



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique et informatique industrielle

Réf. :

Présenté et soutenu par :
Tauririt Abd elhamid

Le : 2020

Conception et Réalisation d'un Robot Mobile à Base d'Arduino

Jury :				
.	Meghrebi Hassina	.	Université de Biskra	Président
.	Mechgoug Raihane	.	Université de Biskra	Rapporteur
.	Rechid Naima	.	Université de Biskra	Examineur

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne
Démocratique et Populaire
وزارة التعليم العالي والبحث
العلمي



Ministère de l'enseignement Supérieur et de la recherche scientifique

Université Mohamed Khider Biskra Faculté

des Sciences et de la Technologie

Filière : Automatique

Option : Automatique et informatique industrielle

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme :

MASTER

Thème

**Conception et Réalisation d'un Robot Mobile à
Base d'Arduino**

Présenté par:

Tauririt abd elhamide

Avis favorable de l'encadreur:

Dr. MECHGOUG Raihane

Avis favorable du Président du Jury

Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne
Démocratique et Populaire
وزارة التعليم العالي والبحث
العلمي



Ministère de l'enseignement Supérieur et de la recherche scientifique

Université Mohamed Khider Biskra

Faculté des Sciences et de la Technologie

Département de Génie Electrique

Filière : Automatique

Option : Automatique et informatique industrielle

Thème

Conception et Réalisation d'un Robot Mobile à Base d'Arduino

Présenté par : Taouririt abd elhamide

Dirigé par : Dr. MECHGOUG Raihane

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicace

À la plus belle créature que Dieu a créée sur terre.

À cette source de tendresse, de patience et de

Générosité,
À mes parents!

Aucune dédicace ne saurait exprimer l'amour,
L'estime, le dévouement et le respect que j'ai
Toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et

Nuit pour mon éducation et mon bien être.

Ce travail est le fruit de vos sacrifices que vous

Avez consentis pour mon éducation et ma

Formation.



Remerciements

Tout d'abord, je voudrais remercier Dieu Tout-Puissant de m'avoir donné le courage et la force de mener à bien ce travail, et je nommerai mon superviseur le Dr Mechgoug Raihan, et mon ami Saif Eddine Bouzid, ainsi que tous mes camarades étudiants.

Je remercie ma famille, mes amis et tous mes proches de m'avoir accompagné tout au long de cette aventure. Sans leur aide, cette entreprise n'aurait pas réussi, en espérant que le résultat final serait à la hauteur de leurs attentes.



Résumé

Dans ce projet on a réalisé un système de commande d'un robot mobile à 4 roues. Ce système contrôle le robot d'une façon manuelle ou autonome, avec ou sans PC, local ou à distance (vocale, Bluetooth). On a utilisé l'Arduino UNO, capteur ultrason et un circuit de pilotage des moteurs à courant continue. On a pu aussi introduire une technique moderne de commande en utilisant un téléphone portable avec l'application App Inventer.

ملخص:

لقد قمنا في المشروع بإنجاز نظام تحكم لروبوت متحرك بأربع عجلات. يسمح هذا النظام بالتحكم بالروبوت يدويا أو بشكل مستقل مع أو بدون جهاز كمبيوتر محليا أو عن بعد (الصوت ، وتقنية بلوتوث). استخدمنا لذلك بطاقة الاردوينو، وأجهزة الاستشعار بالموجات فوق الصوتية ودائرة التحكم في محركات تيار مستمر. و قد قمنا كذلك بإدخال تقنية التحكم الحديثة باستخدام الهاتف المحمول باستخدام تطبيق آب انفتور.



Tableau des Matière

Dédicace.....	I
Remerciements.....	II
Résumé.....	III
Tableau des Matière.....	IV
Liste des Figures.....	VI
Liste des Tableau.....	VIII
Introduction Général.....	IX
Chapitre I : Généralités sur la robotique mobile.	1
I.1 Introduction	1
I.2 Généralités sur la robotique mobile.....	1
Définition.....	1
I.2.2 Historique	1
I.2.3 Modèle du robot	2
I.3. Type de robots.....	4
I.3.1 Robots manipulateurs	4
I.3.2 Robots mobiles	4
I.3.2.1 Robots mobiles à roues.....	5
I.4 Les moteurs :	10
I.4.1 Les moteurs à courant continu :	10
I.4.2 Le moteur pas à pas :	11
I.4.3 Les servomoteurs :.....	11
I.5 Évitement d’obstacles :	12
I.7. Conclusion	13
Chapitre 2 : Arduino UNO.	14
II.1. Introduction	14
II.2. Historique.....	14
II.3. Applications.....	15
II.4.Arduino Uno.....	16
II.4.1. Caractéristiques de la carte Arduino Uno	17

II.4.2. Les composantes de la carte Arduino Uno.....	18
II.5. Présentation de l'Espace de développement Intégré (EDI) Arduino.....	21
II.5.1.. Description de l'interface.....	21
II.5.2. Description de la structure d'un programme.....	22
II.6. Conclusion.....	25
Chapitre III : Simulation et Réalisation d'un Robot mobile.....	26
III. 1 Introduction	26
III.2. Cahier de charge	26
III.3. Principaux éléments constituant le système.....	27
III.3.1. La carte ARDUINO UNO	28
III.3.2. Le driver pour moteur à courant continu L293B	28
III.3.3. Double ponts H Contrôleurs de moteur, le L298N	29
III.3.4. Capteur à ultrasons HC-SR04	30
III.3.5. Mini moteur motoréducteur 6V	31
III.3.6. Les roues.....	31
III.3.7.Le servomoteur de l'ultrason HC-SR04	32
III.3.8.Bluetooth Transcribe Module HC-06	32
III.3.9. Arduino capteur Shield V4.0	33
III.4. La partie logicielle	35
III.4.1.Le logiciel proteus	35
III.4.2. ISIS	35.
III.4.3. La simulation du projet	35
III.5. Réalisation de robot mobile.....	39
III.6 Présentation d'App inventor.....	42
III.6.1 Créé un projet :.....	43
III.7. La partie commande	45
III.7.1.La commande à distance de robot	45
III.8.Conclusion	48
Conclusion générale	49
Bibliographie.....	50
Annexe A.....	53



Liste des figures :

Chapitre I : Généralités sur la robotique mobile.

Figure I.1 :ROBOT MOBILE	5
Figure I. 2 :SCHEMA D'UN ROBOT DE TYPE UNI-CYCLE	6
Figure I. 3:ROBOT MOBILE TRICYCLE.....	6
Figure I.4 :SCHEMA D'UN ROBOT MOBILE TIPE VOITURE.....	7
Figure I.5 :SCHEMA D'UN ROBOT MOBILE OMNIDIRECTIONNEL	7
Figure I.6 :EXEMPLES DE ROBOTS MOBILES A CHENILLES	9
Figure I. 7: EXEMPLES DE ROBOTS MOBILES MARCHEURS.	10
Figure I.8 : MOTEUR A COURANT CONTINU	11
Figure I.9 :EXEMPLE DE MOTEUR PAS A PAS.....	11
Figure I.10 :UN SERVOMOTEUR	12

Chapitre II : ARDUINO UNO

Figure II. 1 : DESCRIPTION DE LA CARTE ARDUINO UNO.....	17
Figure II.2 : LES COMPOSANT DE LA CARTE ARDUINO UNO	18
Figure II.3: PRESENTATION DES ELEMENTS DE L'ARDUINO SOFTWARE	21
Figure II.4: MODULE TERMINAL SERIE.....	22
Figure II.5 : FENETRE GRAPHIQUE DE L'EDI.....	23

Chapitre III : Simulation et Réalisation d'un Robot mobile

Figure III.1 :Méthodes de commande de robot mobile	27
Figure III.2 :le matériel utilisé.....	28
Figure III.3 :Circuit L293B	29
Figure III.4:Le brochage du Module L298N double ponts en H	29
Figure III.5 :CAPTEUR ULTRASON HC-SR 04.....	30
Figure III.6 :Chronogramme du capteur ultrason HC-SR04.....	30
Figure III.7:Mini moteur .Motorèducteur	31
Figure III.8 : la roue Tamiya	31
Figure III.9 : Servomoteur.	32
Figure III.10 :Impulsion de commande d'un Servomoteur.	32
Figure III.11 : module Bluetooth HC-06	33
Figure III.12 :Module Capteur shield d'Arduino	34
Figure III.13 : Photo du projet réalisé	34
Figure III.14 : la simulation d'un robot mobile AUTONOME par le logiciel ISIS.....	36
Figure III.15: la commande du robot par Bluetooth en simulation	36
Figure III.16 : Interface de logiciel IDE avec explication des différents parties	17
Figure III.17: Organigramme de commande D'un robot mobile autonome par IDE.	38
Figure III.18 :Organigramme de commande D'un robot mobile par Bluetooth	39
Figure III.19 :placement des moteurs et de module L298N sur chassis.....	39
Figure III.20: Arduino avec le module L298 et 4 moteur.	40
Figure III.21 : placement de l'Arduino et le module ultrason sur le robot.....	40

Figure III.22: Arduino avec l'ultrason et le servomoteur.....	41
Figure III.23: le brochage de Bluetooth avec Arduino.....	41
Figure III.24 :pile BAKTH.	42
Figure III.25 site mit app inventor.	43
Figure III.26 : première interface de la création App Inventor	45
Figure III.27 :Application Android pour commande de robot.	46
Figure III.28 Interface du logiciel App Inventer pour applications Androide.....	46
Figure III.29 : la commande par vocale	47
Figure III.30 :Application Android pour commande de robot vocale.....	47
Figure III.31 : Interface du logiciel App Inventer pour applications Android vocale.....	48

Liste des tableaux

Tableau I.1 : Comparaison des différents types de robots mobile.....	8
Tableau II.1 : Les caractéristiques des cartes Arduino.....	16
Tableau II.2 : Caractéristiques de la carte Arduino Uno	17
Tableau III.3 : Les différents types de données.....	24

Introduction Général

La robotique est un ensemble de disciplines (mécanique, électronique, automatique, informatique), elle se subdivise en deux types : les robots industrielle est les robots mobiles.

Les robots industriels sont généralement fixes, ils sont utilisés dans des nombreuses applications industrielles. Les robots mobiles ne sont pas fixes, ils sont classifiés selon la locomotion en robot marcheurs a Rous, a chenilles

Dans le vaste domaine de la robotique mobile, l'étude du déplacement a une grande Importance. Les robots mobiles sont des robots qui peuvent se déplacer d'un endroit à un autre de manière autonome, qui est sans l'aide d'opérateurs humains externes. Contrairement à la majorité des robots industriels qui peuvent se déplacer que dans un espace de travail spécifique, robots mobiles ont la particularité de se déplacer librement dans un espace de travail prédéfini pour atteindre leurs objectifs souhaités.

Le but de ce travail est de réaliser un système de commande d'un robot mobile à 4 roues en vue d'évitement d'obstacles sans tenir en compte certains paramètres géométriques telles que dimensions du robot ou formes des obstacles car cela nous amènera à un axe de recherche très compliqué.

Notre robot mobile est basé sur une carte de capteurs émetteur/récepteur ultrasonique et une carte de commande et d'interfaçage à base d'Arduino, avec des moteurs a courant continue qui assure le déplacement du robot à travers un mécanisme. Pour cela, notre mémoire est organisé comme suit :

- ✚ Le premier chapitre présente une introduction sur la robotique : définitions, historique, capteurs et différents techniques de commande.

- ✚ Le deuxième chapitre est une explication de la carte de commande Arduino.

- ✚ Le troisième chapitre présente principalement les deux parties constituant le système de commande : la partie matérielle (hardware) et la partie logicielle (software).

- ✚ Finalement, nous terminons notre travail par une conclusion générale qui présente le bilan de ce travail

Chapitre I

Généralités sur la robotique mobile

Chapitre I : Généralités sur la robotique mobile.

I.1 Introduction

La robotique a été un contributeur dominant pour le développement de la société humaine au cours des années. Elle est un domaine qui a besoin de la synergie d'une variété de domaines scientifiques comme le génie mécanique, génie électrique-électronique, l'ingénierie de contrôle, génie informatique, en génie du capteur, et d'autres. Les robots et les autres fonctions automatisées machines ont à vivre ensemble avec les gens.

Dans cette symbiose, les besoins humains et les préférences devraient être principalement respectées, intégrées, et mises en œuvre.

À cette fin, des robots modernes, en particulier des robots mobiles à roues ou pattes, incorporent et de réaliser d'une manière ciblée et rentable la perception cycle de l'action. L'utilisation des robots mobiles est aujourd'hui couramment envisagée pour l'automatisation de nombreuses tâches. Celles-ci sont particulièrement diversifiées: le nettoyage, le transport dans les ateliers automatisé, l'agriculture, l'exploitation des mines, l'assistance aux personnes handicapées et l'exploration de milieux hostiles en sont quelques exemples.

Les travaux en robotique ont pour but de concevoir et de construire des machines capables d'évoluer et d'interagir avec un environnement physique de manière à accomplir les différentes tâches pour lesquelles elles ont été créés.

I.2 Généralités sur la robotique mobile

I.2.1. Définition

Le terme Robot prend son origine du mot «**slave**» qui veut dire en russe travail ou corvée en tchèque. Il désigne aussi une machine à l'aspect humain, capable de se mouvoir et d'agir, d'effectuer certaines opérations, parfois de modifier lui-même son cycle de fonctionnement et d'exercer un certain choix.[1]

I.2.2 Historique

La robotique est passée par plusieurs générations comme suit :

- ❖ 1947 : Premier manipulateur électrique télé- opéré.
- ❖ 1954 : Premier robot programmable.
- ❖ m1961 : Utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une Chaîne de montage de General Motors.
- ❖ 1961m : Premier robot avec contrôle en effort.
- ❖ 1963 : Utilisation de la vision pour commander un robot.
- ❖ 1978 : Le robot ARGOS. Développé à l'Université Paul Sabatier de Toulouse (France). Le robot ARGOS simule la navigation d'un robot mobile équipé d'un système de vision au fur et à mesure de ses déplacements.
- ❖ 1979 : Le robot HILARE. Les chercheurs du L.A.A.S. de Toulouse (France) étudient la planification des trajectoires d'un robot mobile ponctuel, dans un environnement totalement connu.
- ❖ 1981 : Le robot VESA. Ce robot, construit à l'I.N.S.A (France). de Rennes, est équipée d'un arceau de sécurité pour réaliser la détection d'obstacles dans un environnement totalement inconnu.
- ❖ 1984 : Le robot FLAKEY est équipé de deux roues motrices avec encodeurs, mais sa vitesse maximale est de 66 cm/ms au lieu de quelques centimètres par seconde. Ce robot est capable de naviguer dans des environnements réels.
- ❖ 1993 : Le robot ERRATIC a été conçu par Kurt comme un robot mobile de faible coût pour ses cours de robotique.
- ❖ Les robots mobiles actuels : A présent la plupart des travaux de recherche portent sur les problèmes de perception. La planification de trajectoires, l'analyse et la modélisation de l'environnement de robot, appliqué sur des robots mobiles commerciaux. Également la recherche actuelle sur la conception mécanique des robots mobiles pour des applications hautement spécialisées, comme l'exploration sous-marine, les robots volants et le micro robots.[1]

I.2.3 Modèle du robot

I.2.3.1 Modèle cinématique

La cinématique complète la modélisation géométrique en établissant les relations entre les vitesses des paramètres articulaire (\dot{q}) et le couple vitesse de rotation, vitesse d'un point de l'effecteur que l'on désignera pour l'instant par la notation ($\dot{s} = J\dot{q}$) La propriété évidente du modèle cinématique est sa linéarité par rapport aux vitesses. Cinématique Robot offres avec la

configuration des robots dans leur espace de travail relations entre leurs paramètres géométriques et les contraintes imposées dans leurs trajectoires.

Les équations cinématiques dépendent de la structure géométrique du robot. Par exemple un robot fixe peut avoir un cartésien, cylindrique, sphérique, ou articulé la structure, et un robot mobile peut avoir une, deux, trois, ou plus, avec ou sans roues contraintes dans leur mouvement.

L'étude de la cinématique est une condition préalable fondamentale pour l'étude de la dynamique, les caractéristiques de stabilité, et la commande du robot. Le développement de nouvelles structures spécialisées et cinématiques robotisés est encore un sujet de la recherche continue, vers la fin de la construction des robots qui peuvent effectuer plus tâches sophistiquées et complexes dans les applications industrielles et sociétales [2].

a) Modèle cinématique direct (MCD) : Le modèle cinématique direct permet de déterminer la vitesse d'un point de l'effecteur et la vitesse de rotation en fonction des vitesses articulaire (\dot{q}_i), $i = 1$ à n .

(M C D) est de la forme [3] : $\dot{x} = J(q)\dot{q}$

\dot{x} : Vitesse de l'effecteur par rapport au repère de base.

q : Vitesse des coordonnées généralisées.

J : c'est une matrice Jacobéenne.

b) Modèle cinématique inverse (MCI) : Le but de ce modèle consiste à trouver les vitesses articulaires qui génèrent la vitesse désirée du point terminal. Sa détermination consiste à inverser la matrice Jacobéenne, qui n'est pas toujours possible.

Le modèle cinématique inverse est de la forme : $\dot{q} = J^{-1} \dot{x}$.

I.2.3.2 Modèle dynamique

On s'intéresse ici aux efforts actionneurs produits par les mouvements du S. M. A. Il s'agit d'établir les équations différentielles non linéaires qui relient les efforts Actionneurs $\Gamma_i(t)$ aux variables articulaires $q_i(t)$, aux vitesses articulaires $\dot{q}_i(t)$ et aux accélérations articulaires $\ddot{q}_i(t)$.

L'ensemble de ces équations constitue ce qu'il est d'usage d'appeler le modèle dynamique du manipulateur. Pour cela il existe plusieurs formalisme de modélisation telle que, le

formalisme de Newton-Euler, le formalisme de Lagrange-Euler, le principe d'Alembert, etc.... [2]

I.3. Type de robots

On va située dans ce chapitre les différents types des robots. Il existe deux grandes familles de robots sont :

- Les robots manipulateurs.
- Les robots mobiles

I.3.1 Robots manipulateurs

En robotique un manipulateur est un dispositif utilisé pour manipuler les matières sans contact direct.

Les applications étaient à l'origine pour traiter les matières radioactives ou biologiques dangereuses, à l'aide des bras robotisés, ou ils ont été utilisés dans des endroits inaccessibles. Dans des développements plus récents, ils ont été utilisés dans des applications telles que la chirurgie robotique assistée et dans l'espace. Il est un mécanisme en forme de bras qui se compose d'une série de segments, généralement coulissant ou articulé, qui saisissent et déplacer des objets avec un certain nombre de degrés de liberté [4].

I.3.2 Robots mobiles

L'utilisation des robots mobiles est aujourd'hui couramment envisagée pour l'automatisation de nombreuses tâches. Celles-ci sont particulièrement diversifiées: le nettoyage, le transport dans les ateliers automatisés, l'agriculture, l'exploitation des mines, l'assistance aux personnes handicapées et l'exploration de milieux hostiles en sont quelques exemples. L'architecture mécanique d'un robot mobile dépend largement des spécificités de la mission à effectuer et de l'environnement de travail. Ces données conditionnent entre autres le choix d'un système de locomotion approprié. Trois configurations fondamentales peuvent être distinguées: les robots à roues ou à chenilles, les robots à pattes et les corps articulés (Figure.I.1). Des systèmes hybrides sont également possibles [5].



Figure I.1 : Robot mobile

Dans ce chapitre, la discussion sera basée aux robots mobiles à roues et leurs caractéristiques.

I.3.2.1 Robots mobiles à roues

La mobilité par roues est la structure mécanique la plus simple et la plus utilisée. Ce type de robot assure un déplacement avec une accélération et une vitesse rapide mais nécessite un sol relativement plat. On distingue plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées.

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées. Nous citerons ici les quatre classes principales de robots à roues[6].

a) Robot uni-cycle : Un robot du type monocycle est actionné par deux roues indépendantes, C'est un robot non-holonome, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

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. Le schéma d'un robot de type uni cycle est donné à la figure I.2. [6].

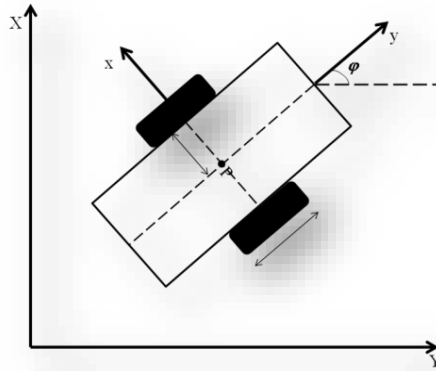


Figure I. 2 : Schéma d'un robot de type uni-cycle

b) Robot 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. (Figure I.3)[6]

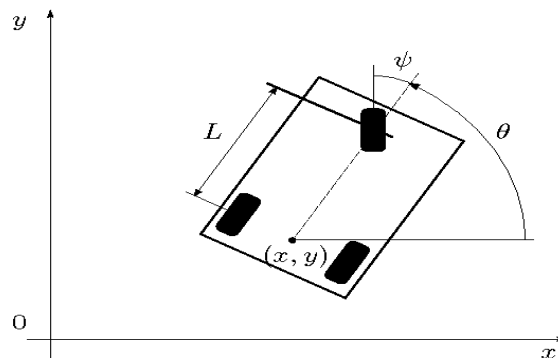


Figure I.3: Robot mobile tricycle.

c) Robot mobile de type voiture : Un robot mobile 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 mobile 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 (figure I.4) [6].



Figure I.4 : Schéma d'un robot mobile type voiture

d) Robot omnidirectionnel : Un robot mobile est dit omnidirectionnel si l'on peut agir indépendamment sur les vitesses : vitesse de translation selon les axes x et y et vitesse de rotation autour de z . D'un point de vue cinématique on montre que cela n'est pas possible avec des roues fixes ou des roues centrées orientables. On peut en revanche réaliser un robot omnidirectionnel en ayant recours à un ensemble de trois roues décentrées orientables de trois roues suédoises disposées aux sommets d'un triangle équilatéral (figure I.5.). Du point de vue de la transmission du mouvement, ceci ne va pas sans poser de problème [6].

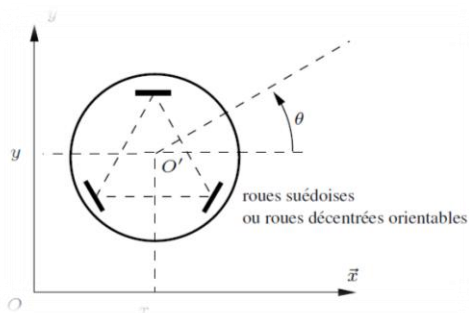


Figure I.5 : Schéma d'un robot mobile omnidirectionnel avec exemple robot Wow Wee

✓ **Comparaison des différents types :**

Nous pouvons observer dans le tableau ci-dessous un récapitulatif des avantages et des inconvénients des différents types de robots à roues [7].

Type du robot	Avantage	Inconvénient
uni cycle	stable rotation sur soi-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

Tableau I.1. Comparaison des différents types de robots mobile

I.3.2.2 Les robots mobiles à chenilles

Lorsque le terrain est accidenté, les roues perdant leurs efficacité de locomotion .ceci limite la capacité dévolution d un robot mobile équipe de se type de système de locomotion .dans ces conditions les chenilles sans plus intéressante, car elle peut augmente l'adhérence au sol et au franchir des obstacles plus importante .ce type de robot présente également des contraintes de nom holonomie.

La conception de base des robots entraînés est simple : deux chenilles, une de chaque côté du robot, agissent comme des roues géantes. Les chenilles tournent et le robot avance vers l'avant ou vers l'arrière. Pour obtenir une adhérence maximale, chaque chenille est de la même

longueur, ou un peu plus court que la longueur du robot lui-même, bien que de nombreuses variantes soient possibles.

Les robots avec des chenilles sont pratiques pour de nombreuses fonctions comme la capacité à tondre au milieu de toutes sortes d'obstacles, comme des rochers, des fossés et des nids de poule. Compte tenu de la matière des chenilles, la traction est excellente, même sur des surfaces glissantes comme la neige, le béton humide, ou un carrelage propre. Pour la plupart, la construction et l'entraînement des chenilles de manière efficace est plus difficile que les roues d'exécution.

Les chenilles présentent une grande surface de contact, cette grande surface de contact augmente la traction lors du déplacement vers l'avant ou vers l'arrière, mais elle limite également le fait de pouvoir tourner.

Les véhicules à chenilles, comme des chars, dérapent ou glissent autour d'un point tournant, par conséquent, ils ont donc un dispositif de direction par dérapage. Si les chenilles sont super-souples, et que la surface est difficile (comme un sol de cuisine), le frottement ajouté peut considérablement altérer la capacité du véhicule à tourner [8]



Figure I.6 : Exemples de robots mobiles à chenilles

I.3.2.3 Les robots mobiles marcheurs

Les robots mobiles marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est difficile et dangereux à l'homme. Leur structure dans plusieurs degrés de liberté

permet un rapprochement avec les robots manipulateurs. On distingue les robots marcheurs à deux jambes (humanoïdes), à quatre pattes (type cheval), et à six pattes (type araignée).

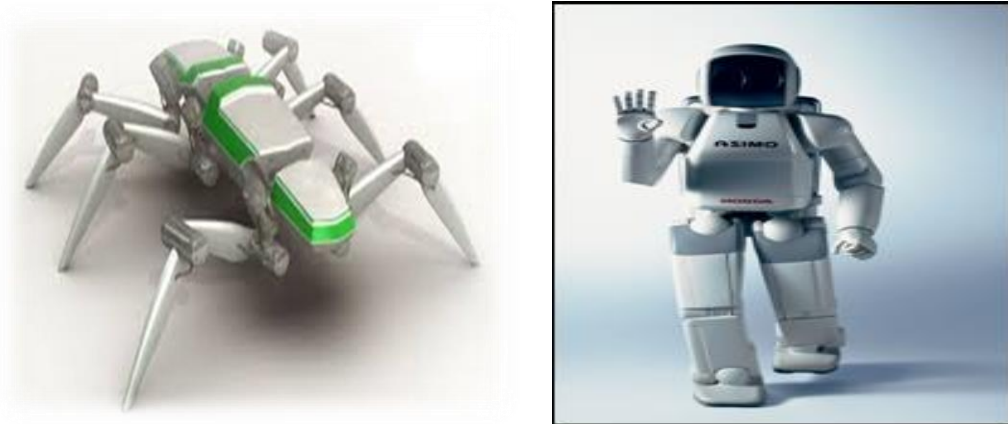


Figure I.7: Exemples de robots mobiles marcheurs.

I.4. Les moteurs

Les moteurs rencontrés dans le domaine de la robotique sont essentiellement de deux types:

I.4.1 Les moteurs à courant continu

Les moteurs à courant continu sont très utilisés dans le domaine de robotique, et possèdent un stator fixe à aimant permanent et un rotor bobine mobile.

Le stator et son aimant créent un champ magnétique fixe.

Le rotor est mis en mouvement lorsqu'une tension est appliquée sur ses bobines par des balais en frottement sur un collecteur.

Ce type de moteurs est équivalent à une résistance en série avec une inductance à l'arrêt, mais il faut ajouter une source de tension en série lorsque le moteur est en rotation. C'est sa nature inductive qui définit ses caractéristiques principales; mais la nature des balais (lames ou charbons) détériore ses performances différemment, les charbons permettant une plus grande longévité et une résistance de contact plus faible [9].



Figure I.8 : moteur à courant continu

I.4.2 Le moteur pas à pas :

Un Moteur pas à pas est à la base un actionneur de positionnement ne nécessitant pas d'asservissement, mais peut être aussi utilisé pour une commande de déplacement.



Figure I.9 : Moteur pas à pas

I.4.3 Les servomoteurs :

Un servomoteur est un système motorisé capable d'atteindre des positions prédéterminées, puis de les maintenir. La position est : dans le cas d'un moteur rotatif, une valeur d'angle et, dans le cas d'un moteur linéaire une distance.

On utilise des moteurs électriques (continu, asynchrone, brushless) aussi bien que des moteurs hydrauliques. Le démarrage et la conservation de la position prédéterminée sont commandés par un système de réglage [10].



Figure I.10 : un servomoteur.

Parmi les trois moteurs, nous avons utilisé quatre moteurs à courant continu (pour le mouvement et l'orientation).

I.5.Évitement d'obstacles :

L'évitement d'obstacles est un comportement de base présent dans quasiment tous les robots mobiles. Il est indispensable pour permettre au robot de fonctionner dans un environnement dynamique et pour gérer les écarts entre le modèle interne et le monde réel.

Les méthodes d'évitement d'obstacles doivent avoir une perception correcte de l'environnement.

✓ Les méthodes évitement d'obstacles :

Il existe plusieurs méthodes parmi lesquelles on cite :

- ✚ Méthode des champs de potentiel.
- ✚ Méthode Victor Field Histogramme.
- ✚ Méthode de la fenêtre dynamique.
- ✚ Méthode de réseaux neurones.
- ✚ Méthode de la logique floue. [11]

I.7. Conclusion

Nous avons présenté, dans ce chapitre les robots mobiles qui sont des véhicules qui remplacent l'homme dans les tâches pénibles et dangereuses actuellement plus encore dans l'avenir, les robots sont utilisés à jouer un rôle de plus en plus important dans notre vie mais ceci n'annule pas l'existence de certains problèmes pour assurer une bonne application de ces robots. Comme exemples de ces problèmes nous citons l'analyse de l'environnement, planification, navigation ...

Chapitre II

Arduino UNO

Chapitre II. Arduino UNO.

II.1. Introduction

Le système Arduino est une carte électronique à base d'un microcontrôleur et de composants minimum pour réaliser des fonctions. Elle possède une interface USB pour la programmer. C'est une plateforme open-source qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino peut être utilisé pour développer des applications matérielles industrielles légères, et peut recevoir en entrées une très grande variété de capteurs. Arduino peut aussi contrôler une grande variété d'actionneurs (lumières, moteurs ou toutes autres sorties matériels). Les projets Arduino peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (Flash, Processing ou Max MSP). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré-assemblées; le logiciel de développement open-source est téléchargeable gratuitement. [12].

II.2. Historique

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrée (Italie).

en 2005. Professeur Massimo Banzi et David Cuartielles ont éteint la recherche d'un moyen peu coûteux pour le rendre plus facile pour les étudiants en design là pour travailler avec la technologie.

Les produits existants sur le marché étaient coûteux et relativement difficile à utiliser (software et hardware). Banzi et Cuartielles décidé de faire un microcontrôleur qui pourrait être utilisé par leur art et de design les étudiants dans leurs projets.

Le nom Arduino trouve son origine dans le nom du bar dans lequel l'équipe avait l'habitude de se retrouver.

Arduino est aussi le nom d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort ».

La conception originale a été améliorée et de nouvelles versions ont été introduites. Il ya maintenant un certain nombre de versions différentes de cartes Arduino (Arduino uno, due, méga, mini)

II.3. Applications

Le système Arduino nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines, l'étendue de l'utilisation de l'Arduino est gigantesque. Pour vous donner quelques exemples, vous pouvez:

- Contrôler les appareils domestiques
- Fabriquer votre propre robot
- Communiquer avec l'ordinateur
- Télécommander un appareil mobile (modélisme)

Il existe plusieurs de cartes Arduino :

- La carte arduino LENARDO ;
- La carte arduino Mega 2560;
- La carte Arduino Mega ADK ;
- La carte Arduino DUE ;
- La carte Arduino Nano ;
- La carte Arduino Mini Pro ;
- La carte Arduino Yun ;
- La carte Arduino UNO.

Le tableau ci-dessous montre les différentes caractéristiques des cartes Arduino .

Nom	Taille (L×l×h) mm	Micro - contrôleur	Mémoire flash	SRAM	EEPROM	E/S numérique (PWM)	Entrée analogique	Alimentation
LEONARDO	70×54×16	Atmega 32u4	32	2.5	1	70(7)	12	7-12
MEGA2560	107×53×15	Atmega2560	256	8	4	54(15)	16	7-12
MEGA ADK	102×54×16	Atmega 2560	256	8	4	54(12)	16	7-12
DUE	102×53×16	AT91SAM3X8E	512	96		14(6)	12	7-12
NANO	44×20×18	Atmega168/328	16/32	1/2	0.5/1	14(6)	9	7-12
MINI PRO	33×18×10	Atmega 168	16	1	0.5	14(6)	8	3,3/12
YUN	72×53×17	Atmega 32u4	32	2.5	1	20(7)	20	5
UNO	74×54×16	Atmega 328	32	2	1	14(6)	5	7-12

Tableau II.1: Les caractéristiques des cartes Arduino

II.4.Arduino Uno

On va utiliser dans notre projet une carte Arduino uno, qui a base d'une carte microcontrôleur basé sur l'Atmega328. Il dispose de 14 broches numériques d'entrée / sortie (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un résonateur céramique de 16 MHz, d'une connexion USB, une prise d'alimentation, d'une embase ICSP et d'un bouton de réinitialisation.

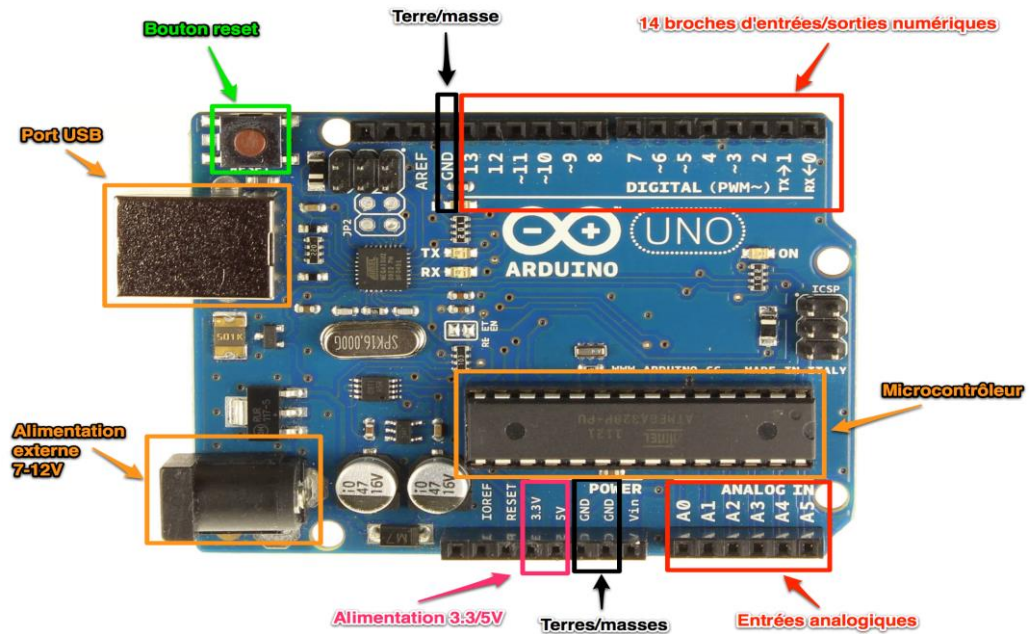


Figure II.1: Description de la carte Arduino Uno.

II.4.1. Caractéristiques de la carte Arduino Uno

Microcontrôleur	Atmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (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 de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (Atmega328) dont 0.5 KB sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	2 KB (Atmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (Atmega328)
Vitesse d'horloge	16MHz

Tableau II.2: Caractéristiques de la carte Arduino Uno

II.4.2. Les composantes de la carte Arduino Uno

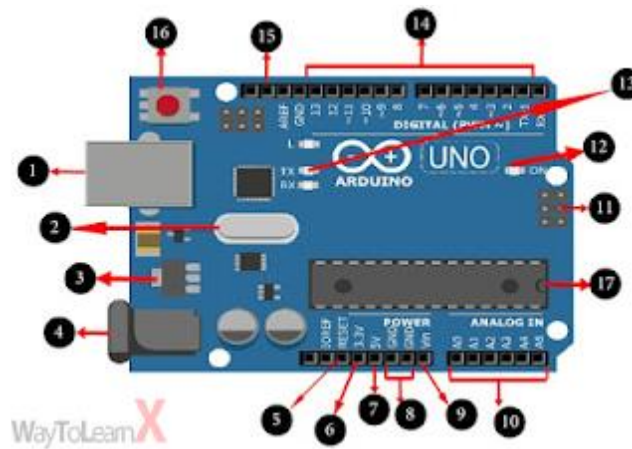


Figure II.2: les composantes de la carte arduino UNO

<p>1</p>	<p>Power USB La carte Arduino peut être alimentée à l’aide du câble USB de votre ordinateur. Tout ce que vous devez faire est de connecter le câble USB à la connexion USB.</p>
<p>2</p>	<p>Oscillateur L’oscillateur aide Arduino à traiter les problèmes de temps. Comment Arduino calcule-t-il le temps? La réponse est, en utilisant l’oscillateur. Le nombre imprimé sur l’oscillateur Arduino est de 16.000H9H. Il nous dit que la fréquence est de 16 000 000 Hertz ou 16 MHz.</p>
<p>3</p>	<p>Régulateur de tension La fonction du régulateur de tension est de contrôler la tension donnée à la carte Arduino et de stabiliser les tensions continues utilisées par le processeur et d’autres éléments.</p>
<p>4</p>	<p>Power Cartes Arduino peuvent être alimentés directement à partir de l’alimentation AC en le connectant à la prise.</p>
<p>5</p>	<p>Arduino Reset Vous pouvez réinitialiser votre carte Arduino, c’est-à-dire démarrer votre programme dès le début. Vous pouvez réinitialiser la carte UNO de deux façons. Tout d’abord, en utilisant le bouton de réinitialisation (16) sur la carte. Deuxièmement, vous pouvez connecter un</p>

	bouton de réinitialisation externe à la broche Arduino identifiée RESET (5). de sortie sur le maître du bus SPI.
6	Pin (3.3) 3.3V – Sortie alimentation 3.3 volt
7	Pin (5) <ul style="list-style-type: none"> • 5V – Sortie alimentation 5 volt • La plupart des composants utilisés avec la carte Arduino fonctionnent bien avec 3,3 volts et 5 volts.
8	Pin (GND) GND (Ground) – Il existe plusieurs broches GND sur l’Arduino, dont l’une peut être utilisée pour relier votre circuit.
9	Pin (Vin) Vin – Cette broche peut également être utilisée pour alimenter la carte Arduino à partir d’une source d’alimentation externe, comme l’alimentation secteur.
10	Pins analogique La carte Arduino UNO comporte cinq broches d’entrée analogiques A0 à A5. Ces broches peuvent lire le signal d’un capteur analogique comme le capteur d’humidité ou le capteur de température et le convertir en une valeur numérique pouvant être lue par le microprocesseur.
11	Pin ICSP Principalement, ICSP (12) est un AVR, un petit en-tête de programmation pour Arduino composé de MOSI, MISO, SCK, RESET, VCC et GND. Il est souvent appelé un SPI (Serial Peripheral Interface), qui pourrait être considéré comme une « expansion » de la sortie. En fait, vous débrouillez le périphérique
12	Indicateur d’alimentation LED Cette LED s’allume lorsque vous branchez votre Arduino dans une source d’alimentation pour indiquer que votre carte est sous tension correctement. Si cette lumière ne s’allume pas, il y a quelque chose qui ne va pas avec la connexion.
13	LEDs TX et RX Sur votre carte, vous trouverez deux étiquettes: TX (transmission) et RX (réception). Ils apparaissent en deux endroits sur la carte Arduino. Tout d’abord, au niveau des broches

	<p>numériques 0 et 1, pour indiquer les broches responsables de la communication série. Deuxièmement, la led TX et RX (14). La led TX clignote avec une vitesse différente lors de l'envoi des données en série. La vitesse de clignotement dépend du débit en bauds utilisé par le tableau. RX clignote pendant la réception.</p>
14	<p>E/S numérique</p> <p>La carte Arduino UNO comporte 14 broches d'E/S numériques (14) (dont 6 fournissent une sortie PWM (Modulation de largeur d'impulsion). Ces broches peuvent être configurées pour fonctionner comme broches numériques d'entrée pour lire les valeurs logiques (0 ou 1) ou comme numérique Des broches de sortie pour piloter différents modules comme les LED, les relais, etc. Les broches étiquetées «~» peuvent être utilisées pour générer PWM. CI. Pour plus de détails sur la construction et les fonctions CI, vous pouvez consulter la fiche technique.</p>
15	<p>AREF</p> <p>AREF signifie Référence analogique. Il est parfois utilisé pour régler une tension de référence externe (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.</p>
16	<p>Arduino Reset</p> <p>Vous pouvez réinitialiser votre carte Arduino, c'est-à-dire démarrer votre programme dès le début. Vous pouvez réinitialiser la carte UNO de deux façons. Tout d'abord, en utilisant le bouton de réinitialisation (16) sur la carte. Deuxièmement, vous pouvez connecter un bouton de réinitialisation externe à la broche Arduino identifiée RESET (5).</p>
17	<p>Microcontrôleur</p> <p>Chaque carte Arduino possède son propre microcontrôleur. Vous pouvez l'assumer comme cerveau de votre carte Arduino. Le CI (circuit intégré) principal sur l'Arduino est légèrement différent de la carte. Les microcontrôleurs sont habituellement de la société ATMEL. Vous devez savoir quel type du CI sur votre carte avant de charger un nouveau programme de l'IDE Arduino. Cette information est disponible en haut du CI. Pour plus de détails sur la construction et. les fonctions CI, vous pouvez consulter la fiche technique.</p>

II.5. Présentation de l'Espace de développement Intégré (EDI) Arduino

II.5.1.. Description de l'interface

Le logiciel Arduino a pour fonctions principales :

- ✚ de pouvoir écrire et compiler des programmes pour la carte Arduino
- ✚ de se connecter avec la carte Arduino pour y transférer les programmes
- ✚ de communiquer avec la carte Arduino

Cet espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino comporte :

- ✚ une **BARRE DE MENUS** comme pour tout logiciel une interface graphique (GUI),

- ✚ une **BARRE DE BOUTONS** qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation,

- ✚ un **EDITEUR** (à coloration syntaxique) pour écrire le code de vos programme, avec onglets de navigation,

- ✚ une **ZONE DE MESSAGES** qui affiche indique l'état des actions en cours,

- ✚ une **CONSOLE TEXTE** qui affiche les messages concernant le résultat de la compilation du programme

- ✚ un **TERMINAL SERIE** (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une

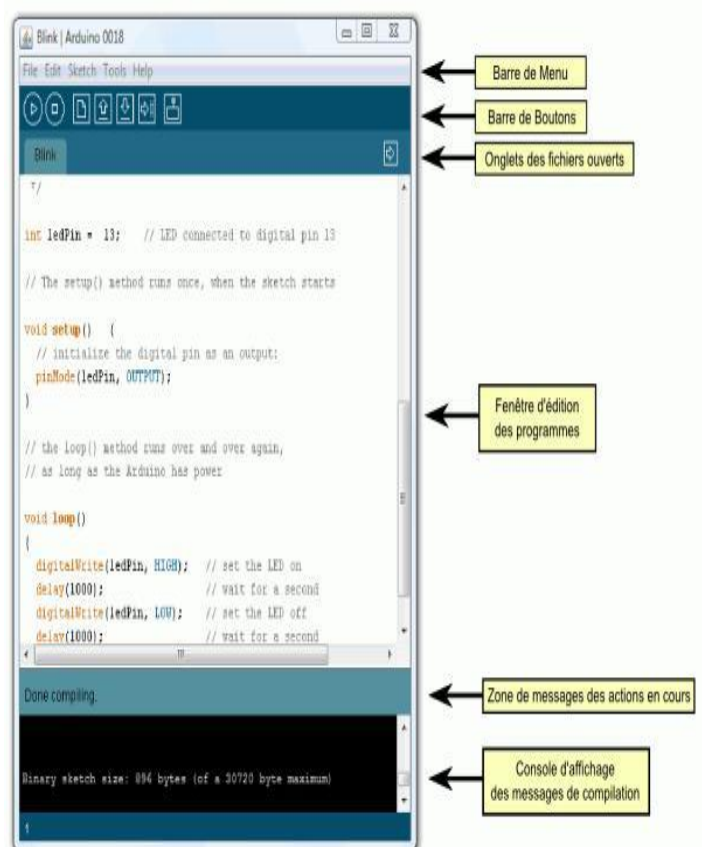


Figure II.3: présentation des éléments de L'ARDUINO software

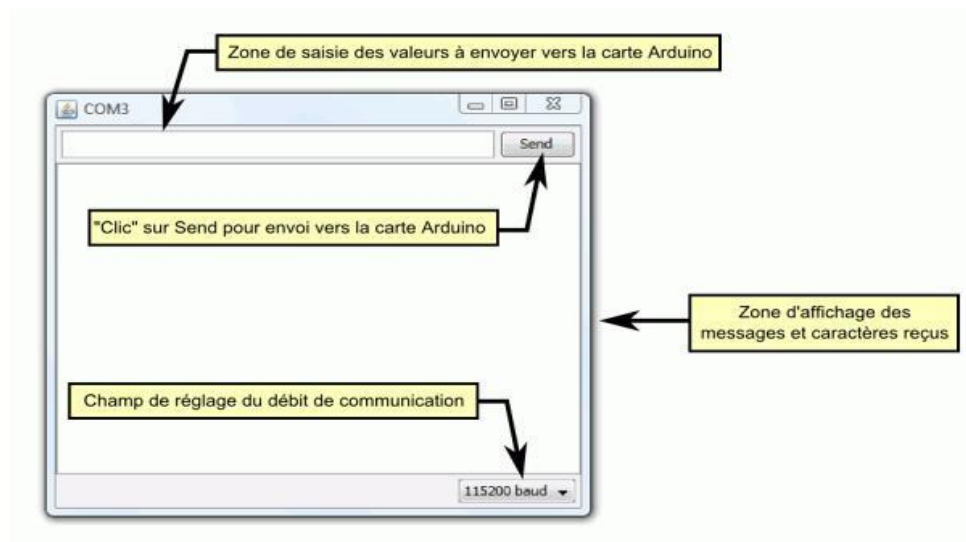


Figure II.4: module TERMINAL SERIE

II.5.2. Description de la structure d'un programme

II.5.2.1 Description générale des parties :

Un programme utilisateur 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 code, comme mors d'une programmation classique. Cette structure se décompose en trois parties :

- ✚ Description des constantes et variables du programme
- ✚ Fonction principale : configuration des entrées/sorties et éléments à configurer (cette partie ne sera exécutée qu'une seule fois) dans le partie *VOID SETUP()*
- ✚ Fonction boucle : description du fonctionnement général du programme (gestion des interactions entre les entrées/sorties) dans la partie *VOID LOOP()*



Figure II.5: Fenêtre graphique de L'EDI

II.5.2.2 Description détaillée des parties

a. Définition des variables et constantes : Dans cette partie, on déclare des éléments utilisés tout au long du programme : les constantes (statiques) et les variables (dynamiques). Ce sont des emplacements mémoire utilisés pour stocker des données (des valeurs) utilisables dans la suite du programme.

Variable : Une variable peut aussi bien représenter *renvoyé étape de calcul* numéro ‘physique’ de ces entrées ou sorties. Une “variable” n’est donc pas exclusivement un paramètre variant dans le programme. On la déclare de la façon suivante :

TYPE_DE_LA_DONNEE NOM_DE_LA_DONNEE

Exemple : int led

Constante : Une constante est une variable dont la valeur est inchangeable lors de l'exécution d'un programme. On la déclare de la façon suivante :

CONST TYPE_DE_LA_DONNEE NOM_DE_LA_DONNEE

Exemple : const int led

Les différents types de données: En programmation informatique, un type de donnée, ou simplement type, définit les valeurs que peut prendre une donnée, ainsi que les opérateurs qui peuvent lui être appliqués.

NOM DU TYPE	VALEUR MIN/MAX	TAILLE EN MEMOIRE
VALEURS BINAIRES		
boolean	0/1	1 octet
VALEURS NUMERIQUES ENTIERES SIGNEES		
int	-32 768 / +32 767	2 octets
long	-2 147 483 648 / +2 147 483 647	4 octets
VALEURS NUMERIQUES ENTIERES NON SIGNEES		
byte	0 / +255	1 octet
unsigned int	0 / +65535	2 octets
word	0 / +65535	2 octets
unsigned long	0 / +4 294 967 295	4 octets
VALEURS NUMERIQUES A VIRGULE		
float	-3.4028235E+38 / +3.4028235E+38	4 octets
double	-3.4028235E+38 / +3.4028235E+38	4 octets
CARACTERES		
char	-128 / +127 (ASCII)	1 octet

Tableau II.3: Les différents types de données

b. Fonction principale : void setup()

Cette fonction n'est exécutée qu'une seule fois au démarrage du programme. Elle permet la configuration des entrées et sorties de la carte. Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées qu'en sorties. Ici on a configuré LED_Pin_13 en sortie. pinMode (nom, état) est une des quatre fonctions relatives aux entrées et sorties numériques que nous verrons plus bas.

```
void setup()
{
ici se trouve la configuration des entrées et des sorties
}
```

c. Fonction boucle : void loop()

Cette fonction `loop()` (boucle en anglais) fait exactement ce que son nom suggère et s'exécute en boucle sans fin, permettant à votre programme de s'exécuter. Dans cette boucle, on définit les opérations.

La fonction `loop()` est obligatoire, même vide, dans tout programme.

```
void loop()
```

```
{
```

```
ici se trouve la description générale du programme en boucle
```

```
}
```

II.6. Conclusion

Il y a de nombreux microcontrôleurs et de nombreuses plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, et beaucoup d'autres qui offrent des fonctionnalités comparables.

Tous ces outils prennent en charge les détails compliqués de la programmation des microcontrôleurs et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs, tout en offrant plusieurs avantages pour les enseignants, les étudiants et les amateurs intéressés par les autres systèmes.

Chapitre III

**Simulation et Réalisation
d'un Robot mobile**

Chapitre III : Simulation et Réalisation d'un Robot mobile

III. 1 Introduction

En se basant sur l'étude théorique, nous allons présenter en détails dans ce chapitre le matériel utilisé et les caractéristiques des composants du système pour aboutir à la réalisation du projet.

L'idée principale de ce projet été de réaliser un robot mobile type voiture puis de le guider avec différentes méthodes : à distance, par internet et autonome en l'obligeant à suivre une trajectoire et d'éviter des obstacles.

La commande de ce véhicule sera assurée principalement par Arduino. Pour cela on a partagé ce chapitre en deux parties :

- ✚ La partie simulation qui est effectuée à l'aide de logiciel Proteus et Arduino IDE.
- ✚ La partie réalisation de robot qui se concentre sur l'Arduino, le driver des moteurs et l'ultrason.

III.2. Cahier de charge

Le but de ce projet consiste à concevoir un robot mobile capable d'être commander par un ordinateur, téléphone portable et télécommande. La figure III.1. Présente les quatre méthodes possibles pour commander notre robot mobile :

- ✚ Commande par Arduino seulement (Autonome).
- ✚ Commande par PC via le port USB.
- ✚ Commande par mobile via Bluetooth.
- ✚ Commande par voix vocale.

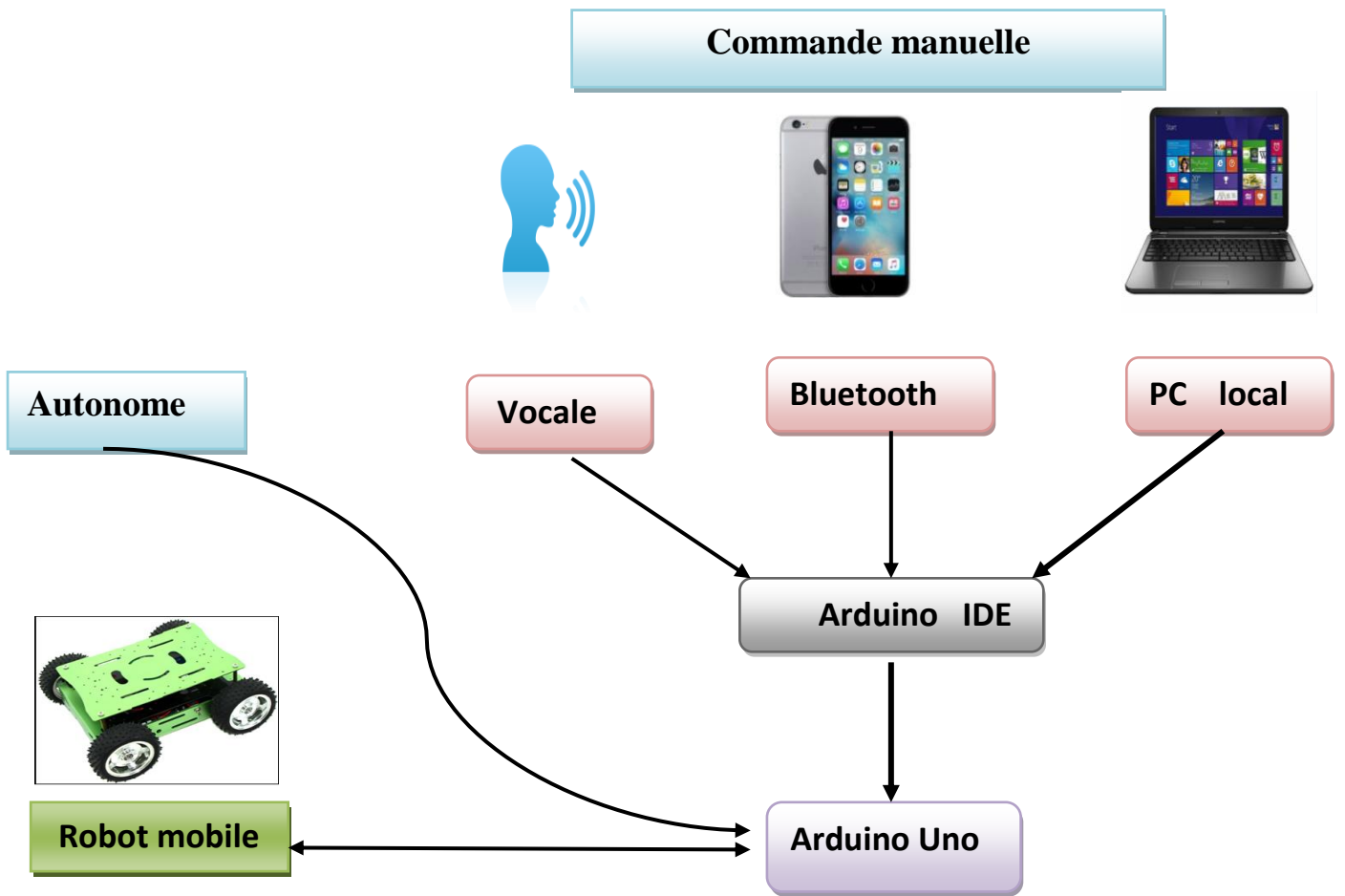


Figure III.1: Méthodes de commande de robot mobile

III.3. Principaux éléments constituant le système

On va présenter dans cette partie le matériel que nous avons utilisé pour réaliser ce projet

Chapitre III : Simulation et Réalisation d'un Robot mobile

(Figure III.2) .

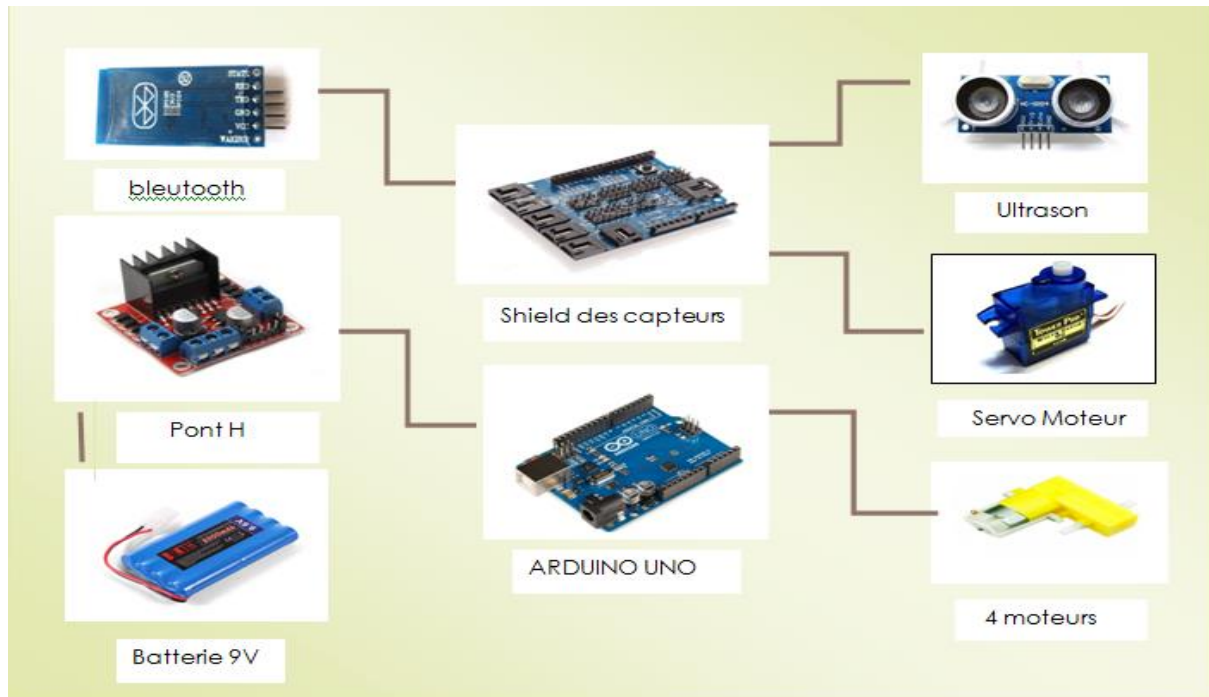


Figure III.2 : Matériel utilisé

III.3.1. La carte ARDUINO UNO

Arduino Uno est parmi les meilleurs circuits d'interfaçage pour commencer avec les systèmes de commande en électronique. Il est conseillé alors d'utiliser l'Uno car il est largement utilisé et aussi bien documenté. Pour une compréhension approfondie des fonctionnalités d'Arduino Uno, voir le chapitre précédent.

III.3.2. Le driver pour moteur à courant continu L293B

L293B est un pilote de moteur typique ou pilote de moteur à courant continu lui permettant la rotation dans les deux sens. L293B est un circuit intégré à 16 broches qui permet de contrôler un ensemble de deux moteurs à courant continu simultanément dans les deux directions. Cela signifie qu'on peut contrôler deux moteurs à courant continu avec un seul L293B. En raison de sa taille, il est très utilisé pour les applications robotiques pour la commande de moteurs à courant continu. Ci-après est le schéma de brochage d'un contrôleur de moteur L293B. Le l293B peut conduire silencieusement petits et gros moteurs ainsi il est

Chapitre III : Simulation et Réalisation d'un Robot mobile

conçu pour fournir des courants d'attaque bidirectionnelle jusqu'à 1 A à des tensions de 4,5 V à 36 V [13].

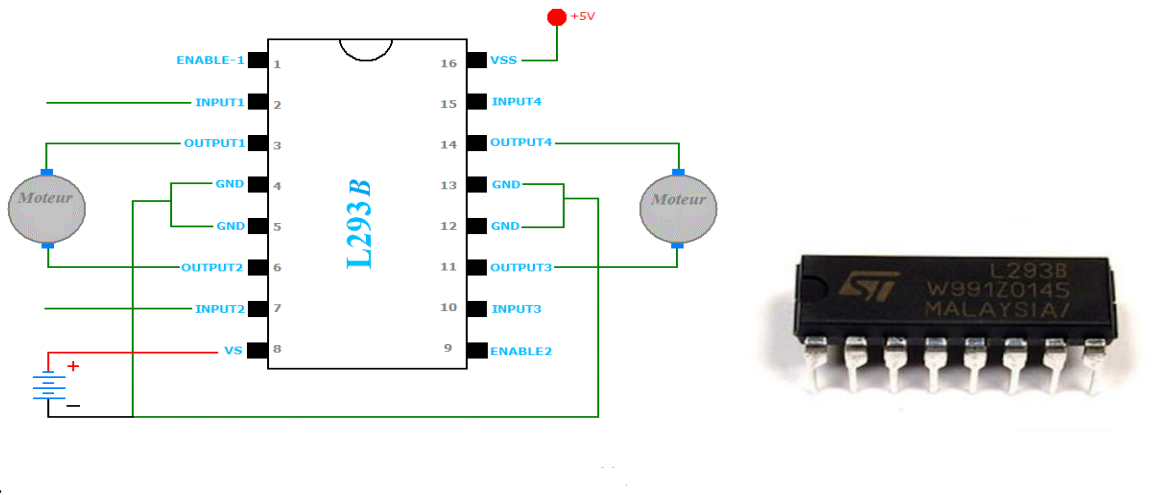


Figure III.3: Circuit L293B

III.3.3. Double ponts H Contrôleurs de moteur, le L298N

Ce circuit nous permet de contrôler la vitesse et la direction de deux moteurs à courant continu, ou contrôler un moteur pas à pas bipolaire facilement. Le module de pont en H L298N, qui possède le même bouchage du L293B, peut être utilisé avec des moteurs ayant une tension comprise entre 5 et 35 V en courant continu.

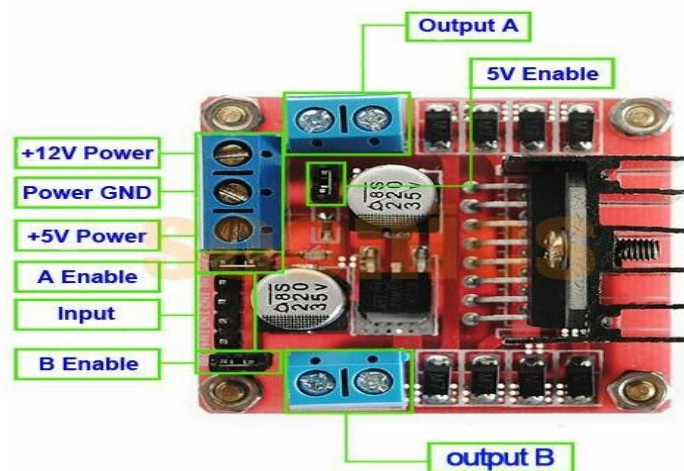


Figure: III.4 : Le brochage du Module L298N double ponts en H

III.3.4. Capteur à ultrasons HC-SR04

Le capteur à ultrasons HC-SR04 utilise un sonar pour déterminer la distance à un objet comme les chauves-souris ou les dauphins faire. Il offre une excellente plage de détection sans-contact avec une grande précision et des lectures stables, dans un package facile à utiliser, avec une portée de 2cm à 500 cm ou de 1 à 16 pieds.

Son fonctionnement n'est pas affecté par la lumière du soleil ou d'une matière noire. Il est livré avec émetteur d'ultrasons de 40kHz et un module récepteur. La distance est donnée par la formule simple suivante :

$$x = \frac{\text{Temps de retard entre impulsions émises et impulsions reçues} \times 340 \cdot 10^2}{2} \text{ (cm)}$$

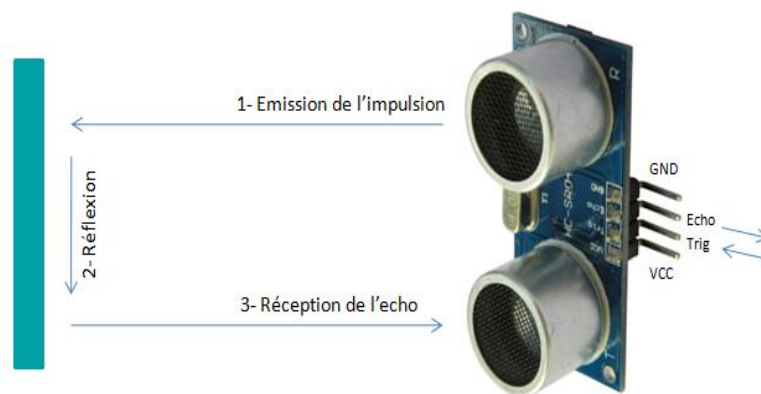


Figure III.5 : Capteur ultrason HC-SR 04

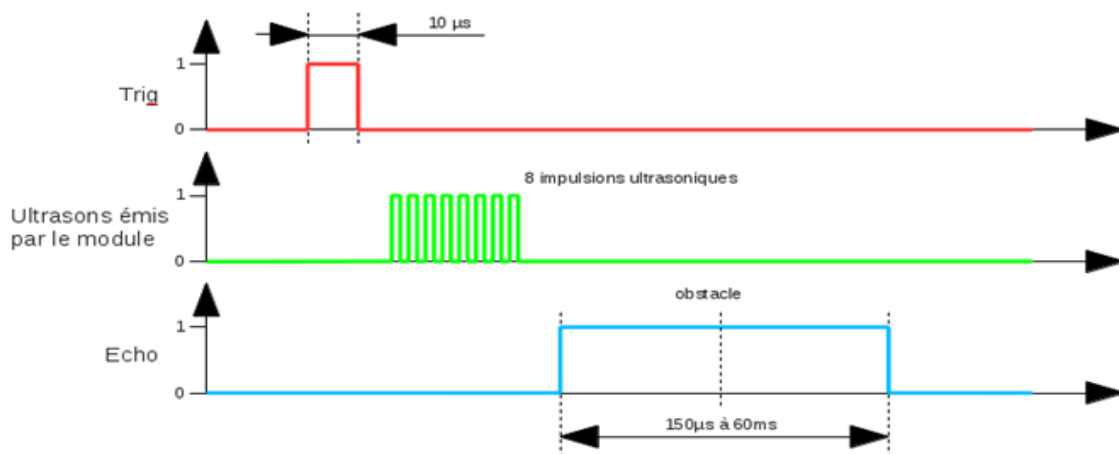


Figure III.6: Chronogramme du capteur ultrason HC-SR04

III.3.5. Mini moteur motoréducteur 6V

Ce micro motoréducteur (moteur + engrenage) 6V, à 180 tours/min à renvoi d'axe dans lequel l'axe de l'arbre de sortie s'étend de manière parallèle à l'axe des moyens moteurs. Ce moteur, de petite taille, est simple à installer. Il convient parfaitement à la construction d'une plateforme robot mobile.

➤ Caractéristiques :

- Référence: DIR00167
- Rapport de réduction : 1:120
- Vitesse hors-charge (3V) : 100 tours/min
- Vitesse hors-charge (6V) : 200 tours/min
- Courant hors-charge (3V) : 60 mA.
- Courant hors-charge (6V) : 71 mA.
- Courant d'arrêt (3V) : 260 mA.
- Courant d'arrêt (6V) : 470 mA.
- Couple (3V) : 1.2 Kg cm.
- Couple (6V) : 1.92 Kg cm.
- Dimension : 55 x 48.3 x 23 mm.
- Poids : 45g.
- Prix : 900DN. [17]

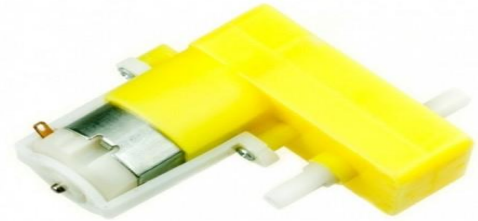


Figure III.7 : Mini moteur. Motoréducteur

III.3.6. Roues

Les roues sont de la marque Tamiya, ils conviennent bien aux moteurs que nous avons utilisés.

Caractéristique

- Diamètre extérieur : 65 mm
- Largeur de la roue : 26 mm
- Moyeu : 6 mm



Figure III.8 : la roue Tamiya

III.3.7. Le servomoteur de l'ultrason HC-SR04

Les servomoteurs sont des actionneurs très utilisés dans l'industrie, ils ont comme caractéristique principale leur « couple », c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des « membres » lourds comme déplacer un bras qui porte une charge.

Pour la robotique de loisirs, les servomoteurs ont en général peu de couple et sont de taille réduite, bien adaptés à un encombrement minimal et à une énergie disponible limitée. Pour faire bouger l'axe de sortie il faut lui envoyer une impulsion. C'est la largeur de cette impulsion qui détermine l'angle de rotation de l'axe de sortie.

La durée de l'impulsion peut être comprise entre 0.5 et 2.3ms. La largeur de l'impulsion de 1.5ms correspond au centre (Figure III.10). Cette impulsion doit être répétée toutes les 20ms. Le servomoteur se compose de:

- un moteur à courant continu.
- Un circuit électronique
- un potentiomètre.



Figure III.9 : Servomoteur.

Un système réducteur de vitesse à roues dentées [18].

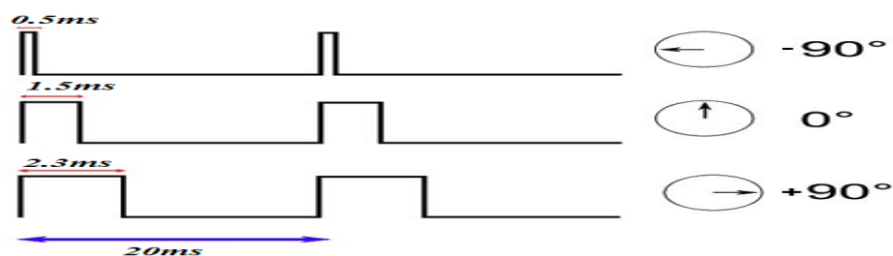


Figure III.10 : Impulsion de commande d'un Servomoteur.

III.3.8. Bluetooth Transceive Module HC-06

Ce module Bluetooth peut facilement obtenir une transmission de données sans fil de série. Sa fréquence de fonctionnement est parmi la bande de fréquence 2.4GHz ISM le plus populaire (à savoir industrielle, scientifique et médicale).

Chapitre III : Simulation et Réalisation d'un Robot mobile

Il adopte la norme Bluetooth 2.0 + EDR. Dans Bluetooth 2.0, le signal de temps d'émission des différents dispositifs se trouve à un intervalle de 0,5 secondes de sorte que la charge de travail de la puce Bluetooth peut être considérablement réduite et plus de temps de sommeil peut être sauvegardé pour Bluetooth.

Ce module est configuré avec une interface série, qui est facile à utiliser et simplifie l'ensemble du cycle de conception / développement.

Specification:

Protocol Bluetooth: norme Bluetooth 2.0+ EDR.

Protocole USB: USB v1.1 / 2.0

Fréquence de fonctionnement: 2,4 GHz bande de fréquence ISM

Puissance d'émission: $\leq 4\text{dBm}$,

Vitesse de transmission: 2.1Mbps (Max)

Température de fonctionnement: -20 à 55 °C.

Taille: 36.5 * 16mm.

Poids : 4g .



Figure III.11: Module Bluetooth HC-06.

III.3.9. Arduino capteur Shield V4.0

Les Shields Arduino sont des cartes qui se branchent sans soudure aux cartes Arduino ou à d'autres Shields Arduino pour augmenter leurs capacités. Avec le shield des capteurs, vous pouvez connecter les cartes compatibles Arduino facilement avec diverses interfaces I2C, Uart, numériques et analogiques. Des modules de capteurs, potentiomètres, relais, des servomoteurs peuvent être facilement connectés à Arduino Capteur Shield.

Ce module est conçu pour connecter aisément les capteurs et actionneurs. Ce module capteur vous permet de réaliser vos expériences électroniques sans soudure ni utilisation d'un breadboard . Il vous suffit simplement d'utiliser les câbles afin d'alimenter et d'utiliser les capteurs et actionneurs.

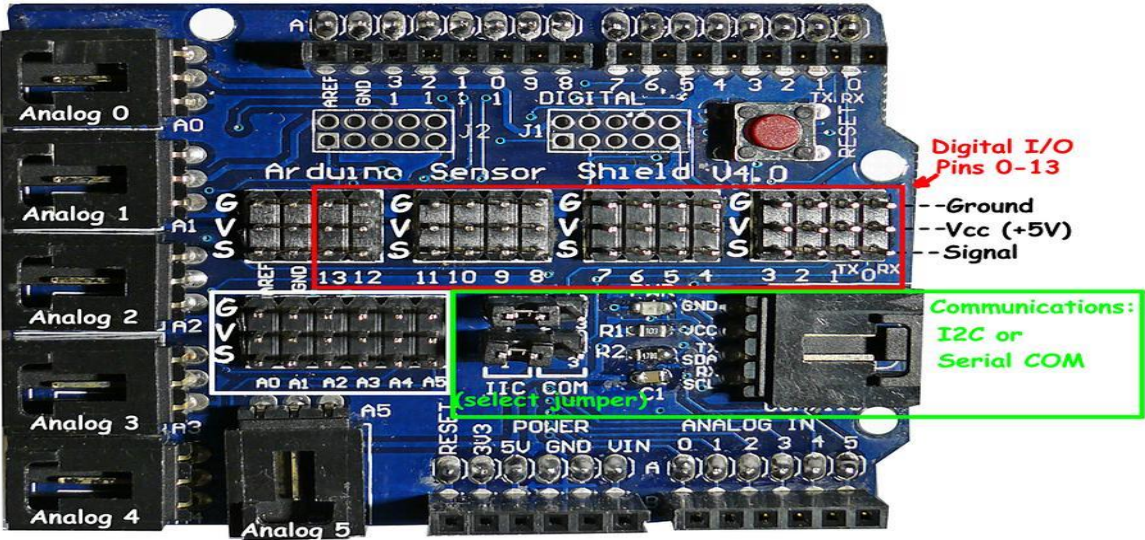


Figure III.12 : Module Capteur shield d'Arduino [32].



Figure III.13 : Photo du projet réalisé

III.4. Partie logicielle

Dans cette partie on va voir les logiciels de simulation utilisés dans notre projet.

III.4.1. Le logiciel proteus

Proteus est un logiciel destinée à la simulation des circuits électroniques analogiques et numériques, développé par la société Labcenter Electroniques. Les logiciels incluses dans Proteus permettent la CAO (Conception Assistée par Ordinateur) dans le domaine électronique, on parle de ISIS et ARES.

Outre la popularité de l'outil, Proteus possède d'autres avantages

- + Pack contenant des logiciels facile et rapide à comprendre et à utiliser
- + Le support technique est performant
- + L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet

III.4.2. ISIS

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

III.4.3. La simulation du projet

Pour commencer la simulation il faut d'abord ajouter les bibliothèques nécessaires à la simulation comme la bibliothèque Arduino, la bibliothèque Ultrason et la bibliothèque Bluetooth au logiciel Proteus.

On charge le programme 'HEX' crée par le logiciel Arduino IDE dans l'Arduino. La variation de la distance dans Proteus se fait par un pont diviseur de tension.

Chapitre III : Simulation et Réalisation d'un Robot mobile

III.4.3.1. La simulation de robot autonome

Dans cette partie on va charger l'Arduino avec un programme IDE qui assure l'autonomie du robot mobile. On va varier la distance est on va voir le changement qui passe dans les moteurs est le servomoteur (Figure III.14).

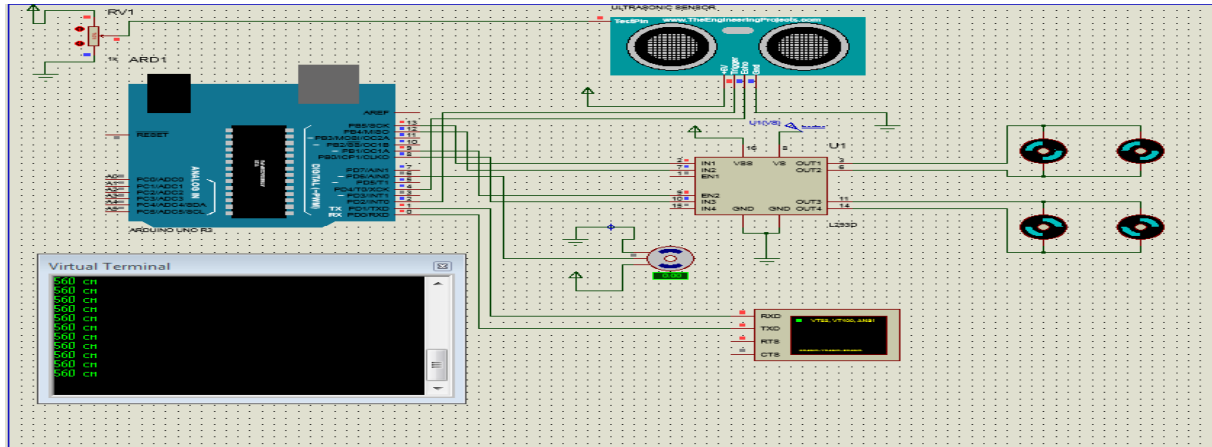


Figure III.14: la simulation d'un robot mobile AUTONOME par le logiciel ISIS.

III.4.3.2. La simulation du mouvement de robot commandé par Bluetooth

Dans ce cas on charge l'Arduino avec un programme IDE qui assure la commande du robot mobile. On va commander par des ordres (un caractère est envoyé pour chaque mouvement : par exemple on envoie 'G' pour indiquer 'tourne à gauche').

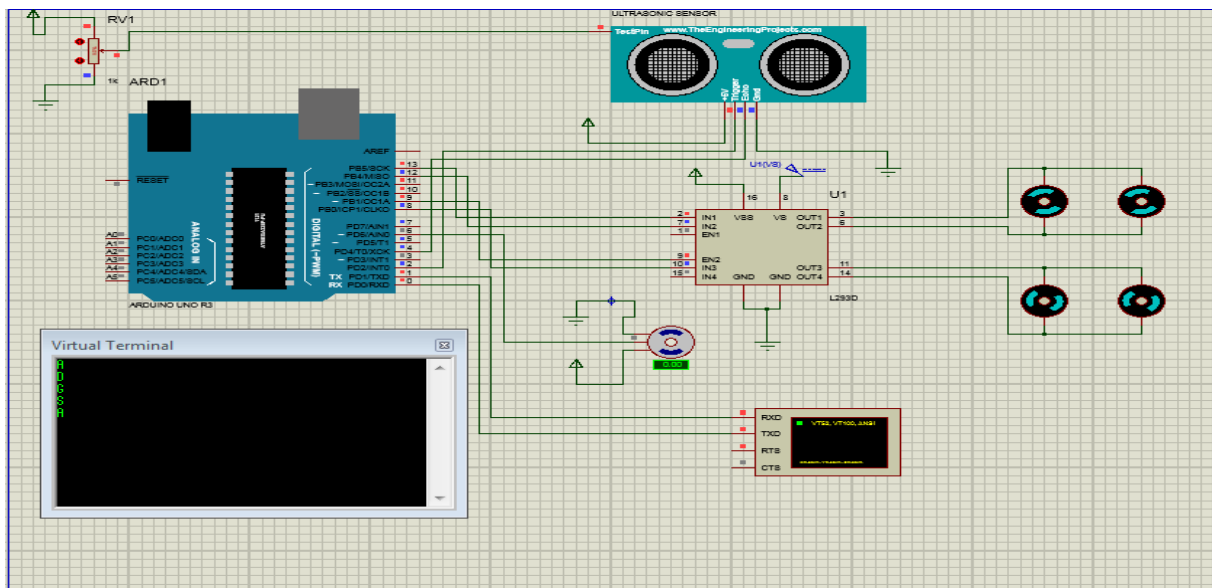


Figure III.15: la commande du robot par Bluetooth en simulation

Chapitre III : Simulation et Réalisation d'un Robot mobile

III.4.3.3. L'Arduino IDE

L'environnement de programmation Arduino (IDE) est une application Java, libre et multiplateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers la liaison série (RS-232, Bluetooth ou USB selon le module).

Il est également possible de se passer de l'interface Arduino, et de compiler les programmes via l'interface en ligne de commande.

Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties.

La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++.

Un programme Arduino comporte trois parties comme il est montré sur la figure (III.15) pour un exemple d'un module ultrason.

Dans chaque partie du programme ont utilise différentes instructions issues de la syntaxe du langage Arduino.

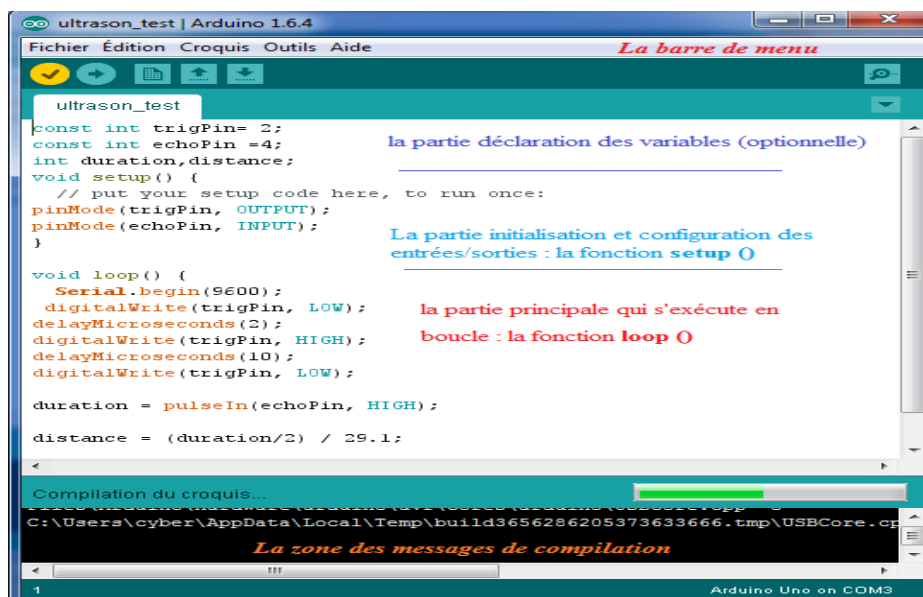


Figure III.16 : Interface de logiciel IDE avec explication de différentes parties.

III.4.3.4. L'organigramme de commande du robot mobile autonome par IDE

On présente dans se organigramme de programme IDE, qui permet de commander le robot mobile d'une manière autonome(le programme IDE et motionner dans l'annexe A).

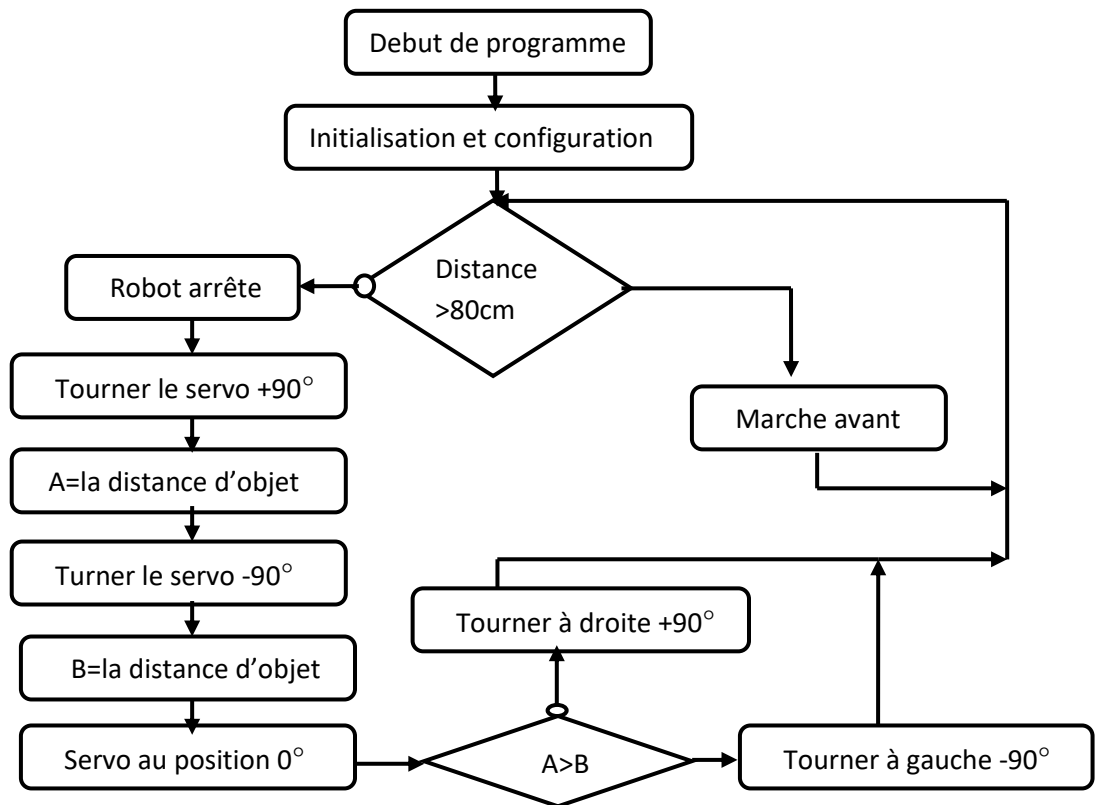


Figure III.17 : Organigramme de commande D'un robot mobile autonome par IDE.

III.4.3.5. L'organigramme de commande du robot mobile par Bluetooth

Cette technique permet d'établir une communication Bluetooth et de commander les déplacements d'un robot comprenant un circuit Arduino. On va présenter dans cet organigramme le programme IDE, qui permet de commander le robot par Bluetooth (le programme IDE et motionner dans l'annexe A).

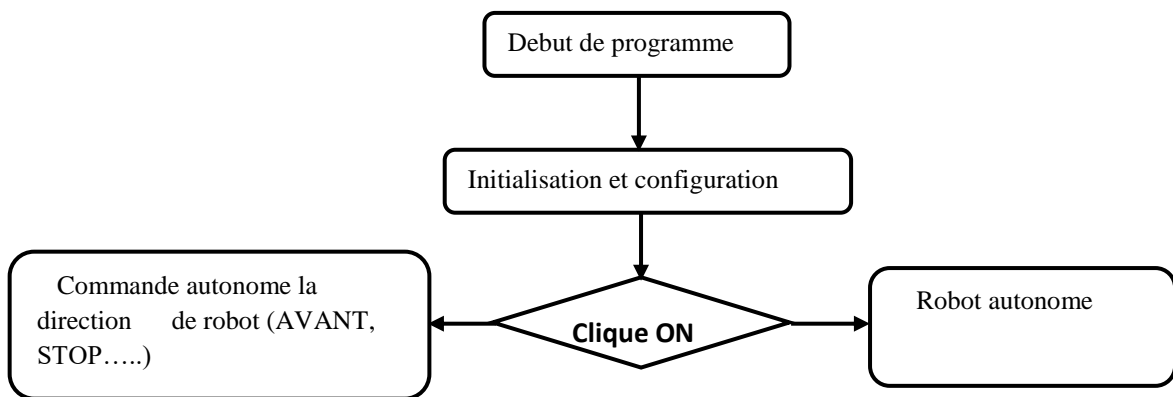


Figure III.18 : Organigramme de commande D'un robot mobile par Bluetooth

III.5. Réalisation de robot mobile

Nous avons mentionné le matériel que nous avons utilisé, maintenant ce matériel est monté pour construire notre robot mobile type voiture. On place les quatre moteurs avec leurs roues. Nous avons utilisé quatre moteurs pour commander le robot mobile. Deux moteurs de chaque côté rattaché parallèlement (de point de vue électrique) avec le pont en h L298N. Le mécanisme de notre robot est illustré dans la figure III.19.

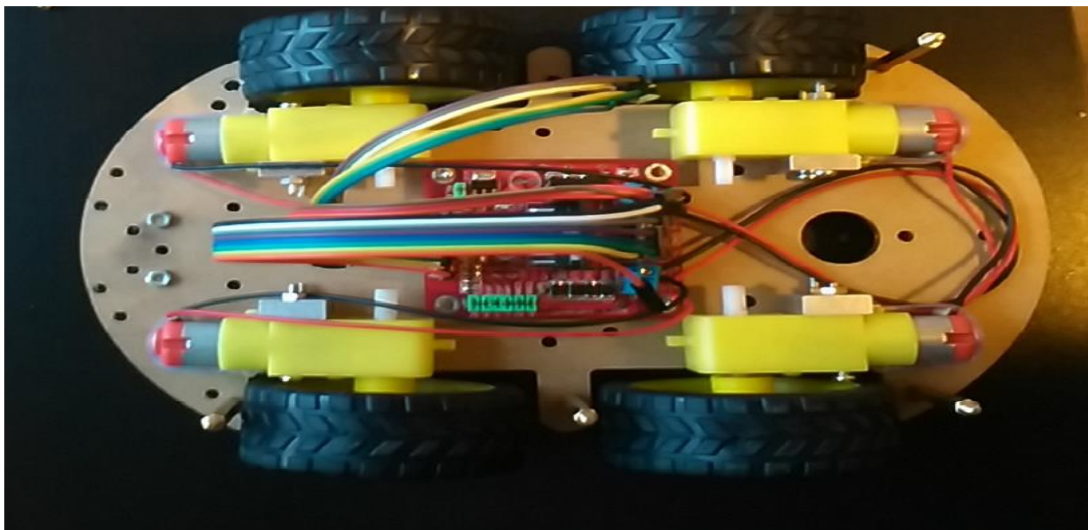


Figure III.19 : placement des moteurs et de module L298N sur châssis

Chapitre III : Simulation et Réalisation d'un Robot mobile

Pour faire marcher le robot avant/arrière on alimente les quatre moteurs en même temps (la polarisation du moteur détermine le sens de rotation). Pour la rotation gauche/droite on alimente deux moteurs de même côté seulement.

- ✚ Le câblage avec l'Arduino
- ✚ Câblage du pont en H et moteurs

On relie les pins 7,8 et 9 du pont en H qui sont responsables de commander la direction des deux moteurs à gauche avec les pins 5,6 et 7 de l'Arduino, et les pins 10,11 et 12 de pont en H avec les pins 8,9 et 10 d'Arduino pour commander la direction des moteurs à droite comme il est montré dans la figure III.20.

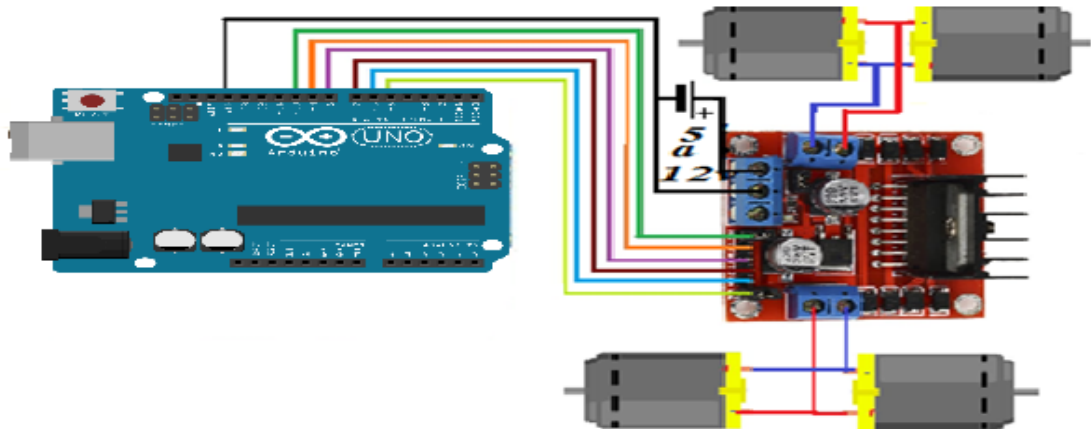


Figure III.20: Arduino avec le module L298 et 4 moteur.

Câblage de l'ultrason : On place le module ultrason avec son support sur le servomoteur figure III.21. Le servomoteur permet de tourner l'ultrason à droite et à gauche pour scanner l'environnement et par conséquent de déterminer approximativement la forme de l'obstacle.

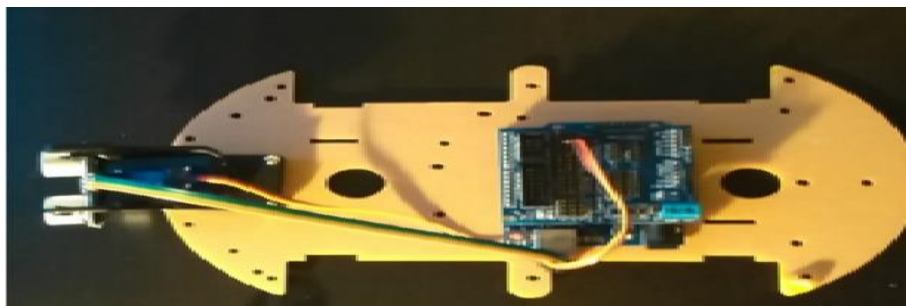


Figure III.21: placement de l'Arduino et le module ultrason sur le robot.

Chapitre III : Simulation et Réalisation d'un Robot mobile

On relie la pin de signal de servomoteur avec le pin 13 d'Arduino, et les pin 'écho' et 'trig' d'ultrason avec les pin 11 et 12 (Figure III.22).

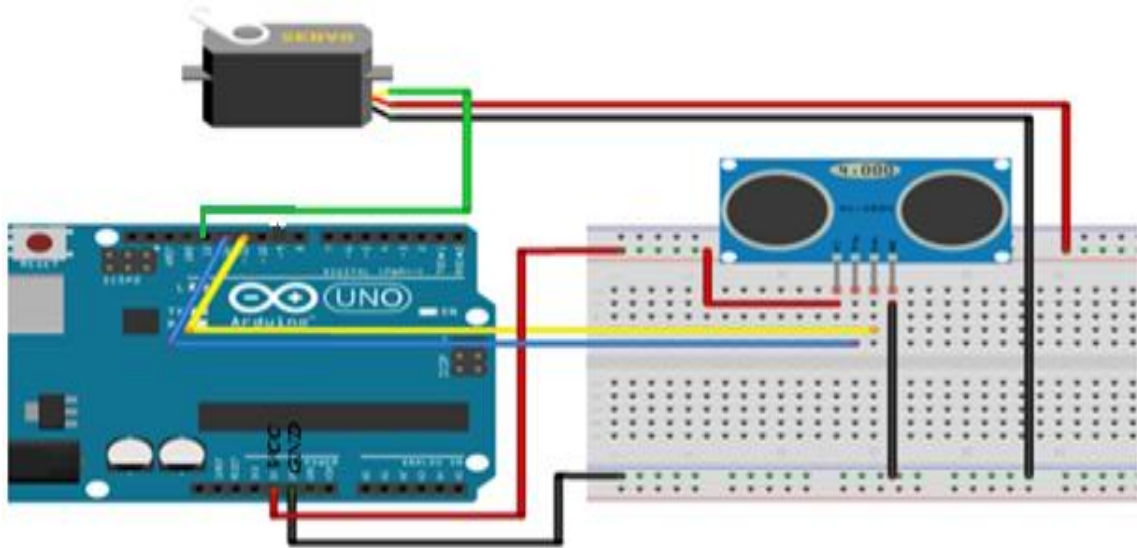


Figure III.22 : Arduino avec l'ultrason et le servomoteur.

- ✓ **Câblage de Bluetooth :** Le Bluetooth HC-06 contient 4 broches, deux pour l'alimentation (GND, VCC) et broche de réception (RX) attaché avec le pin de transmission d'Arduino (TX), et le broche de transmission attaché avec le pin de réception d'Arduino.

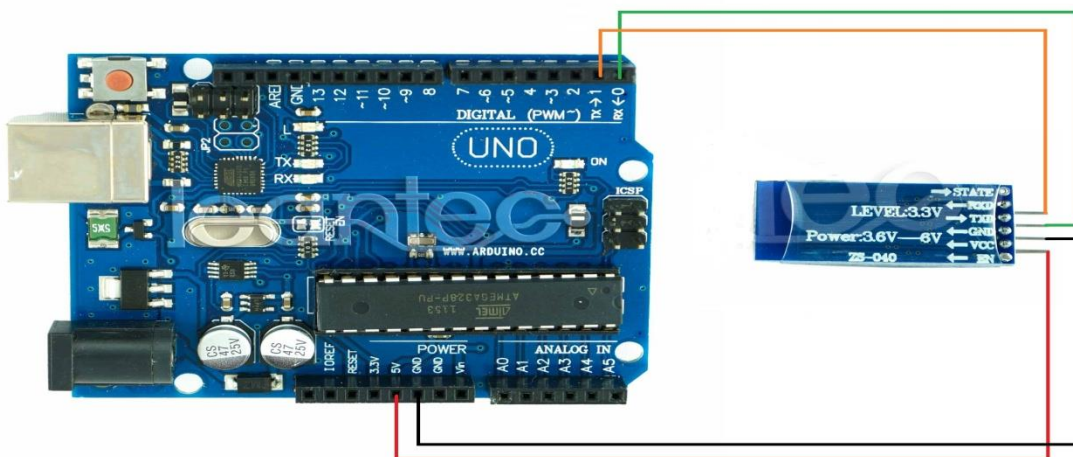


Figure III.23 : le brochage de Bluetooth avec Arduino

- ✓ **L'alimentation :** Nous avons alimenté le pont H avec une pile BAKTH haute Capacité, 9.6V avec ampérage de 2Ah (Figure III.23). Cela permet d'assurer un fonctionnement

Chapitre III : Simulation et Réalisation d'un Robot mobile

efficace de longue durée de notre robot surtout en cas d'utilisation dans des tâches qui demande de puissance. Pour l'alimentation d'Arduino on a utilisé 6 piles de 1.2v.



Figure III.24 : pile BAKTH

III.6 Présentation d'App inventor



APP INVENTOR est un IDE (environnement de développement intégré) qui permet la création d'applications destinées à des systèmes équipés de plateformes Android tels que les téléphones (tactiles ou non) et les tablettes (ACER, SAMSUNG ...).

A l'origine le projet APP INVENTOR a été créé par le Google Labs puis abandonné. Aujourd'hui ce concept est repris par le MIT (Massachusetts Institute of Technology). Le téléchargement et l'utilisation de APP Inventor nécessite un compte google (gmail).

Cet environnement de programmation permet une programmation graphique, basée sur l'assemblage de blocs (langage Scratch). Des connaissances en programmation orientée objet sont toutefois nécessaires. Cependant, on pourra s'affranchir complètement de la connaissance des noms des propriétés et méthodes liées aux objets. En effet des blocs de propriétés, méthodes et événements seront directement proposés dès la création d'un objet.

L'IDE est formé de deux composantes logicielles :

Un site Web permettant de créer l'interface homme machine (aspect graphique) de la future application ; Un programme résidant sur l'ordinateur, qui permet par association des blocs liés aux objets précédemment définis dans l'espace web, de créer le comportement de l'application. Il permet également le test de l'application sur un émulateur ou le téléchargement vers le média désiré (tablette, téléphone...).

III.6.1 Créé un projet :

Depuis un navigateur internet ouvrir le compte google.

Aller sur le site du MIT App Inventor : <http://appinventor.mit.edu/>.

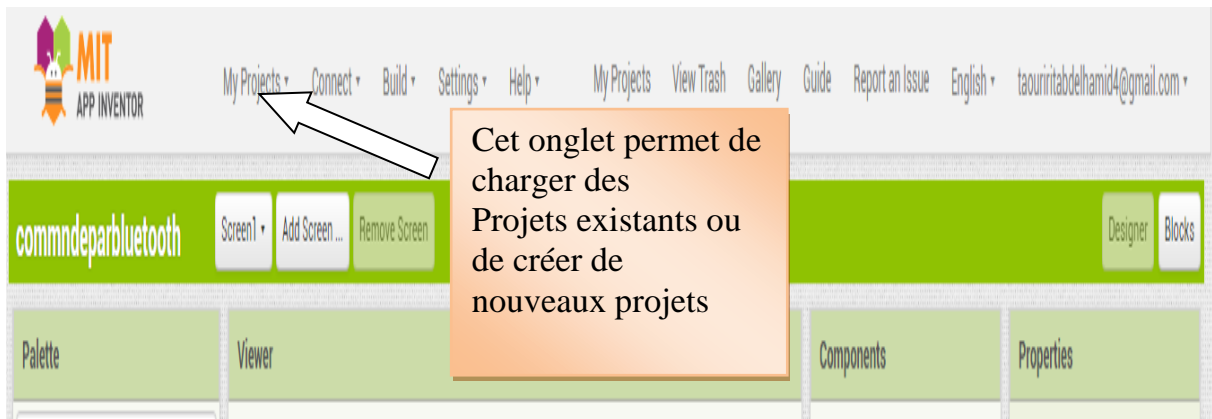


Figure III.25: site mit app inventor.

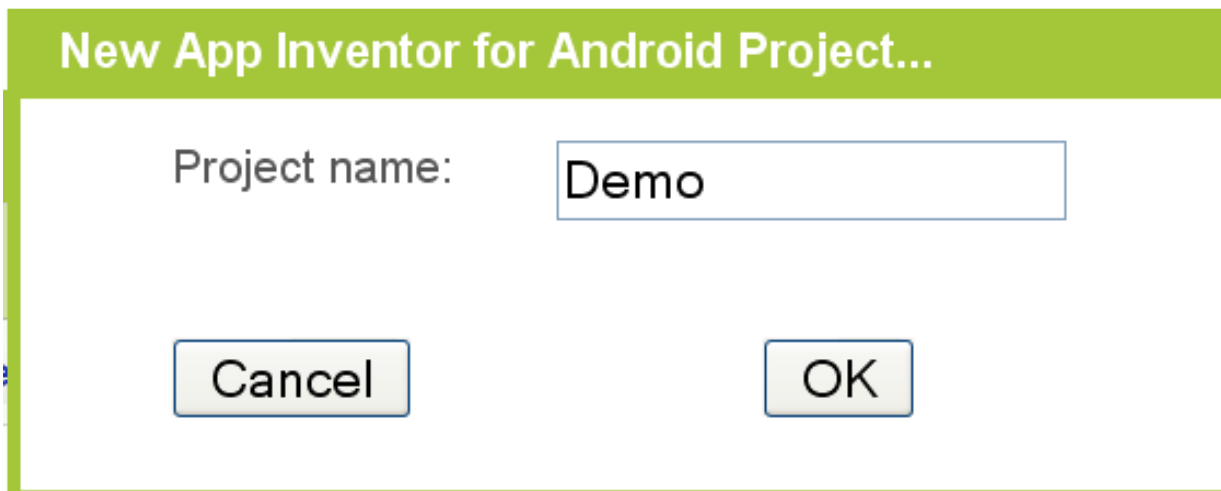
Cliquer sur



Chapitre III : Simulation et Réalisation d'un Robot mobile



Cliquer sur  . **Donner** un nom au nouveau projet.



Et voilà Interface graphique d'utilisateur :

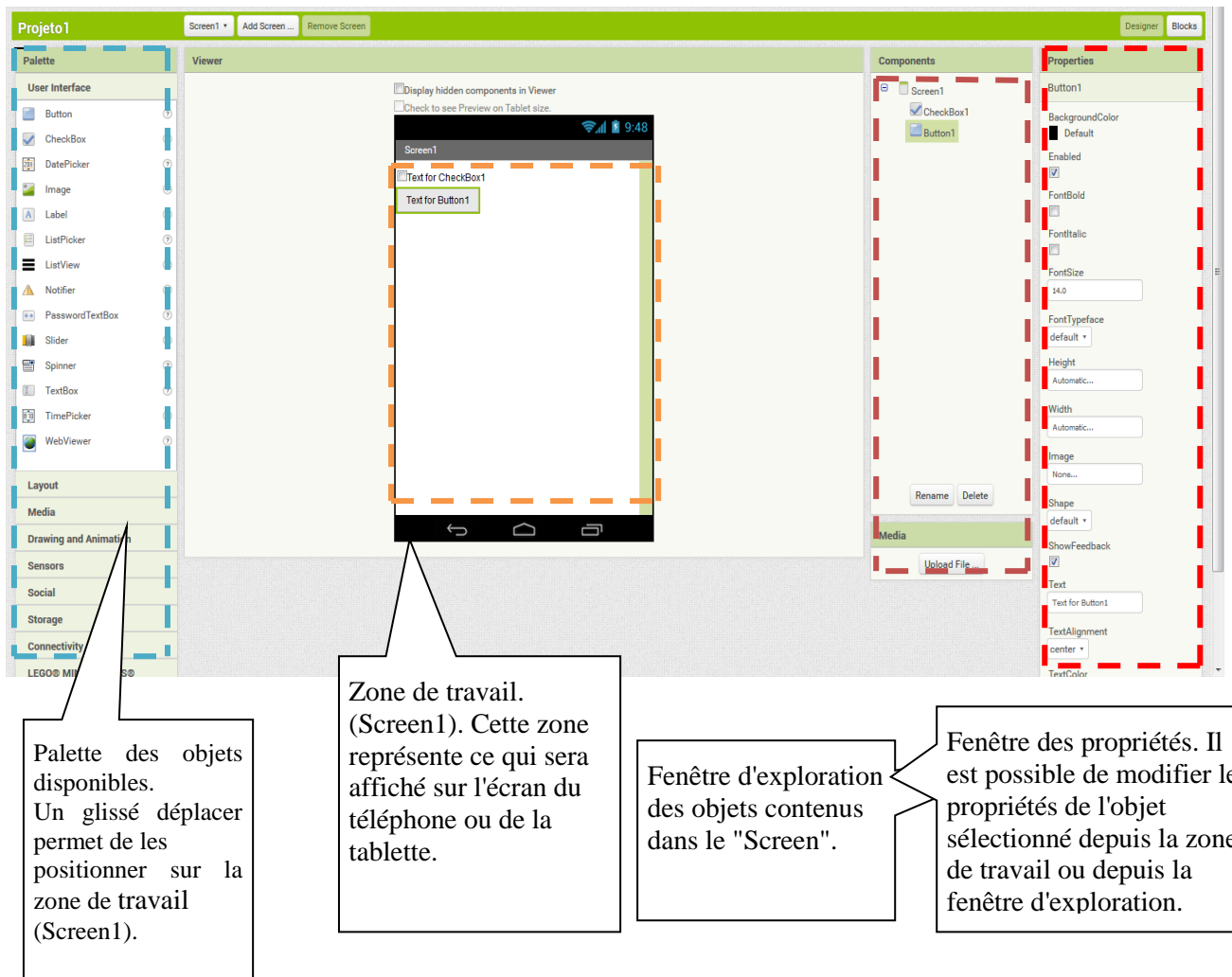


Figure III.26: première interface de la création App Inventor

III.7. La partie commande

III.7.1. La commande à distance de robot

III.7.1.1. La commande par utilisation de Bluetooth :

La disponibilité de Schild Bluetooth nous ouvre une autre technique de commande à la place de commande par câble USB utilisée au début de ce projet. On a pu aussi faire commander notre robot avec un téléphone portable avec une application Androïde. Pour ce faire, on commence par écrire un programme en Arduino IDE (voie l'annexe A) qui commande notre robot d'une façon autonome au manuellement par portable dont on a affecté à chaque bouton d'action un code spécial.

Chapitre III : Simulation et Réalisation d'un Robot mobile

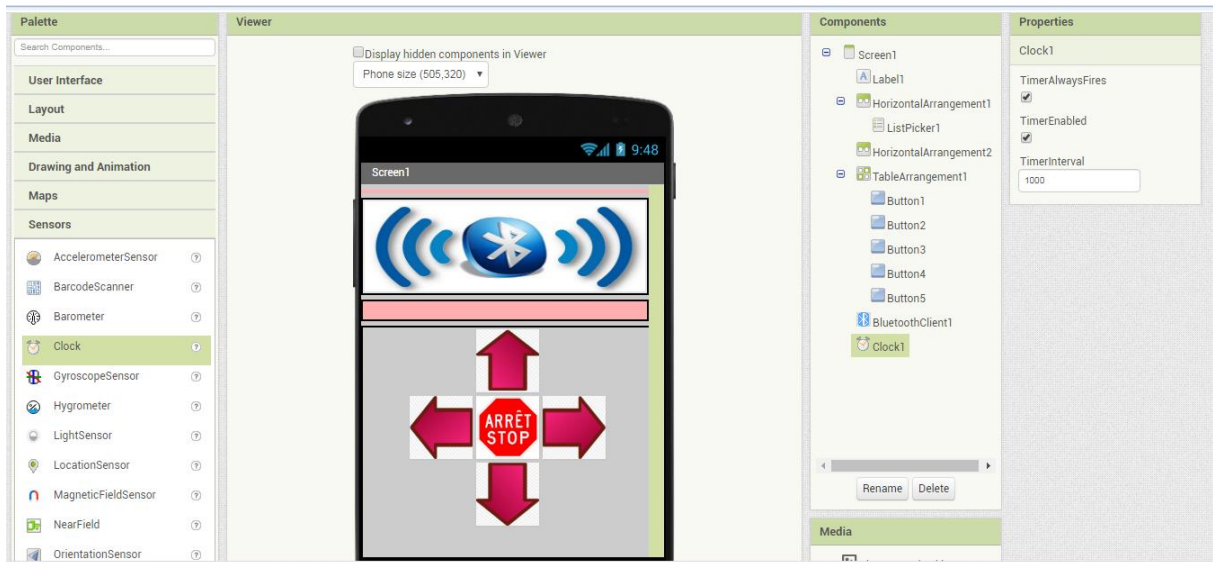


Figure III.27 : Application Androïde pour commande de robot.

Le langage utiliser pour crée l'application est *App Inventor* pour Androïde qui est une application développée par Google. Elle simplifie le développement des applications sous Androïde. Elle est basée sur une interface graphique attirante.

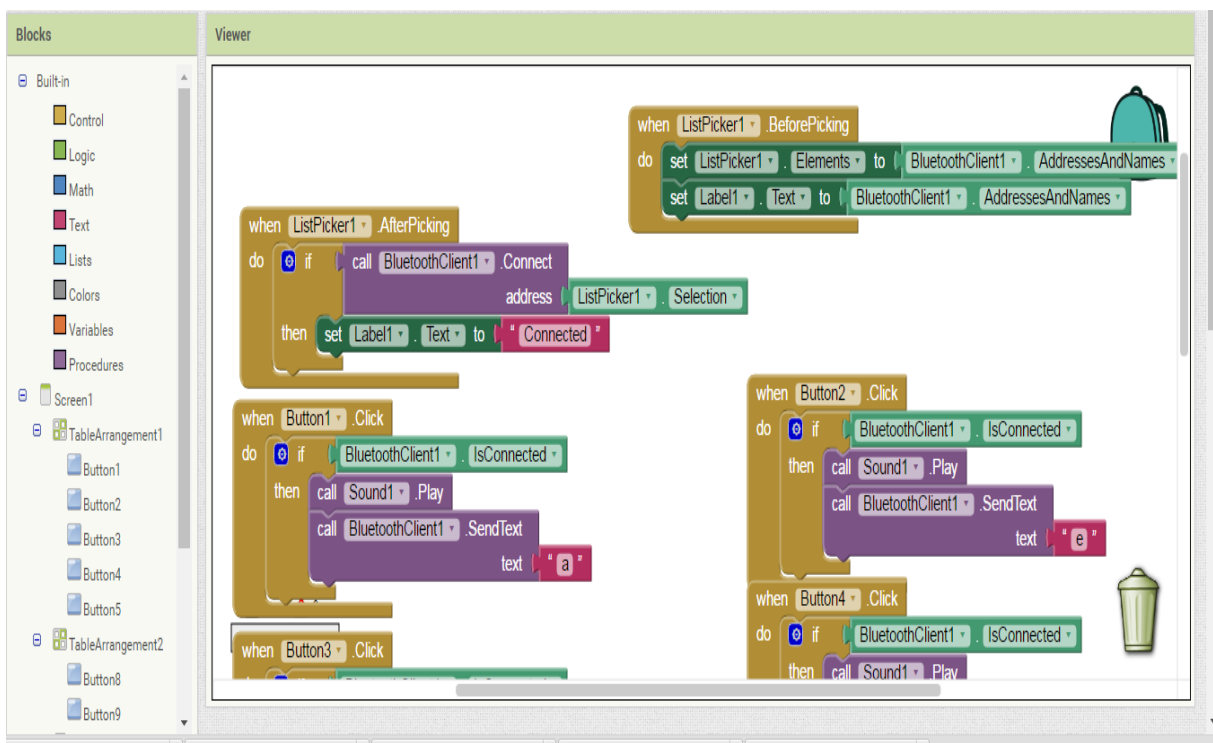


Figure III.28 : Interface du logiciel App Inventor pour applications Androïde.

Chapitre III : Simulation et Réalisation d'un Robot mobile

III.7.1.2. La commande par vocale : Le Smartphone permet aux utilisateurs de contrôler le robot avec la commande vocale. Lorsque la touche parole du GSM est enfoncée, il déclenche la voix de Google à la fonction de texte qui est intégré dans la plupart des appareils Androïde aujourd'hui. Une fois que la fonction est lancée, les utilisateurs sont invités à donner quelques commandes qui décideront de la direction du robot.

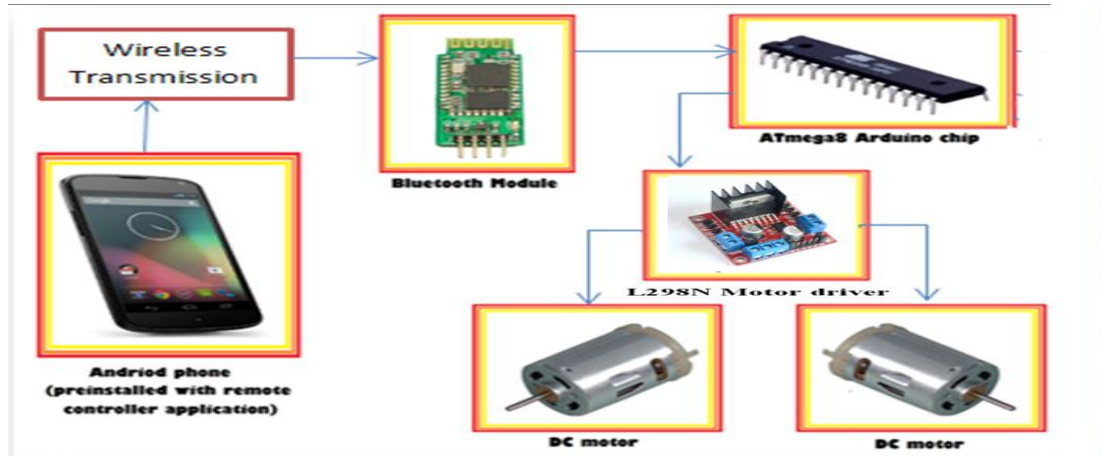


Figure III.29 : la commande par vocale.

a) Utilisation de App Inventeur

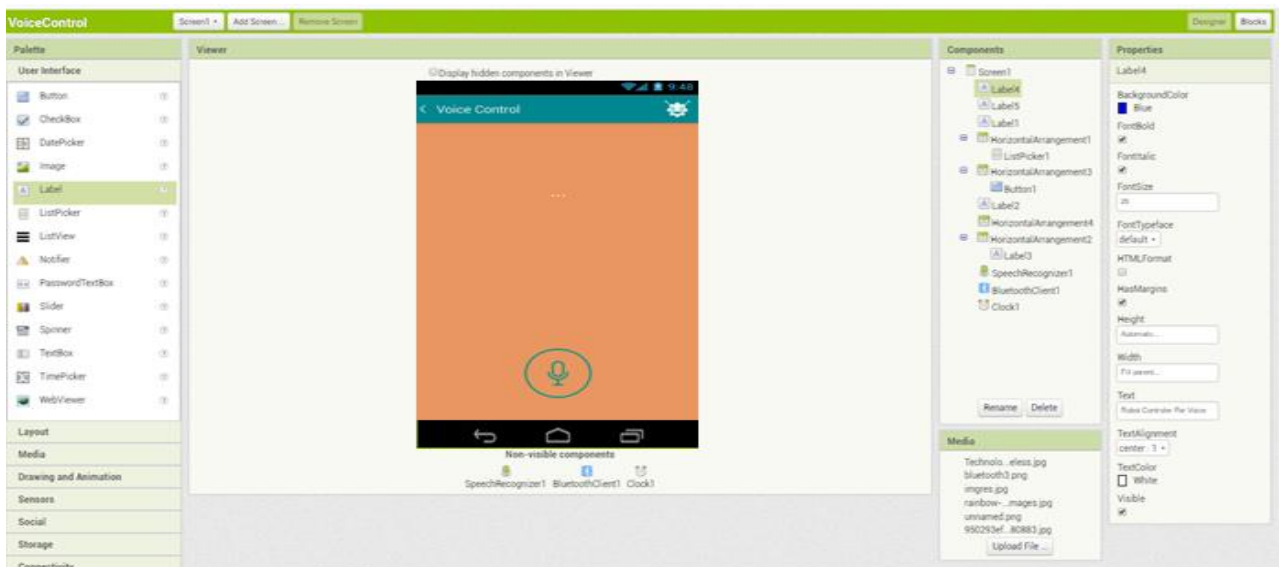


Figure III.30 : Application Androïde pour commande de robot vocale.

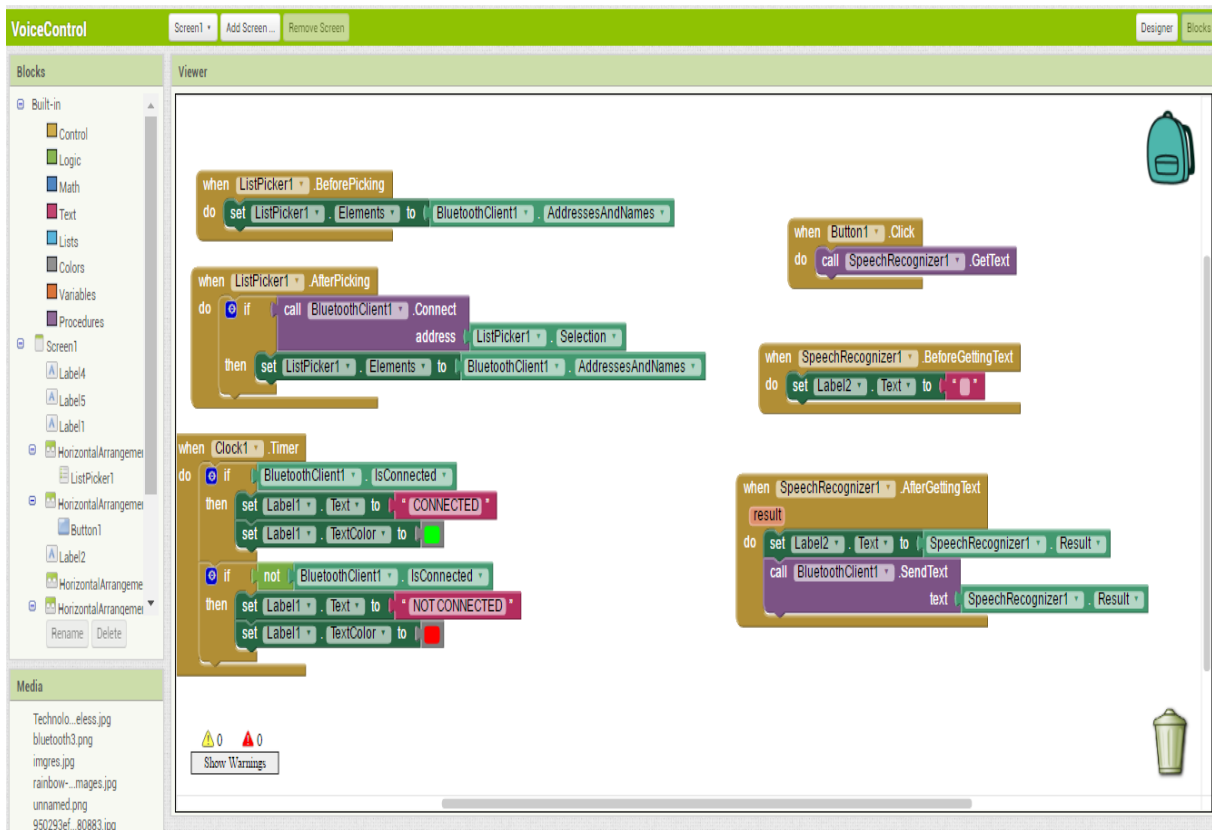


Figure III.31 : Interface du logiciel App Inventer pour applications Androïde vocale.

III.8.Conclusion

Dans ce chapitre nous avons présenté la constitution générale de notre système, ou on a cité les différentes composantes utilisées, et les schémas réalisés sous ISIS et la simulation par Proteus ainsi qu'une réalisation d'un système de commande d'un robot mobile.

On a touché dans ce chapitre Différents domaines: informatique, électronique programmé et électronique câblé ainsi que l'utilisation du l'automatique (autonome).

On a testé quelques méthodes de commande : directement par PC local, par téléphone portable via Bluetooth, et par vocale.

Conclusion générale

Le premier but de notre projet est de réaliser un système de commande d'un robot mobile à 4 roues. Ce dernier peut rouler d'une façon autonome avec évitement d'obstacle ou bien contrôlé par PC local, par téléphone portable via Bluetooth ou bien par voix.

Et pour aboutir aux fonctions citées ci-dessus, on a choisi d'utiliser la carte Arduino UNO comme une carte de commande et/ou d'interfaçage. Il est évidemment que nous avons utilisé des capteurs ultrason et des circuits intégrés de pilotage des moteurs à courant continu.

Concernant le système de commande réalisé, on peut dire qu'on a réalisé quatre variantes pour la commande du robot mobile :

- ✚ Commande par Arduino seulement (Autonome).
- ✚ Commande par PC via le port USB.
- ✚ Commande par mobile via Bluetooth.
- ✚ Commande par voix vocale.

Ce travail nous a permis de pratiquer et de développer nos connaissances dans la programmation, l'électronique et communication.

Enfin, nous estimons que ce mémoire va être développé au futur, pour introduire d'autres techniques de commande comme la commande visuelle et vocale en utilisant d'autres langages comme C++, Python, ...etc.

Bibliographie

- [1]: N. BANGSI Ulrich (Conception et réalisation d'un robot fouilleur) université de Douala
- [2]: TAKHI Hocine (conception et réalisation d'un robot mobile a base d'arduino) Memoire De Master Université Amar Telidji- Laghouat
- [3]: S. KERMICHE « Modélisation et commande d'un robot par méthodes intelligentes » Thèse doctorat d'état 2006, université Annaba .
- [4] : <https://en.wikipedia.org/wiki/Manipulator>.
- [5] : P. ALEXANDRE. (Le Contrôle Hiérarchise d'un Robot Marcheur Hexapode). Thèse vue de l'obtention du grade de Docteur 1996-1997
- [6] : A. AMRAOUI, Y. CHALA. « Réalisation D'un Robot mobile commandé à distance ».Mémoire d'ingénieur d'état université Biskra, Juin 2010.
- [7] : R. Carona « CONTROL OF UNICYCLE TYPE ROBOTS » Université de Lisboa, Portugal 2008.
- [8] :W. BENAZIZA « Commande par modes glissants du suivi de trajectoires pour un robot mobile » mémoire de Magister .université Batna 2013.
- [9] : <http://www.intechopen.com> .
- [10] : <http://www.aliexpress.com> .
- [11]: B. BAYLE. «Robotique Mobile», Ecole Nationale supérieur de Strasbourg, France, 2008-2009.
- [12] : <http://www.techpin.com>
- [13] : H. Takhi « commande optimal applique à un robot » mémoire de Magister .université Batna 2013.

Bibliographie

[14] : <http://www.generationrobots.com>

[15] : <http://www.robot-maker.com/forum/tutorials/article/85-deplacement-du-robot-roues-chenilles-ou-pattes/>

[16] : <http://www.energieplus-lesite.be>.

[17] : k. Khezazna ,I. Azzaoui « Réalisation et commande d'un robot mobile autonome » mémoire d'ingénieur université Biskra .

[18] : <https://fr.wikipedia.org/wiki/Servomoteur>.

[19] : D. Filliat: «Robotique Mobile». Rapport de l'École Nationale Supérieure de Techniques Avancées. cours c10-2 ENSTA pages : 59-64/ 2 octobre 2004.

[20] : N. Morette « Contribution à la Navigation de robots mobile: approche par modèle direct et commande prédictive » Thèse de Doctorat de l'université d'Orléans, 2009.

[21] : <https://learn.sparkfun.com/tutorials/what-is-an-arduino#introduction>

[22] : <https://www.arduino.cc/en/Guide/Introduction>.

[23] : http://f-leb.developpez.com/tutoriels/arduino/univers_arduino/part1/.

[24] : www.farnell.com/datasheets/1682209.Arduinouno.

[25] : <http://www.rakeshmondal.info/L293D-Motor-Driver>.

[26]: http://fastload.simple-download.net/motor_driver .

[27]: <http://lamptronics.com/product/1298n-motor-driver-board-module-for-arduino/>.

[28]: <http://clubelec-carentan.fr/index.php?page=ultrasons>.

[29]: <https://www.dirobots.com/fr/moteurs-servomoteurs/422-mini-moteur-motoreducteur-6v.html>.

Bibliographie

[30] : <http://www.robot-mobile-irbot.com/3-servomoteur-base-robot.html>.

[31] : http://www.sunfounder.com/wiki/index.php?title=Bluetooth_Transceiver_Module_HC-06.

[32] : <https://arduino-info.wikispaces.com/SensorShield>.

[33] : <http://www.elektronique.fr/logiciels/proteus.php>

[34] : F. cottet/, M. Pinard « LabVIEW : programmation et applications » Dunod, 2^e Edition, 2009.

Annexe A

Programme d'un robot mobile commande a distance (Bluetooth) ou autonome par le logiciel Arduino IDE.

```
//Coded By: Taouririt abd elhamide
// Activâtes Arduino (Bluetooth + Android)
#include <Servo.h> //beblioteque du servomotuer
Servo myservo;
const int trigPin= 2;    //la pin 2 d'arduino atacher au trig de l'ultrason
const int echoPin =4;   //la pin 4 d'arduino atacher au echo de l'ultrason
const int mot1a=9;      //  moteur 1+
const int mot1b=8;      //  moteur 1-
const int mot2a=13;     //  moteur 2+
const int mot2b=12;     //  moteur 2-
int ordre = 'STOP';
int duration, distance; // pour la calcule de distance.
void setup() {
  Serial.begin(9600);
  pinMode(mot1a, OUTPUT); //define le pin 9 comme sortie.
  pinMode(mot1b, OUTPUT);
  pinMode(mot2a, OUTPUT);
  pinMode(mot2b, OUTPUT);
  pinMode(echoPin, INPUT); } // define el pin 4 comme entrie
void loop() {
  if(Serial.available(>0){
    ordre = Serial.read();}
  if(ordre=='AVANT'){
    analogWrite(mot1b, LOW);
analogWrite(mot2b, LOW);
    analogWrite(mot1a, HIGH);
    analogWrite(mot2a, HIGH);    }
  if(ordre=='GOUCHE'){
    digitalWrite(mot1a, LOW); //
    digitalWrite(mot1b, LOW); //les deux moteur a gauche stop
    digitalWrite(mot2b, LOW);// les deux moteur a adroite marche avant
    digitalWrite(mot2a, HIGH);  }
  if(ordre=='STOP'){
    digitalWrite(mot1a, LOW);
    digitalWrite(mot1b, LOW);
    digitalWrite(mot2a, LOW);
    digitalWrite(mot2b, LOW); }
  if(ordre=='droite'){
```

```

digitalWrite(mot2a, LOW);
  digitalWrite(mot2b, LOW);
  digitalWrite(mot1b, LOW);
  digitalWrite(mot1a, HIGH); }
if(ordre=='ARRIERE'){
  digitalWrite(mot1a, LOW);
  digitalWrite(mot2a, LOW); //quant on change la polarite de moteur on change la direction
digitalWrite(mot2b, HIGH);
  digitalWrite(mot1b, HIGH); }
if (ordre =='ON'){ // robot mobile outonome
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance > 80 || distance < 0){
digitalWrite(mot1a,HIGH);
digitalWrite(mot1b,LOW);
digitalWrite(mot2a,HIGH);
digitalWrite(mot2b,LOW);}
else {
digitalWrite(mot1a,LOW);
digitalWrite(mot2a,LOW);
delay(200);
myservo.write(2300); //+90 degre
delay(1000);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
A=distance;
  delay(1500);
myservo.write(550); // -90 degre
  delay(1000);
digitalWrite(trigPin, LOW);

```

```

delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
B=distance;
delay(1500);
myservo.write(1300); // 0 degré
delay(1500);
if(A>=B) { // la comparaison rédige la direction du robot
  digitalWrite(mot1a,HIGH);
digitalWrite(mot1b,LOW);
digitalWrite(mot2a,LOW);
digitalWrite(mot2b,LOW);
delay(1500);}
else{
  digitalWrite(mot1a,LOW);
digitalWrite(mot1b,LOW);
digitalWrite(mot2a,HIGH);
digitalWrite(mot2b,LOW);
  delay(1500); }}
Serial.print(distance );
Serial.println (" cm");
delay(500);}
if(ordre=='OFF'){
  digitalWrite(mot1a, LOW);
digitalWrite(mot1b, LOW);
digitalWrite(mot2a, LOW);
digitalWrite(mot2b, LOW); }}

```

Programme d'un robot mobile commande par voix vocale

```
/Coded By: Taouririt Abd elhamide (2020)
//Voice Activated Arduino (Bluetooth + Android)
//Feel free to modify it but remember to give credit
#include <Servo.h>

Servo myservo;

String voice;

const int led1=11;
const int led2=10;
const int led3=5;
const int led4=7;
const int trigPin= 2;
const int echoPin =3;
int A;
int B;
int duration,distance;
void setup () {
  Serial.begin(9600);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  myservo.attach(6);
```



```

}

void loop() {

while (Serial.available()){ //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
  char c = Serial.read(); //Conduct a serial read
  if (c == '#') {break;} //Exit the loop when the # is detected after the word
  voice += c; //Shorthand for voice = voice + c
}
  if (voice.length() > 0) {
    Serial.println(voice);
//-----//
    if(voice == "o"){
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1;

if (distance > 80||distance < 2)
{
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
digitalWrite(12,HIGH);

```

```
digitalWrite(13,LOW);  
}
```

```
else {  
digitalWrite(8,LOW);  
digitalWrite(12,LOW);  
delay(200);  
myservo.write(2300); // sets the servo position according to the scaled value  
delay(1000);
```

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

```
duration = pulseIn(echoPin, HIGH);
```

```
distance = (duration/2) / 29.1;
```

```
A=distance;
```

```
delay(1500); // waits for the servo to get there
```

```
myservo.write(500); // sets the servo position according to the scaled value
```

```
delay(1000);
```

```
digitalWrite(trigPin, LOW);
```

```
delayMicroseconds(2);
```

```
digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);
```

```
duration = pulseIn(echoPin, HIGH);
```

```
distance = (duration/2) / 29.1;
```

```
B=distance;
```

```
delay(1500);
```

```
myservo.write(1300);           // sets the servo position according to the scaled value
```

```
delay(1500);
```

```
if(A>=B) {
```

```
    digitalWrite(8,HIGH);
```

```
digitalWrite(9,LOW);
```

```
digitalWrite(12,LOW);
```

```
digitalWrite(13,LOW);
```

```
delay(1500);
```

```
}
```

```
else{
```

```
    digitalWrite(8,LOW);
```

```
digitalWrite(9,LOW);
```

```
digitalWrite(12,HIGH);
```

```
digitalWrite(13,LOW);
```

```
    delay(1500);
```

```
}
```

```
}}
```

```
else if(voice == "2") {  
    digitalWrite(8, LOW);    // VOICE DE STOP
```

```
  
    digitalWrite(9, LOW);  
    digitalWrite(12, LOW);  
    digitalWrite(13, LOW);  
    digitalWrite(11,LOW);  
    digitalWrite(10,LOW);  
    digitalWrite(5,HIGH);  
    digitalWrite(7,HIGH);  
    delay (3000);
```

```
  
    digitalWrite(11,LOW);  
    digitalWrite(10,LOW);  
    digitalWrite(5,LOW);  
    digitalWrite(7,LOW);
```

```
}
```

```
else if(voice == "1")  
{ digitalWrite(9, HIGH);    // VOICE DE AVANT  
  digitalWrite(8, LOW);  
  digitalWrite(13, HIGH);
```

```
  
digitalWrite(12, LOW);  
    digitalWrite(11,HIGH);  
    digitalWrite(10,HIGH);  
    digitalWrite(5,LOW);
```

```
digitalWrite(7,LOW);
delay (3000);
digitalWrite(9, LOW);
digitalWrite(8, LOW);
digitalWrite(13, LOW);
digitalWrite(12, LOW);

}

else if(voice == "5") {
    digitalWrite(8, HIGH);    // VOICE DE ARIERRE
digitalWrite(9, LOW);
digitalWrite(12, HIGH);
digitalWrite(13, LOW);

digitalWrite(5,HIGH);
digitalWrite(7,HIGH);
digitalWrite(10,LOW);
digitalWrite(11,LOW);
delay (3000);
digitalWrite(8, LOW);    // VOICE DE ARIERRE
digitalWrite(9, LOW);
digitalWrite(12, LOW);
digitalWrite(13, LOW);

}
else if(voice == "4") {
digitalWrite(8, LOW);    // VOICE DE GAUCHE
```

```
digitalWrite(9, HIGH);  
digitalWrite(12, LOW);  
  digitalWrite(10,HIGH);  
digitalWrite(7,HIGH);  
  digitalWrite(11,LOW);  
digitalWrite(5,LOW);  
digitalWrite(13, LOW);
```

```
}
```

```
else if(voice == "3") {  
  digitalWrite(8, LOW);    // VOICE DE DROITE  
digitalWrite(9, LOW);  
digitalWrite(12, LOW);  
digitalWrite(13, HIGH);
```

```
  digitalWrite(11,HIGH);  
digitalWrite(5,HIGH);  
  digitalWrite(10,LOW);  
digitalWrite(7,LOW);  
}
```

```
//-----//  
voice="";} //Reset the variable after initiating
```

