendante de la fréquence du réseau, est aisément adaptable par construction à toutes les applications. Ils sont en revanche moins robustes que les moteurs asynchrones et beaucoup plus chers, tant en coût matériel qu'en maintenance, car ils nécessitent un entretien régulier du collecteur et des balais.

II.7.1. Constitution :

Un moteur à courant continu est composé des éléments suivants :

a. L'inducteur ou stator

C'est un élément du circuit magnétique immobile sur lequel un enroulement est bobiné afin de produire un champ magnétique. L'électro-aimant ainsi réalisé comporte une cavité cylindrique entre ses pôles.

b. L'induit ou rotor

C'est un cylindre en tôles magnétiques isolées entre elles et perpendiculaires à l'axe du cylindre. L'induit est mobile en rotation autour de son axe et est séparé de l'inducteur par un entrefer. A sa périphérie, des conducteurs sont régulièrement répartis.

c. Le collecteur et les balais

Le collecteur est solidaire de l'induit. Les balais sont fixes, ils frottent sur le collecteur et ainsi alimentent les conducteurs de l'induit.

II.7.2. Principe de fonctionnement

Le principe de fonctionnement d'un moteur à courant continu repose sur la création d'un couple électromagnétique sur un rotor porteur de conducteurs parcourus par un courant continu, et qui se déplacent dans un champ magnétique.

Un inducteur (stator) crée un champ magnétique de direction fixe. Ce champ peut être obtenu par un aimant permanent ou par un électro-aimant.

L'induit (rotor) porte des conducteurs parcourus par un courant continu. Les spires sont le siège de forces qui créent un couple entraînant la rotation du rotor. De cette rotation résulte une variation du flux traversant la spire, variation qui engendre une force électromotrice (f.e.m.) aux bornes de la spire.

Un dispositif permet l'alimentation des conducteurs en inversant leur sens deux fois par tour. Ce dispositif est appelé collecteur. Des solutions plus récentes permettent par une solution électronique de s'affranchir du collecteur. Les moteurs sont alors appelés moteur à courant continu [31].

II.8. Conclusion

Dans ce chapitre nous avons projeté la lumière sur une carte d'acquisition qui est l'Arduino donnant ainsi les raisons pour lesquelles en l'as choisis puis nous avons cité ces différents types et expliqué le principe de fonctionnement de la carte sans oublier ces caractéristiques. Ensuite nous avons également présenté le module adxl345 et son principe de fonctionnement, ainsi que le module de communication sans fil (Bluetooth).

Le chapitre suivant sera consacré à la réalisation d'un dispositif de commande domotique comme une première phase et une deuxième phase de commande à distance de véhicule géré par une commande gestuelle.

Chapitre III

Réalisation d'un prototype

expérimental du

Robot gestuel

III.1. Introduction

Après avoir donné dans le chapitre précédent une description théorique sur les composants utilisés, plus précisément le module Arduino et son environnement de développement, on va procéder à l'application expérimentale. Pour cette raison, plusieurs blocs ont été nécessaires afin de réaliser une telle combinaison.

Dans ce chapitre on traite la manière dont on contrôle un robot en reconnaissant différents gestes du corps d'une personne, généralement des mouvements de mains. Cela aidera une personne à contrôler facilement la voiture automatisée pour se déplacer dans une direction différente selon les besoins de l'utilisateur.

III.2. Principe de fonctionnement d'un Robot à commande gestuelle

Comme son nom l'indique, il s'agit d'un robot contrôlé par des gestes de la main de la personne. Pour les mouvements simples de la voiture, si la main d'une personne est inclinée vers le bas, la voiture se déplace vers l'avant, si elle est inclinée au-dessus puis la voiture se déplace vers l'arrière, si elle est inclinée vers la gauche, la voiture tourne dans la direction de gauche et si elle est inclinée vers la droite, la voiture tourne dans la bonne direction. Tout d'abord, un accéléromètre est utilisé (ADXL345). Ce capteur a 3 axes X, Y, Z qui détecte la position de la main d'une personne et donne une sortie dans une plage de séries. Ces données sont reçues par Arduino Uno (de la partie émettrice).

Le microcontrôleur d'Arduino (ATMEGA328) calcule et transmet les données sur la direction vers laquelle la voiture doit se déplacer. Ces données sont ensuite envoyées à un autre Microcontrôleur d'Arduino Uno installé sur le robot à l'aide de deux modules de la communication sans fils Bluetooth (HC-05) émetteur et récepteur à une bande de fréquence de 2.4 GHz, ces données sont envoyées dans les airs au format codé.

Maintenant, les données capturées par le module récepteur seront envoyées à une carte de pilotage de moteur (L293D) qui fait tourner les moteurs dans les quatre directions souhaitées par les gestes de la main de l'utilisateur, après leur traitement dans le microcontrôleur de l'Arduino placé sur le robot pour sélectionner les broches à régler haut pour que les moteurs de la voiture automatisée puissent fonctionner en conséquence. Tout ce processus est donné dans le schéma de la figure III.1.

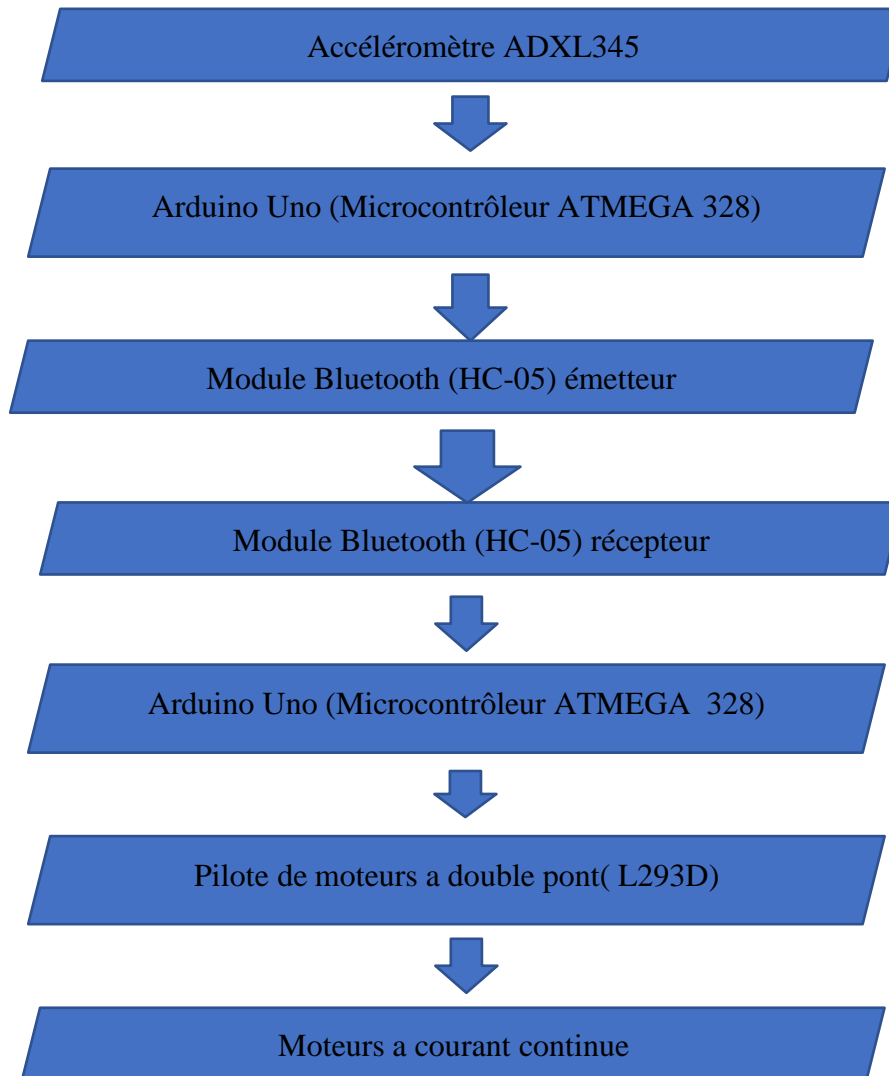


Figure III.1 : chaîne de commande de robot

III.3. Choix technique des composants utilisés :

L'ensemble des composants sont déjà cités en détails dans le chapitre (II). Notre choix a été fait selon plusieurs critères comme : la tension et le courant requis pour le fonctionnement de notre robot, la compatibilité, leurs capacités et le coût bien sûr comme c'est mentionné dans le tableau suivant :

Chapitre III : *Réalisation d'un prototype expérimental du Robot gestuel*

Tableau III.1 : liste des composants utilisés

Le composant	La description	Le nombre	Coût
Accéléromètre (ADXL345)	L'accéléromètre numérique à trois axes ADXL345 est un dispositif analogique possède des versions pour les interfaces I2C et SPI. Un régulateur 3,3 V et un commutateur permettant de modifier le niveau logique permettent d'intégrer la platine à un microcontrôleur 3 V ou 5 V comme l'Arduino. La sensibilité des trois axes de mesure X, Y et Z est réglable sur ± 2 g, ± 4 g, ± 8 g ou ± 16 g. Le régulateur peut être alimenté avec une tension de 5 V à l'entrée, elle sera de 3,3 V à la sortie. La carte est entièrement montée et testée et possède deux trous pour une fixation facile, ainsi qu'une barrette à broches à 9 pôles pouvant être utilisée avec une platine d'expérimentation.	1	5.07€
Arduino Uno	Alimentation : 12 V (3 piles en série) Mémoires :(flash : 32 KB ; SRAM : 2 KB ; EEPROM : 1 KB). Interfaces:(14 broches d'E/S dont 6 PWM ; 6 entrées analogiques 10 ; Bus série, I2C et SPI). Intensité par E/S : 40 mA Cadencement : 16 MH Dimensions : 74 x 53 x 15 mm	2	9.86€ x 2

Chapitre III : *Réalisation d'un prototype expérimental du Robot gestuel*

Module Bluetooth (HC-05)	<p>-Tension d'Alimentation : 3.3 à 5V DC (courant continu a 30mA en fonctionnement, 8 mA pour établir la communication)</p> <p>-Paramétrage par défaut du port série : 9600, N, 8, 1.</p> <p>-Bluetooth sur la bande 2.4 GHz, modulation GFSK</p> <p>-LED indicatrice de statut de connexion Bluetooth sur la bande 2.4 GHz.</p> <p>-Le module est appairé avec un mot de passe (1234) modifiable</p>	2	6.16€ x 2
Contrôleur des moteurs à double pont (L293D)	<p>Large gamme de tension d'alimentation : 4,5 V à 36 V</p> <p>Les dispositifs L293 et L293D sont quadruples les conducteurs actuels en demi-H. Le L293 est conçu pour fournir des courants de commande bidirectionnels jusqu'à 1 A à des tensions de 4,5 V à 36 V. Le L293D est conçu pour fournir des courants de commande bidirectionnels jusqu'à 600 mA à des tensions de 4,5 V à 36 V. Les deux appareils sont conçus pour piloter des charges inductives telles que des relais, des solénoïdes, des moteurs pas à pas à courant continu et bipolaires, ainsi que des charges à courant et à tension élevés dans des applications d'alimentation positive</p>	1	5.39€
Moteur a courant continue DC	<p>1. Tension : dc 3v-6v</p> <p>2. Courant : 100 ma-120ma</p> <p>3. vitesse de la voiture (m / minute) : 20-48</p> <p>4. Dimension : 55mm x 48.3mm x 23mm</p> <p>5. Poids : 45g</p>	4	3.54€ x 4
Batteries	(9v) et (12v) tension continue	3	2€
Les files de connexions	File en cuivre (Transport d'électricité)	30	0.43€

III.4. Etapes de la réalisation

Notre réalisation pratique a été faite en deux parties :

- La première partie est la conception des systèmes électroniques du robot (partie réceptrice) et de sa commande (partie émettrice).

La première partie de notre projet est très importante, ou on est passé par plusieurs étapes :

1. Chercher les différentes structures des blocs constituant notre dispositif et qui vont avec les objectifs fixés et les moyens disponibles.
 2. Présenter les différents éléments ou composants constituant chacun des blocs en choisissant des composants aux caractéristiques voulus, à défaut, on choisira ceux disponibles sur le marché.
 3. L'assemblage des composants suivant notre schéma présenté sur les figures (III.5 et III.12).
- La deuxième partie est la programmation du robot et de son système de commande.

Dans la deuxième partie, on est passé par les deux étapes suivantes :

1. L'installation de l'interface IDE sur l'ordinateur (la procédure est expliquée dans le chapitre (II)).
2. Création et téléversement du code dans le robot et son système de commande.

III.5. Schéma synoptique général

Le schéma synoptique général de notre dispositif est indiqué par la figure (III.2).

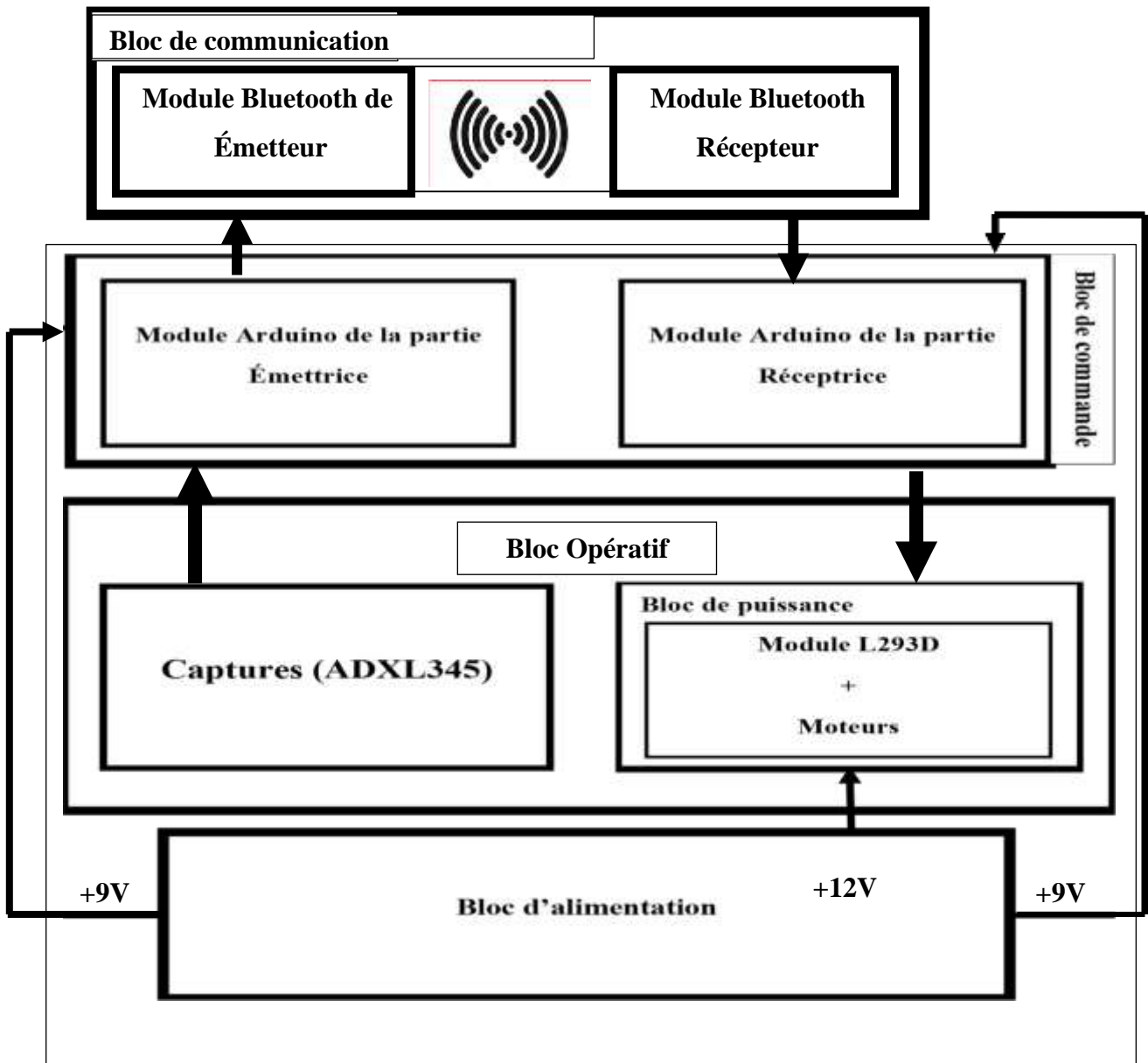


Figure III.2 : Schéma synoptique générale de projet

Pour faciliter cette étude, on a décomposé le schéma synoptique en quatre blocs :

- Bloc d'alimentation ;
- Bloc de commande
- Bloc de communication ;
- Bloc opératif.

En ce qui concerne les éléments principaux de ce dispositif, notre choix était la carte Arduino UNO et le module ADXL345.

Le montage général de notre réalisation est illustré dans l'annexe (B).

III.5.1. Bloc d'alimentation

III.5.1.1. Partie émettrice :

Généralement l'ensemble des composants utilisés dans cette partie exige une tension stable de (+5V).

Pour l'alimentation de l'Arduino nous l'avons relié à une pile de (+9V), ensuite le régulateur de tension intégré dans la carte nous donne une tension de (+5V) pour l'alimentation de : la carte Arduino, l'accéléromètre (ADXL345) et le module Bluetooth.

III.5.1.1.1. Schéma d'alimentation de la partie :

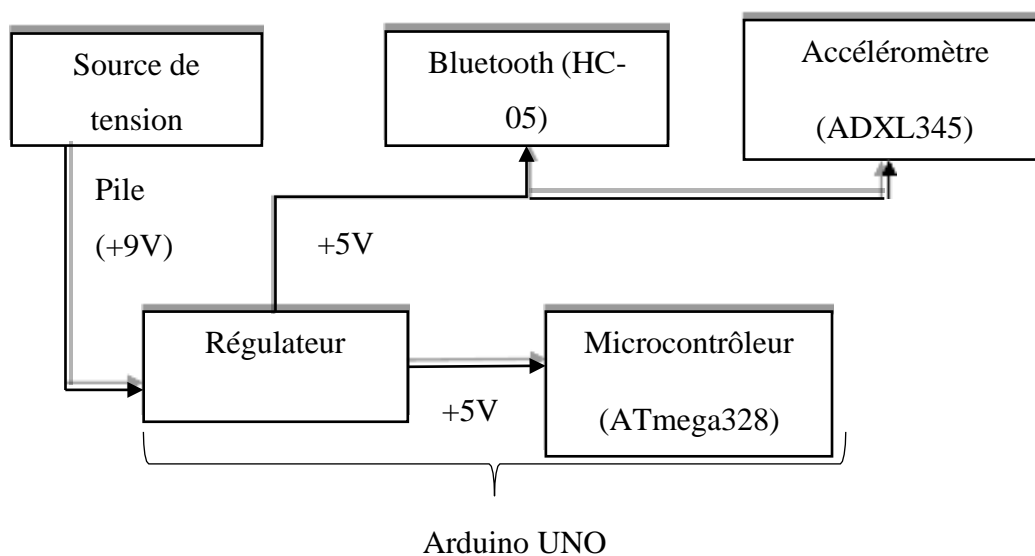


Figure III.3 : schéma électrique de la partie émettrice

III.5.1.2. Partie réceptrice :

Pour l'alimentation de cette partie, deux sources de tension continue (DC) sont utilisées : une batterie de (+12V) et une pile de (+9V).

Les deux composants : Arduino et le module Bluetooth sont alimentés via une tension de (+5V) qui vient de la pile (+9V) grâce à un régulateur intégré dans la carte.

Aussi, nous avons fourni une alimentation continue de +12V, pour alimenter le module L293 D, qui contrôle à son tour le mouvement des moteurs.

III.5.1.2.1. Schéma d'alimentation de la partie :

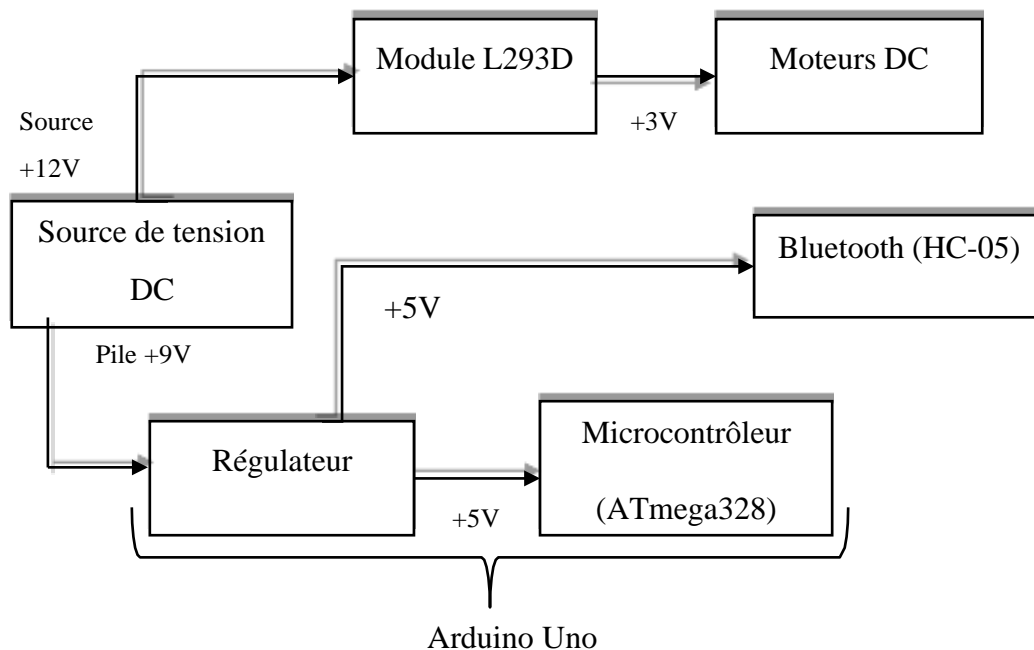


Figure III.4 : Schéma électrique de circuit récepteur

III.5.2. Bloc de commande :

III.5.2.1. Partie émettrice :

Notre bloc de commande, on le résume tout simplement par l'utilisation du module Arduino Uno (détaillé précédemment au chapitre II), qui reçoit des données sur la position de la main captée par le module ADXL345 puis elles les traitent à l'aide du microcontrôleur ATmega 328 pour en fin aboutir un résultat sous forme d'un signal numérique qui sera transmis au microcontrôleur de la partie réceptrice (robot) via une communication Bluetooth.

III.5.2.2. Partie réceptrice :

Un autre module Arduino Uno est utilisé pour traiter les données reçues de la partie émettrice via un module Bluetooth afin de les transmettre la commande au dispositif L293D sous forme d'un signal numérique.

III.5.3. Bloc de communication :

Dans ce bloc on a établi la configuration d'une connexion série entre les deux modules Bluetooth (maitre/esclave) qui est dans notre cas le module (HC-05), pour faire la transmission des données entre les deux cartes Arduino de la partie émettrice et la partie réceptrice.

III.5.4. Bloc opératif :

Constitue de deux parties :

III.5.4.1. Capteurs :

Dans notre cas le capteur utilisé est l'accéléromètre ADXL345, qui détecte et reconnaît la position de la main, par rapport au deux axes X, Y et Z, ces données seront fournies à l'Arduino de la partie maitre. Le capteur fonctionne sur une puissance comprise entre 1,8V et 3,6VDC (3,3V optimal), et ne consomme généralement que 350 μ A de courant qui devient parfaitement compatible pour Arduino. Il dispose de 3 sorties analogiques pour les axes X, Y et Z de mesures, 2 broches d'alimentation et une broche d'auto-test. Les sorties analogiques sont radiométriques, c'est-à-dire 0g est généralement égale à la moitié de la Tension d'alimentation de 5V, c'est-à-dire 2,5V, -3g est à 0v et 3g est à 3,3V avec une mise à l'échelle complète entre les deux. (Pour plus de détails voire le chapitre II).

III.5.4.2. Bloc de puissance :

Notre bloc de puissance se compose de deux sous blocs :

- a- Les actionneurs : qui sont les moteurs à courant continue.
- b- Le pilote des moteurs L293D : c'est un double pont- H qui commande la rotation des moteurs et leur arrêt à travers les impulsions PWM reçu de la carte Arduino.

III.6. Mise en œuvre :

Le processus de travail est divisé en deux parties principales, Circuit émetteur et circuit récepteur.

III.6.1. Partie émettrice :

III.6.1.1. Présentation Fritzing de circuit émetteur :

Le schéma fonctionnel de la partie est représenté dans la figure III.5.

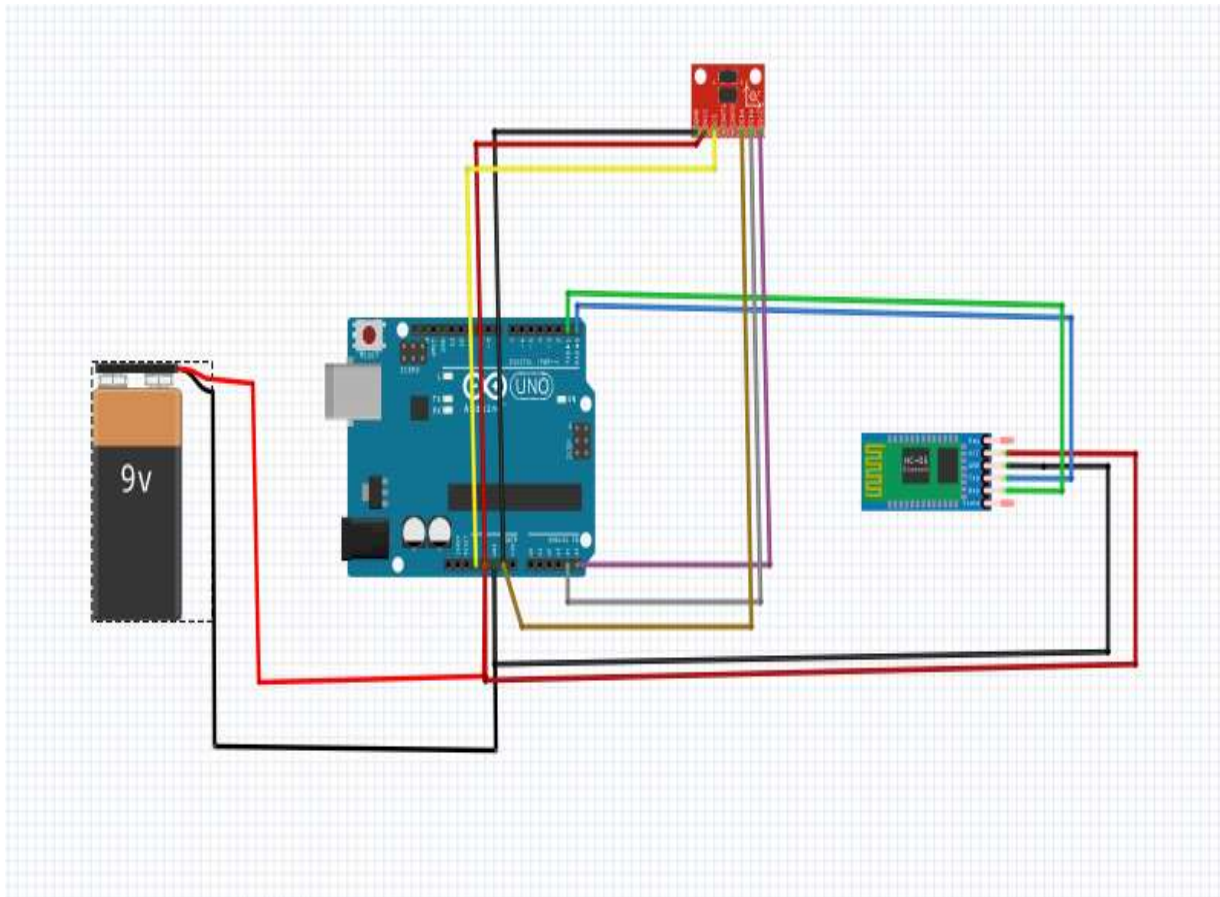


Figure III.5 : présentation Fritzing de la partie émettrice

III.6.1.2. Organigramme

Dans le circuit de transmission, la première étape consiste à reconnaître et à détecter les mouvements de la main. Pour ce faire, on utilise l'accéléromètre ADXL345. La position de la main est identifiée en utilisant les échelles X et Y. Ces données sont transmises au microcontrôleur ATMEGA 328 de l'Arduino qui peut être programmé à l'aide du logiciel Arduino IDE. Le code du programme de la partie émetteur est illustré dans l'annexe (C).

L'algorithme suivant est utilisé pour détecter et analyser les gestes et la direction dans laquelle le robot doit se déplacer.

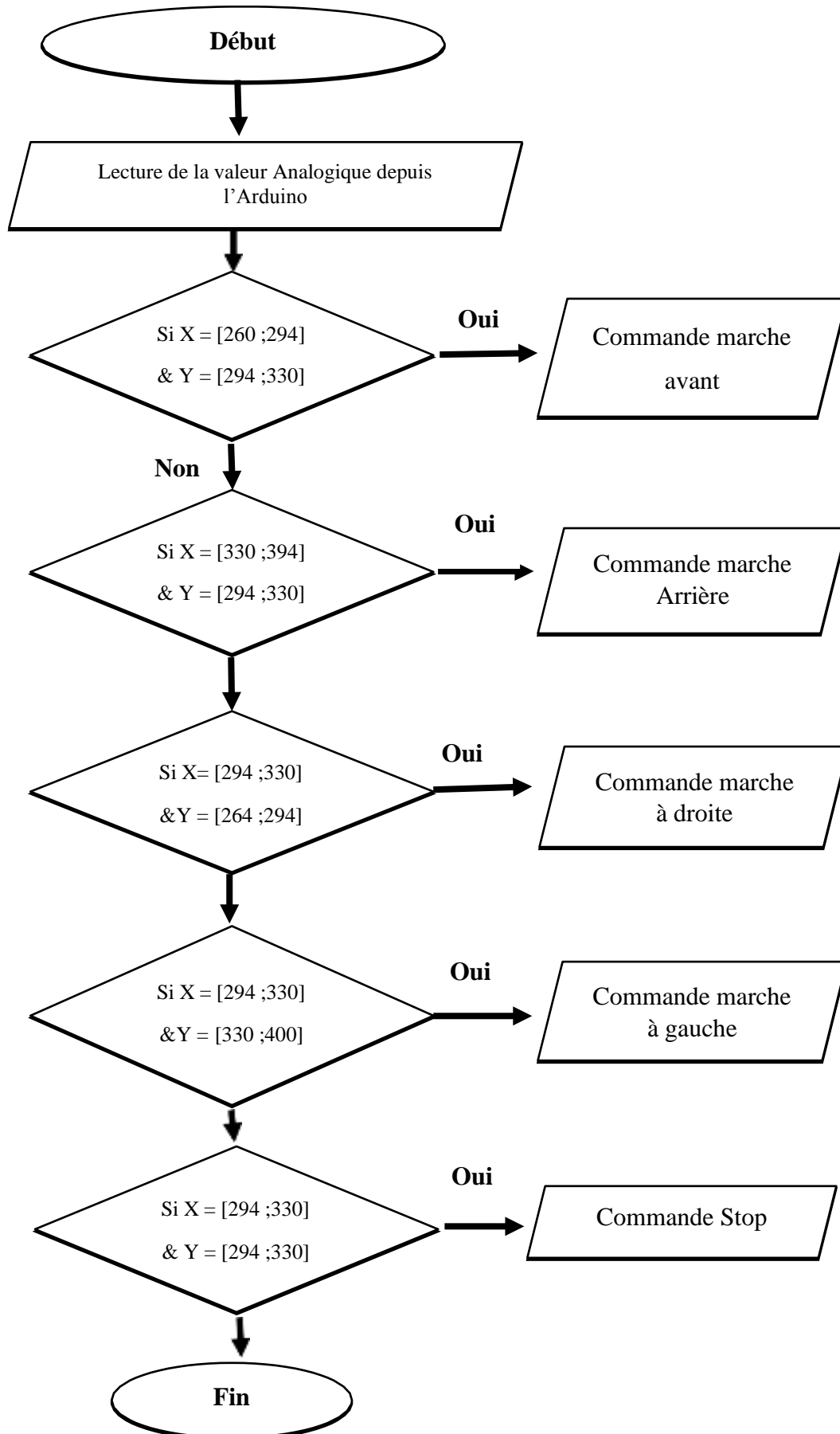


Figure III.6 : Organigramme de la commande émettrice

III.6.1.3. Explication des commandes :

a- Condition d'arrêt :

Si les valeurs de l'axe des X et de l'axe des Y sont comprises entre 294 et 330, la voiture arrête son mouvement.

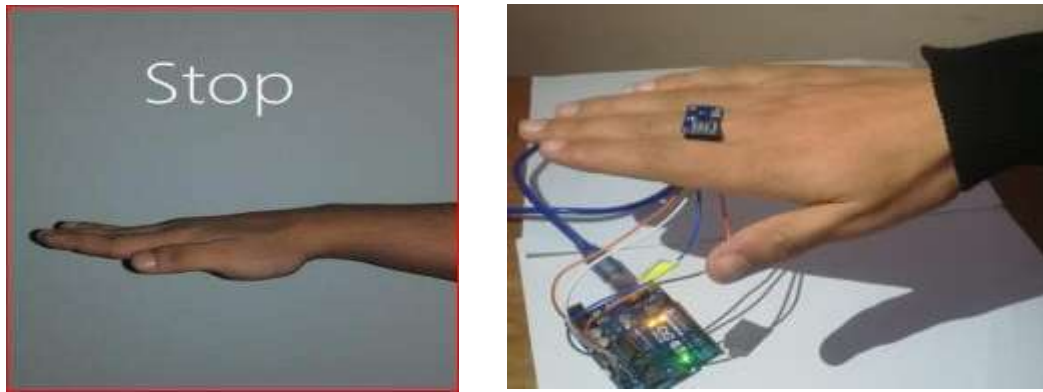


Figure III.7 : position de la main pour la commande stop

b- Pour que la voiture puisse avancer :

Si la valeur X se situe entre 260 et 294 et la valeur Y entre 294 et 330, la voiture avance.

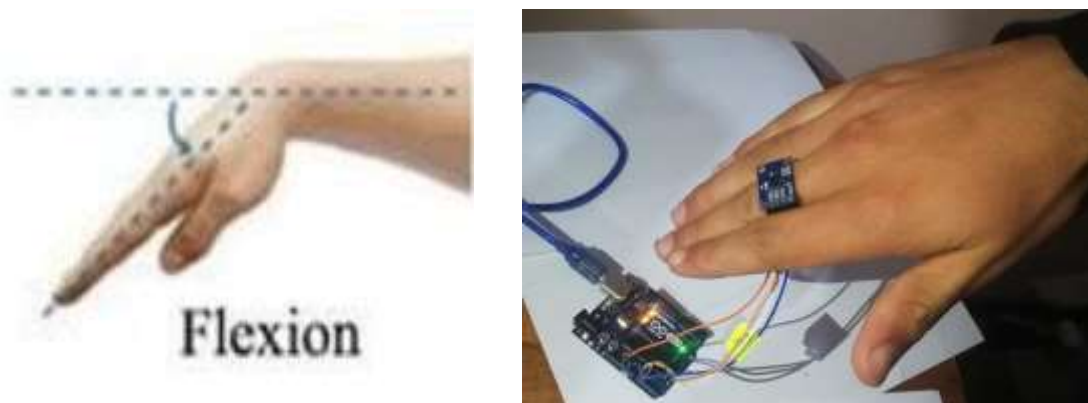


Figure III.8 : position de la main pour la commande marche en avant

c- Pour que la voiture se déplace en arrière :

Si la valeur de l'axe des X se situe entre 330 et 394 et la valeur de l'axe des Y entre 294 et 330, la voiture se déplace vers l'arrière

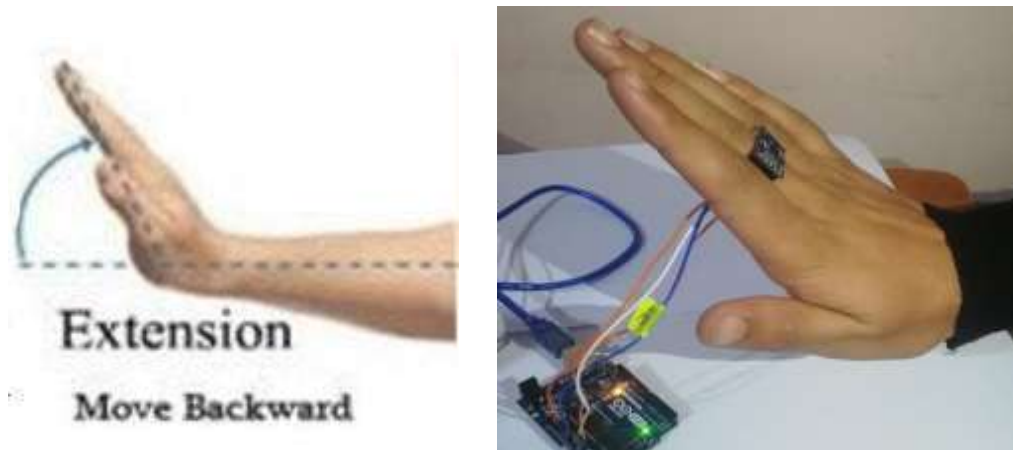


Figure III.9 : position de la main pour la commande marche arrière

d- Pour la voiture, tourner à gauche :

Si la valeur de l'axe des X se situe entre 294 et 330 et la valeur de l'axe des Y entre 330 et 400, la voiture tournera à gauche.

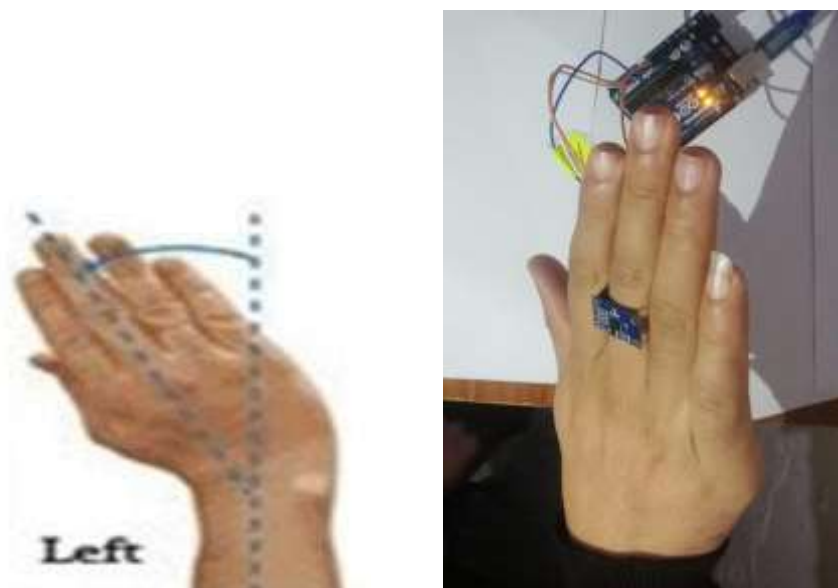


Figure III.10 : position de la main pour la commande tourne à gauche

e- Pour la voiture, tournez à droite :

Si la valeur l'axe des X se situe entre 294 et 330 et la valeur de l'axe des Y entre 264 et 294, la voiture tournera à droite.

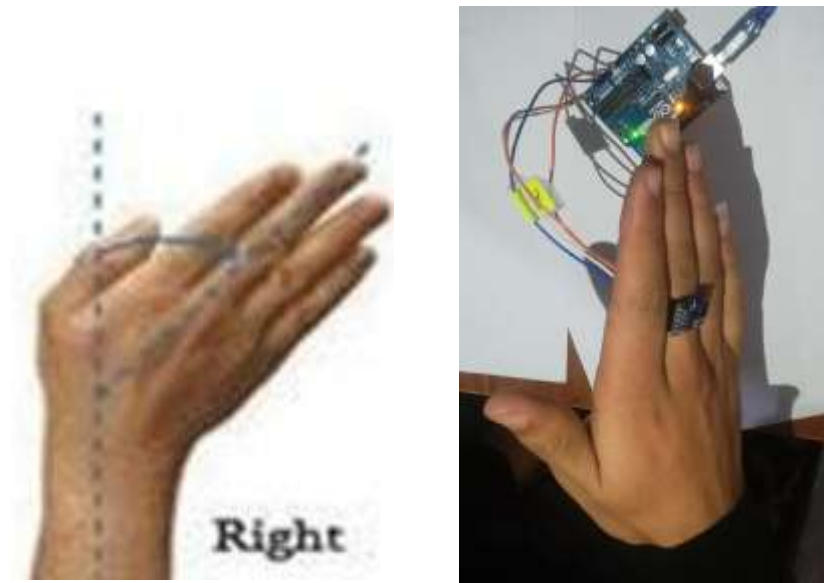


Figure III.11 : position de la main pour la commande tourne à droite

III.6.2. Partie réceptrice :

III.6.2.1. Présentation Fritzing de circuit récepteur :

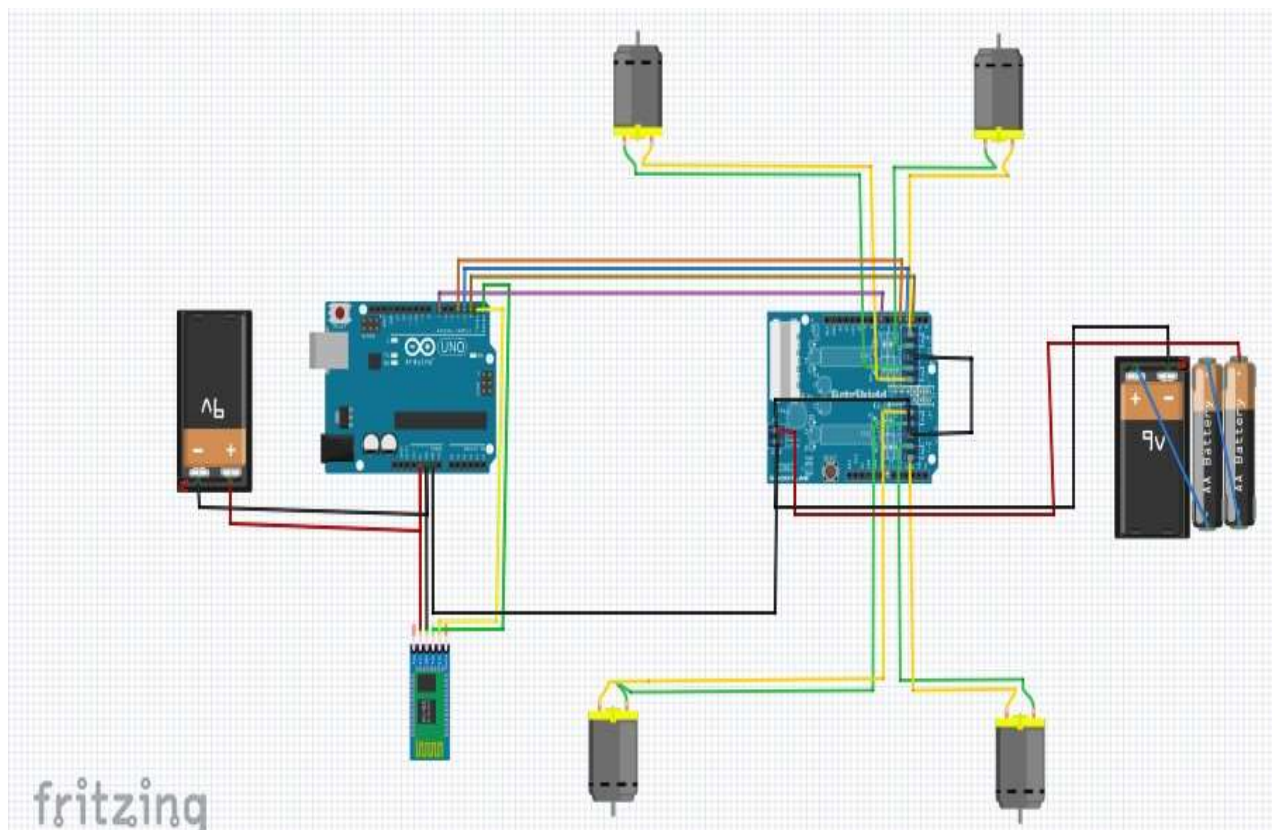


Figure III.12 : Schéma de circuit récepteur sur Fritzing

III.6.2.2. Organigramme :

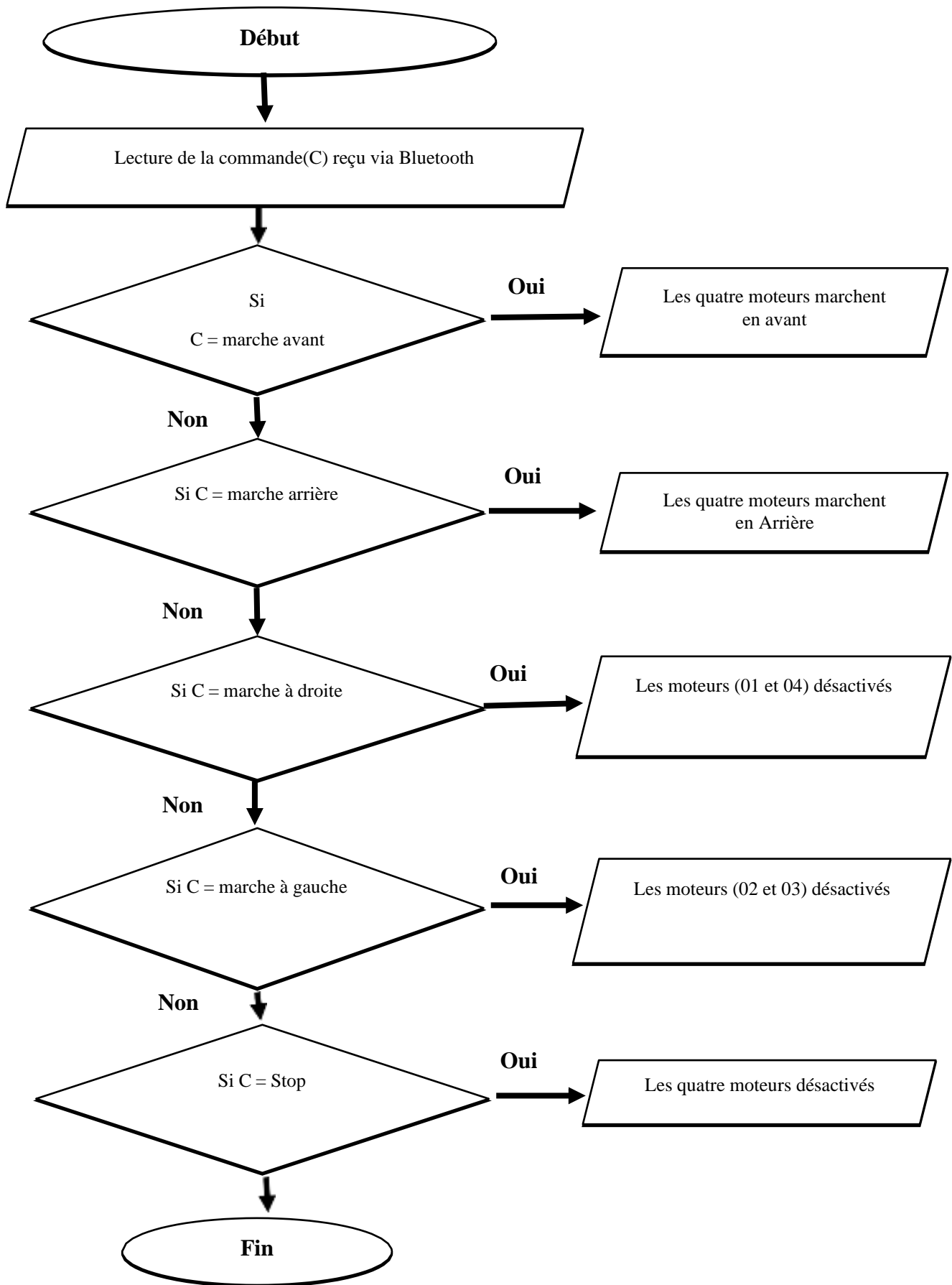


Figure III.13 : organigramme de la partie réceptrice

III.6.2.2.1. Description de l'organigramme

Tableau III.2 : entrée activée par geste

Mouvement de la main	L'entrée activée par le geste						Direction
	Pin 6(5)	Pin 5(8)	Pin 4(7)	Pin3 (3)	Pin2(4)	Pin 1(2)	
Stable	180	0	0	180	0	0	Stop
Incliner vers la droite	180	0	1	180	0	1	Marche en avant
Incliner vers la gauche	180	1	0	180	1	0	Marche arrière
Incliner vers l'arrière	180	0	0	180	0	1	Tourne à gauche
Incliner vers l'avant	180	0	1	180	0	0	Tourne à droite

Ici, les broches 1, 2, 3, 4, 5 et 6 sont les broches du moteur à courant continu lorsque l'accéléromètre est incliné, l'étalonnage des axes de l'accéléromètre est envoyé à l'Arduino et converti en valeur analogique

- Si la valeur de l'axe X comprise entre (264 ; 294), ainsi la valeur de l'axe Y entre (294 ; 330), la charge de la batterie est envoyée aux broches 1 et 4 du moteur à courant continu via le pilote des moteurs L293D et elle se déplace vers l'avant.
- Si la valeur de l'axe X est comprise entre (330 ; 394) et si la valeur de l'axe Y est comprise entre (294 ; 330), la charge de la batterie est envoyée aux broches 2 et 5 du moteur à courant continu via le pilote des moteurs L293D et elle se déplace en sens inverse.
- Si la valeur de l'axe x est comprise entre (294 ; 330) et si la valeur de l'axe y comprise entre (330 ; 400), la charge de la batterie est envoyée à la broche 1 du moteur DC via le pilote des moteurs L293D et elle se déplace vers la gauche
- Si la valeur de l'axe est comprise entre (294 ; 330) et si la valeur de l'axe y comprise entre (264 ; 294), la charge de la batterie est envoyée à la broche 4 du moteur à courant continu via le pilote des moteurs L293D et elle se déplace vers la gauche
- Si aucune des conditions ci-dessus ne satisfait le robot s'arrête de bouger.

Le code du programme de la partie réceptrice est illustré dans l'annexe (c).

III.7. Conclusion :

Cette partie a donné lieu la réalisation d'une voiture automatisée (robot a 4 roues) commandés gestuelles, ces commandes sont gérées à partir d'un accéléromètre (adx1345) et une carte Arduino Uno. L'ensemble travaux décrit dans ce chapitre sont :

1-conception des différents circuits de notre robot, réaliser à l'aide du logiciel Fritzing

2-le choix des composant convenable pour chaque bloc.

3-l'identification des paramètres d'alimentation et de commande et de puissance.

Le pilotage à distance est utilisé pour rendre le fonctionnement du robot simple et facile.

Les organigrammes sont réalisés dans le but d'expliquer des différentes étapes de la programmation des deux parties (émetteur, récepteur)

Conclusion

générale

Conclusion générale :

Au cours de ce projet et ce fil de temps, nous avons pu en créer un robot à partir de zéro, c'est-à-dire couvrir toute branche de la robotique, en partant de la mécanique (création du robot avec des pièces matériels, moteur, roues...etc.), ensuite le cote électronique du robot c'est-à-dire en attribue à ce robot le mouvement nécessaire qui lui permettra de manœuvrer dans le monde extérieur (en utilisant une carte Arduino et son IDE pour pouvoir attribuer ces différents mouvement et l'alimentation requise pour cela).

Nous avons choisi notre matériel en fonction de nos besoins puis essayé différentes solutions afin d'arriver aux à la réalisation de la commande. En revanche, l'utilisation du la carte Arduino programmable en langage C nous a permis de gagner en temps et en simplicité du programme.

Nous avons donc au final pu mener un vrai travail de conception de commande, ce qui s'apparente le plus à un travail d'ingénieur. De manière générale, ce projet nous a fait découvrir le monde de la robotique et toutes ses perspectives professionnelles.

Problèmes rencontrer

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie réalisation comme la configuration des modules Bluetooth émetteur et récepteur.

Cependant, on peut dire que malgré ces difficultés, les résultats o obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études.

Perspectives

Nous souhaitons vivement que ce projet puisse servir comme élément de base pour d'autres études plus approfondies pour le faire intégrer sous des systèmes plus complexes.

Malgré les lacunes restantes de notre robot nous sommes satisfaits du résultat, nous sommes conscients que de nombreuses améliorations pourraient être apportées, notamment sur le plan esthétique, mais nous sommes parvenus à résoudre de nombreux problèmes en travaillant de concert.

Enfin, ce projet nous a permet de nous familiariser avec la programmation des carte Arduino, mettant en application nos connaissances acquises durant nos études tout le long de

notre formation. Ainsi, il nous a permis d'améliorer nos connaissances en robotique et de maîtriser la nouvelle technologie.

Bibliographie

Références Bibliographiques

- [1] H. Hamdi, " introduction à la robotique", revue couverture robot, Université Mentouri Constantine, 2002.
- [2] M. Charles-Henri Eyraud, "Horloges astronomiques au tournant du XVIIIe siècle : de l'à-peu-près à la précision", thèse de doctorat, Université de Lyon, 2004.
- [3] <https://ifr.org/robot-history>, consulté le 10 septembre 2020.
- [4] K. Sahir, H. Bouatmane, "Etude et réalisation d'un bras de robot à base de carte Arduino", mémoire de fin d'étude, Université de Bejaia, 2019.
- [5] N. Nait Chaban, N. Moussaoui, " Conception et réalisations d'un robot mobile", mémoire de fin d'étude, Université Mouloud Maamri de Tizi ousou, 2015.
- [6] C. Ait Ali, " Commande d'un robot mobile via la vision artificiel", mémoire de fin d'étude, Université Mouloud Maamri de Tizi Ouzou, 2016.
- [7] H. Takhi, R. H. Attachi, " Conception et réalisation d'un robot mobile à base d'Arduino ", mémoire de fin d'étude, Université de Laghouat, 2014
- [8] P. Lain, " Robotique générale", Livre, Ellipses-Marketing, 1988
- [9] H. Zerfa, " Conception, Réalisation et Commande Floue d'un Robot Mobile", Mémoire de magister, Université des Sciences et de la Technologie d'Oran, 2013.
- [10] F. R. Hadj-Djilani, N. Bennabi, " Commande optimal appliqué à un robot mobile", mémoire de fin d'études, Université Djilali Bounaama Khemis Miliana, 2017.
- [11] <https://sites.google.com/site/technovhugogassinc31dg1/differents-domaines-d-application-de-la-robotique-actuelle>, consulté le 05 octobre 2020.
- [12] A. Berriche, exposé sur les robots, Université de Paris 13, 2016.
- [13] <https://c8.alamy.com/compfr/d899gj/horloge-a-eau-clepsydre-d899gj.jpg>, consulté le 6 octobre 2020.
- [14] <https://s3.amazonaws.com/s3.timetoast.com/public/uploads/photos/11734393/canard.jpg>, consulté le 6 octobre 2020.
- [15] https://www.industrie-techno.com/mediatheque/5/1/8/000025815_600x400_c.jpg, consulté le 6 octobre 2020.

- [16] Astalaseven, Eskimon, Olyte, "Arduino pour bien commencer en électronique et en programmation ", Livre electronique, 2012.
- [17] <https://maker.pro/arduino/tutorial/a-comparison-of-popular-arduino-boards> , consulté le 20/10/2020.
- [18] C. Tavernier, "Arduino applications avancées", Livre, Edition Dunod, 2012.
- [19] <https://microcontrollerslab.com/pid-controller-implementation-using-arduino/>, consulté le 06/07/2021
- [20] <https://www.orcas-maker.com/post/composants-arduino> , consulté le 23/10/2020.
- [21] A. Moussaoui, "Conception et réalisation d'un bras manipulateur commandé par l'Arduino Méga 2560", Mémoire de Master, Université M'hamed Bougara de Boumerdes, 2017.
- [22] A. Krama, A. Gougui, " Etude et réalisation d'une carte de contrôle par Arduino via le système Android ", Meoire de Mster, Universite Kasdi Merbah Ouargla, 2015.
- [23] datasheet du capteur ADXL345-EP
<https://www.analog.com/media/entechnical-documentation/data-sheets/ADXL345-EP.pdf>, consulté le 08/12/2020.
- [24] <http://instructables.com/Accelerometer-Gyro-Tutorial/>, consulté le 08/12/2020.
- [25] <https://knowledge.parcours-performance.com/arduino-bluetooth-hc-05-hc-06/> consulté le 08/12/2020.
- [26] HC-05 Bluetooth Module User's Manual V1.0, 2011.
- [27] <https://www.gotronic.fr/art-module-bluetooth-hc05-26097.htm>,consulté le 08/12/2020
- [28] H. Weisslinger, S. Landrault " Arduino : Premiers pas en informatique embarquée", Livre, Leblog d'Eskimon, 2014.
- [29] https://www.dzduino.com/index.php?route=product/product&product_id=805, consulté , le 09/12/2020.
- [30] <https://arduino103.blogspot.com/2011/06/controle-moteur-dc-via-l293d-h-bridge.html> consulté le 09/12/2020.
- [31] E. Gaucheron, « Les moteurs électriques ... pour mieux les piloter et les protéger », Schneider Electric, CT 207, 2004.
- [32] <https://www.locoduino.org/spip.php?article69>, consulté le 08/12/2020
- [33] <https://www.cours-gratuit.com/cours-arduino/support-de-cours-carte-arduino-uno-pdf>, consulté le 06/07/2021
- [34] http://microsann.com/images/Atelier_Joomla/Fiches_PDF/La_carte_Arduino_UNO.pdf, consulté le 06/07/2021

Annexes

Annexe (A)



L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

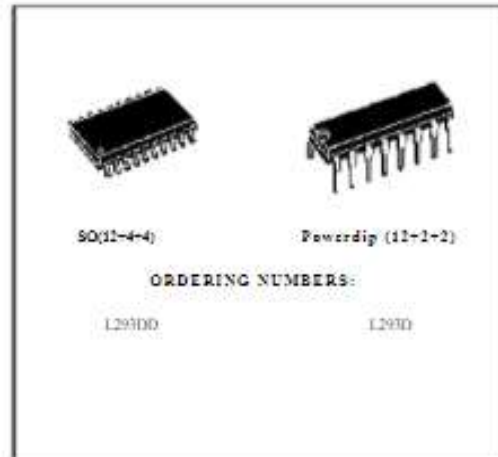
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

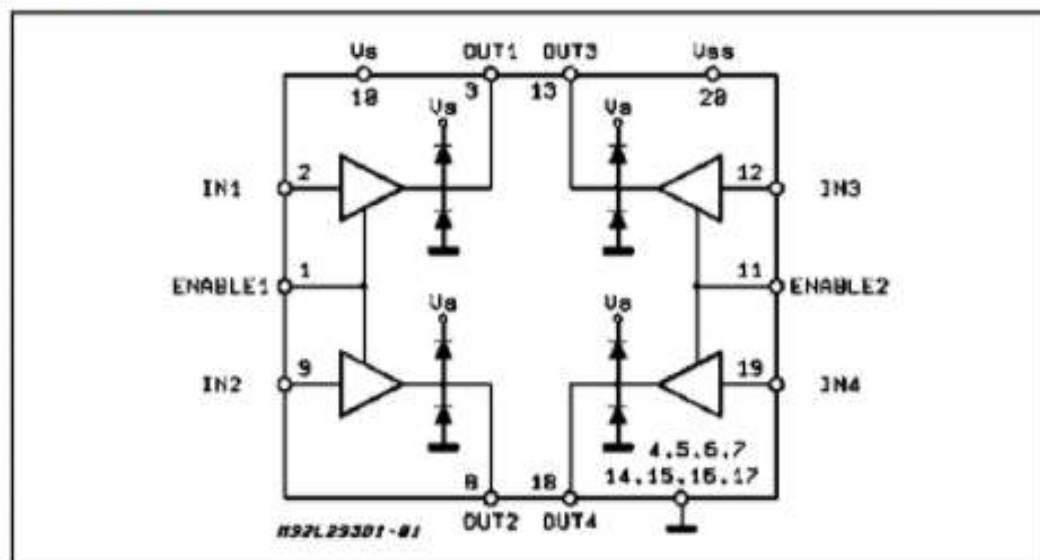
This device is suitable for use in switching applications at frequencies up to 5 kHz.



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM



L293D - L293DD

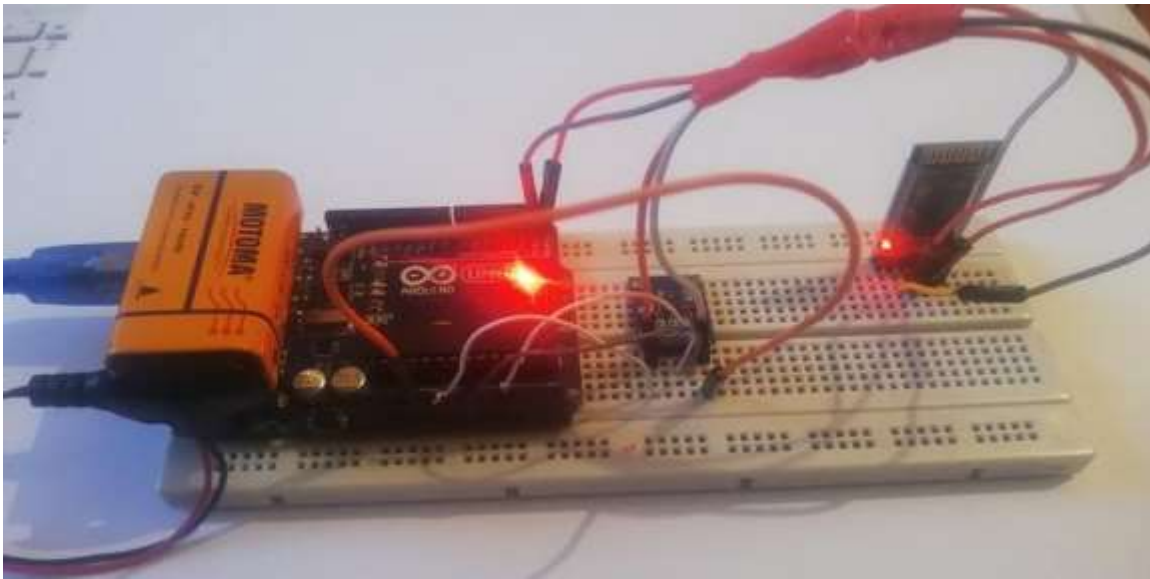
ELECTRICAL CHARACTERISTICS (for each channel, $V_S=24\text{ V}$, $V_{SS}=5\text{ V}$, $T_{amb}=25$ otherwise specified)

$^{\circ}\text{C}$, unless

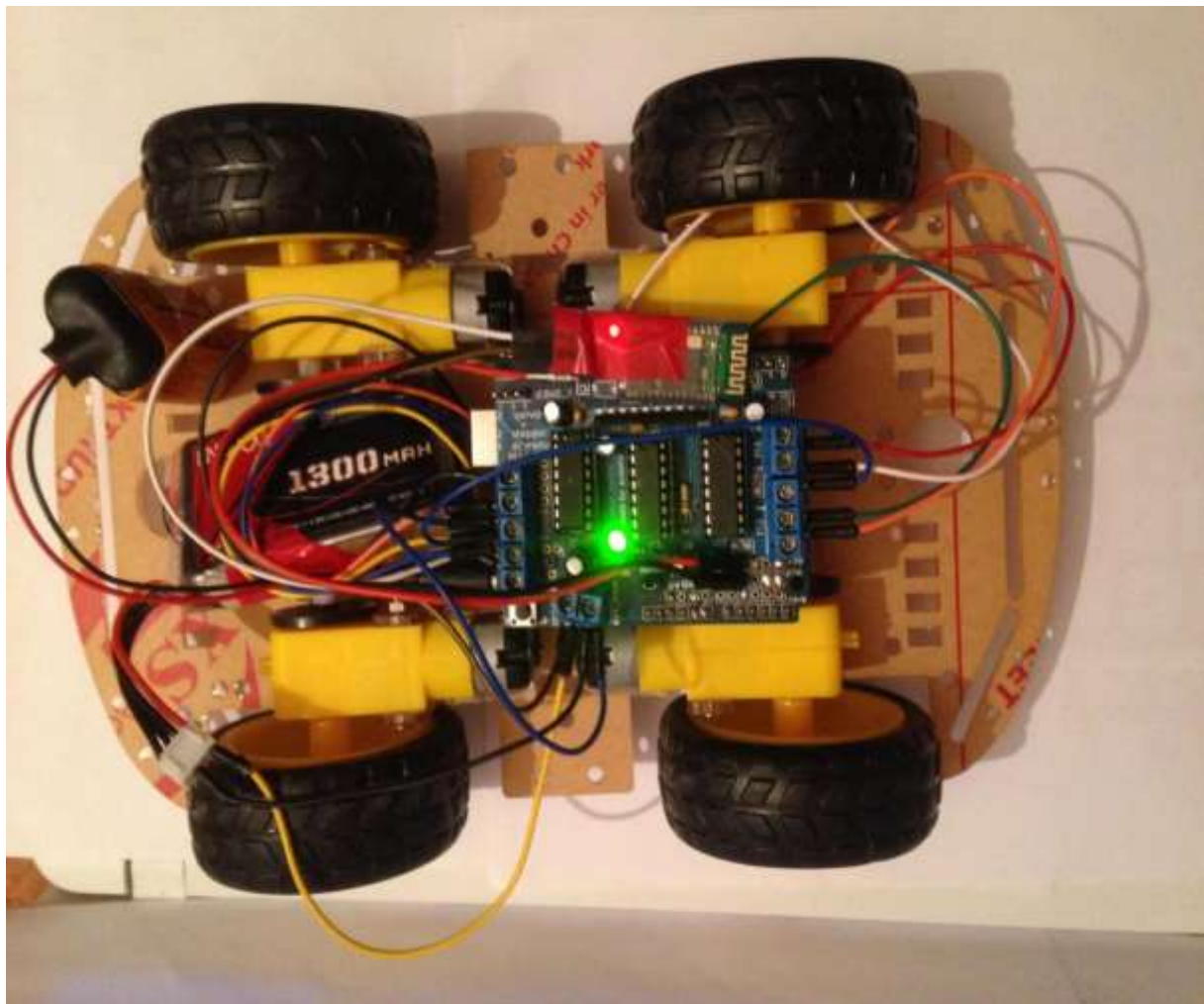
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 10)		V_{SS}		36	V
V_{SS}	Logic Supply Voltage (pin 20)		4.5		36	V
I_S	Total Quiescent Supply Current (pin 10)	$V_i=L; I_o=0; V_{en}=H$		2	6	mA
		$V_i=H; I_o=0; V_{en}=H$		16	24	mA
		$V_{en}=L$			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	$V_i=L; I_o=0; V_{en}=H$		44	60	mA
		$V_i=H; I_o=0; V_{en}=H$		16	22	mA
		$V_{en}=L$		16	24	mA
V_{IL}	Input Low Voltage (pin 2, 9, 12, 19)		-0.3		1.5	V
V_{IH}	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{IL}	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_{IL}=1.5\text{ V}$			-10	μA
I_{IH}	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3\text{ V} \leq V_{IH} \leq V_{SS} - 0.6\text{ V}$		30	100	μA
V_{enL}	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
V_{enH}	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{enL}	Low Voltage Enable Current (pin 1, 11)	$V_{enL}=1.5\text{ V}$		-30	-100	μA
I_{enH}	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{enH} \leq V_{SS} - 0.6\text{ V}$			± 10	μA
$V_{CH(sat)}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_o = -0.6\text{ A}$		1.4	1.8	V
$V_{CH(sat)}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	$I_o = +0.6\text{ A}$		1.2	1.8	V
V_f	Clamp Diode Forward Voltage	$I_o = 600\text{ nA}$		1.3		V
t_r	Rise Time (*)	0.1 to 0.9 V_o		250		ns
t_f	Fall Time (*)	0.9 to 0.1 V_o		250		ns
t_{on}	Turn-on Delay (*)	0.5 V_i to 0.5 V_o		750		ns
t_{off}	Turn-off Delay (*)	0.5 V_i to 0.5 V_o		200		ns

(*) See fig. 1.

Annexe (B)



Partie émettrice de robot



Partie réceptrice de robot

Annexe (C)



```
sketch_jan09a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide
sketch_jan09a $
//Receiver code
//all the best
#include
SoftwareSerial mybt(2,3);
char m=0,n=0;

void setup() {
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);
pinMode(13, OUTPUT);
Serial.begin(9600);
mybt.begin(9600);
}

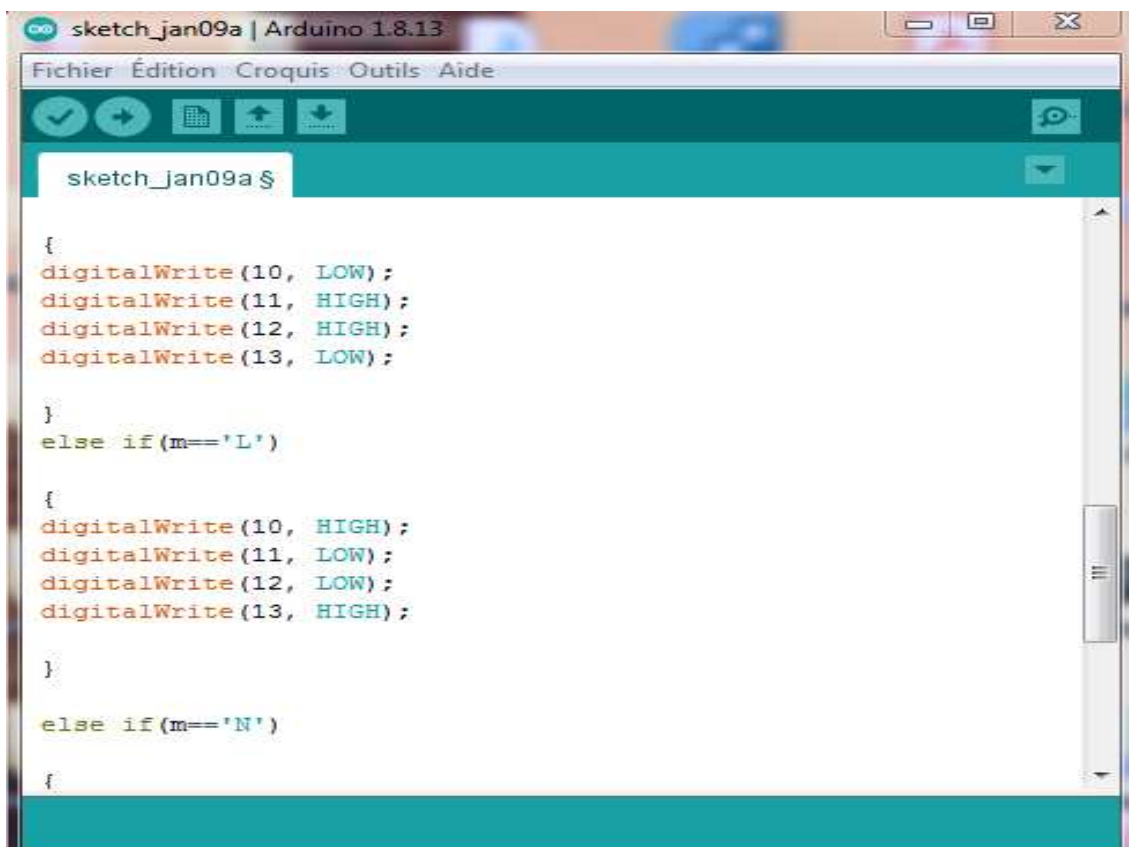
void loop() {
if(mybt.available()>0)
{
m= mybt.read();
Serial.println(m);
if(m=='F')
```



```
sketch_jan09a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide
sketch_jan09a $
{
digitalWrite(10, HIGH);
digitalWrite(11, LOW);
digitalWrite(12, HIGH);
digitalWrite(13, LOW);
}

else if(m=='B')
{
digitalWrite(10, LOW);
digitalWrite(11, HIGH);
digitalWrite(12, LOW);
digitalWrite(13, HIGH);
}

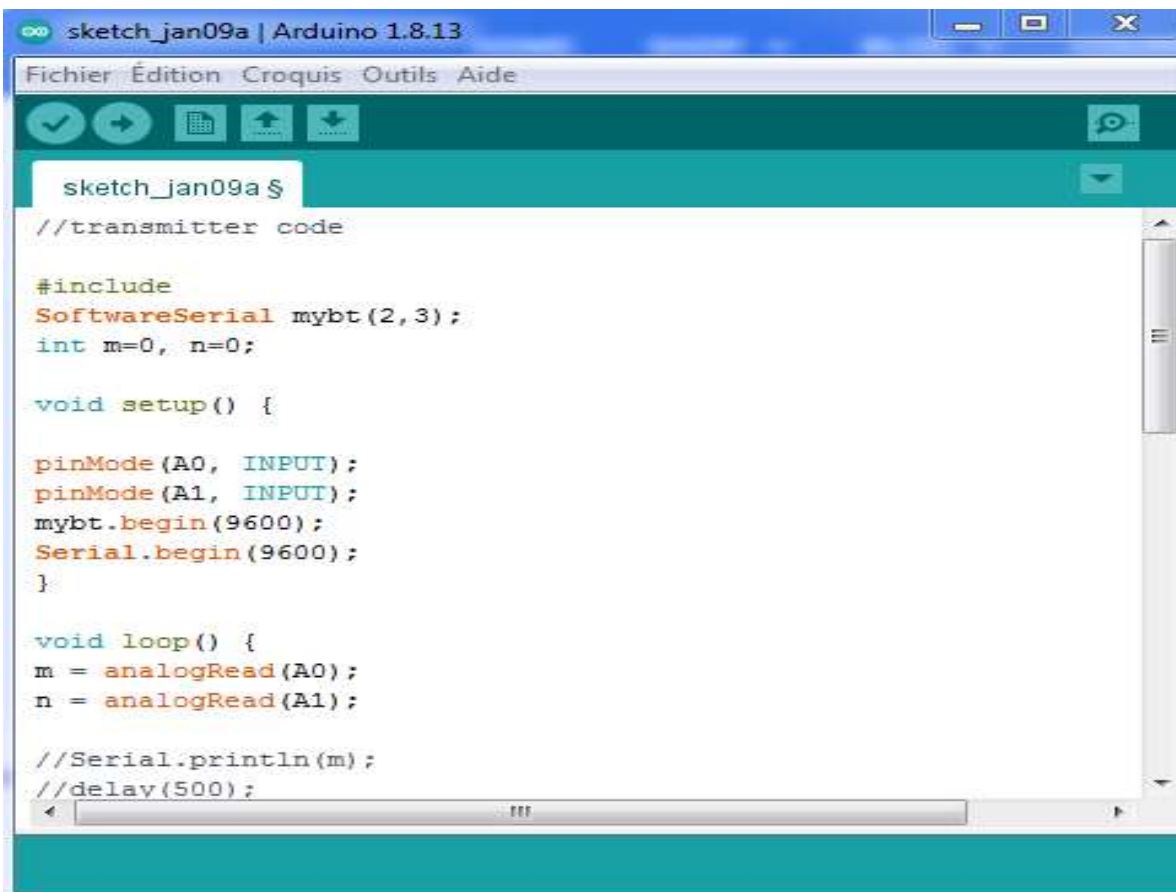
else if(m=='R')
```

```
sketch_jan09a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide
sketch_jan09a $
{
digitalWrite(10, LOW);
digitalWrite(11, HIGH);
digitalWrite(12, HIGH);
digitalWrite(13, LOW);
}
else if(m=='L')
{
digitalWrite(10, HIGH);
digitalWrite(11, LOW);
digitalWrite(12, LOW);
digitalWrite(13, HIGH);
}
else if(m=='N')
{
```



```
sketch_jan09a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide
sketch_jan09a $
else if(m=='N')
{
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, LOW);
digitalWrite(13, LOW);
}
else
{
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, LOW);
digitalWrite(13, LOW);
}
}
} }
```



```
sketch_jan09a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide

sketch_jan09a $
//transmitter code

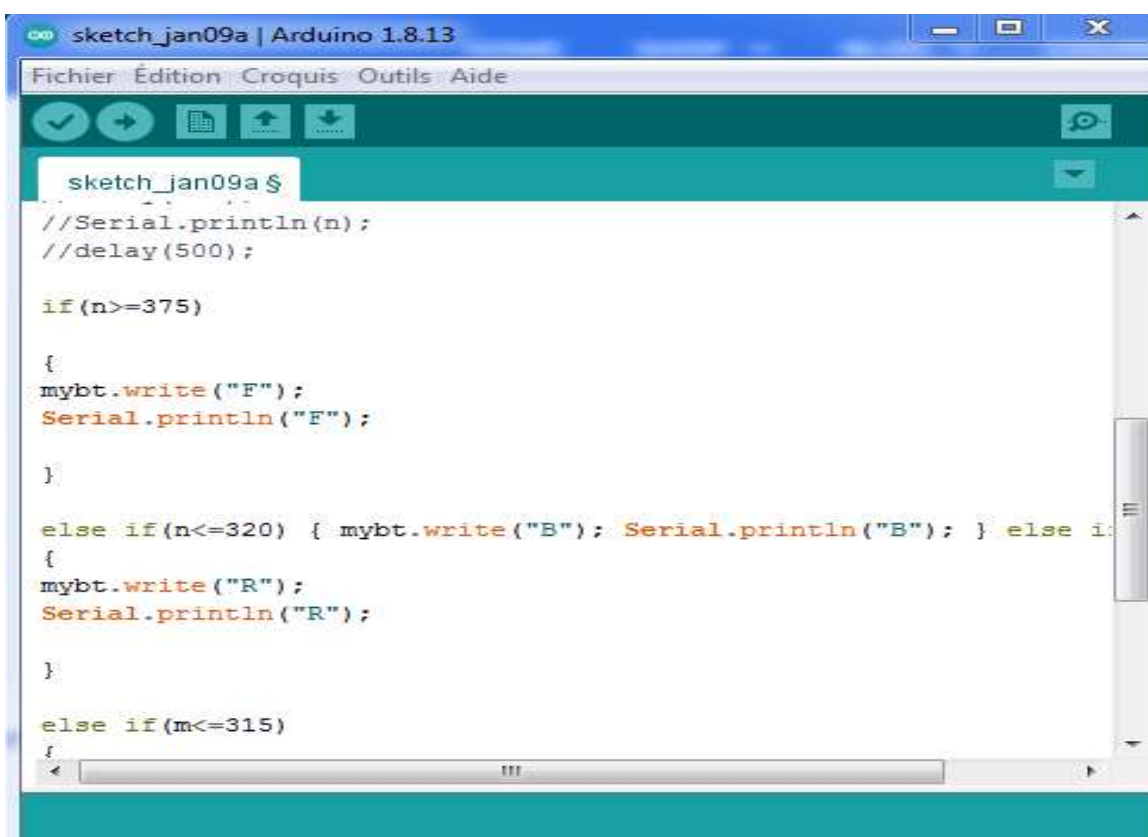
#include
SoftwareSerial mybt(2,3);
int m=0, n=0;

void setup() {

pinMode(A0, INPUT);
pinMode(A1, INPUT);
mybt.begin(9600);
Serial.begin(9600);
}

void loop() {
m = analogRead(A0);
n = analogRead(A1);

//Serial.println(m);
//delay(500);
```



```
sketch_jan09a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide

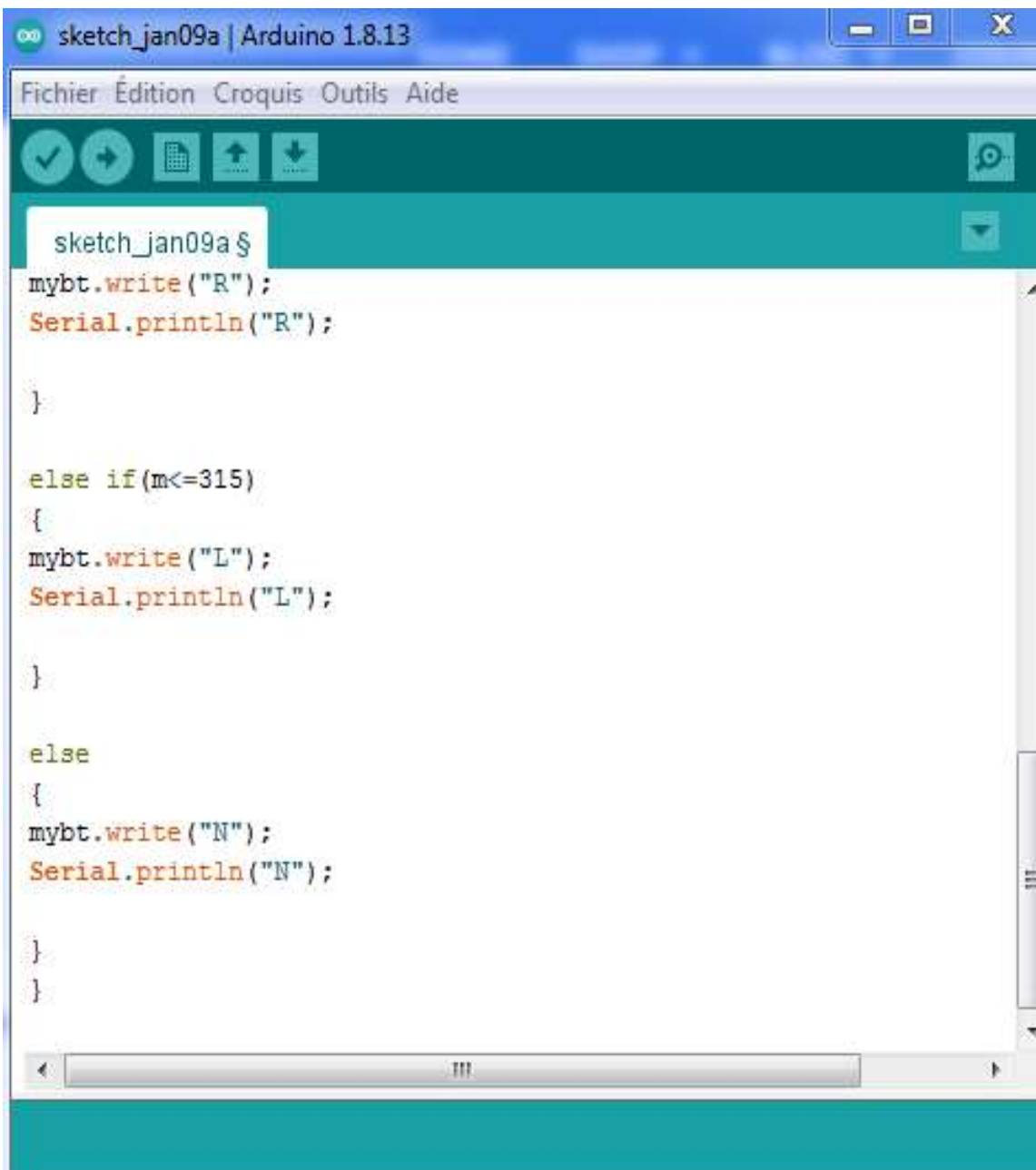
sketch_jan09a $
//Serial.println(n);
//delay(500);

if(n>=375)

{
mybt.write("F");
Serial.println("F");
}

else if(n<=320) { mybt.write("B"); Serial.println("B"); } else if(n>320 & n<375) { mybt.write("R"); Serial.println("R"); }

else if(m<=315)
```



```
sketch_jan09a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide
sketch_jan09a $
mybt.write("R");
Serial.println("R");

}

else if(m<=315)
{
mybt.write("L");
Serial.println("L");

}

else
{
mybt.write("N");
Serial.println("N");

}
}
```