

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

présenté par

Belkacemi Oussama

&

Doual Samir

Robot à commande vocale via Bluetooth et Smartphone

Proposé par : Farid Ykhlef

Année Universitaire 2021-2022

Remerciements

D'abord nous remercions le bon Dieu à nous avoir donné la force, la volonté, et le courage pour réaliser ce travail.

Nos premiers remerciements sont adressés à nos enseignants et enseignantes à Université Saad Dahleb Blida 1 département d'électronique spécialité

« **INSTRUMENTATION** » pour leurs contributions à notre formation en master.

Nous tenons à remercier vivement notre promoteur **Mr.YEKLEF FARID** pour son orientation et sa disponibilité constante tout au long de notre travail

Nos remerciements vont également aux membres de jury qui ont accepté d'évaluer notre travail.

Enfin nous remercions tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet

Ma Mère, Mon Père

Affable, honorable, aimable, vous représentez pour moi Le symbole de la bonté par excellence, la source de tendresse Et l'exemple du Dévouement qui n'a pas cessé de m'encourager et de prier Pour moi.

Soyez sûrs que je continuerai mon chemin. Je vous dédie ce travail en témoignage de mon profond amour.

Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, Longue vie et bonheur.

A mes frères

A tous les membres de ma famille, petits et grands

A tous mes amis de proches Veuillez trouver dans ce modeste travail l'expression de mon Affect

B.oussama

Je dédie ce travail à mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et Persévérance

A ma chère mère qui a éclairé mon chemin en témoignage de ma Profonde gratitude et de mon incontestable reconnaissance, pour tous les sacrifices qu'elle me contente, toute la confiance qu'elle m'accorde et tout l'amour dont elle m'entoure.

A mon cher père qui m'a indiqué la bonne voie A ma compagne de vie qui a été toujours avec moi en me Rappelant que la volonté fait toujours les grands hommes aussi Grâce à son encouragement, son soutien moral en exprimant mes Gratitude.
Que Dieu vous garde.

À Toute mes frères et mes sœurs.

À Tous ma famille.

À tous mes amis

D.samir

ملخص:

هدفنا الرئيسي هو التحكم في الروبوت باستخدام الصوت البشري عبر تقنية البلوتوث للهاتف الذكي. بناءً على النتائج، يمكن لروبوت أداء المهام التالية: التحرك للأمام والخلف والانعطاف إلى اليسار والانعطاف إلى اليمين. يتم توصيل الأوامر الصوتية بشكل فعال من خلال البلوتوث، ويتم تنفيذ الإجراءات المطلوبة بنجاح. لتحقيق هدفنا، استخدمنا لوحة اردوينو ATmega328P مع ملحقاتها مثل Motor Driver والمحركات وأجهزة الاستشعار فوق الصوتية ووحدة البلوتوث من لوحة اردوينو.

كلمات المفاتيح: الروبوت; اردوينو ATmega328P; الهاتف الذكي.

Résumé :

Notre objectif principal est de commander un ROBOT en utilisant la voix humaine via la technologie Bluetooth d'un Smartphone. À partir des résultats, le ROBOT peut effectuer les tâches suivantes : avancer, reculer, tourner à gauche et tourner à droite. Les commandes vocales sont efficacement communiquées grâce à l'innovation Bluetooth, et les actions requises sont effectuées avec succès. Pour atteindre notre objectif, nous avons utilisé une carte Arduino ATmega328P avec ses accessoires tels que le Motor Driver, des moteurs, des capteurs ultrasonores, un module Bluetooth de la carte Arduino.

Mots clés : ROBOT ; Arduino ATmega328P ; Smartphone.

Abstract:

Our main objective is to control a ROBOT using the human voice via the Bluetooth technology of a Smartphone. Based on the results, the ROBOT can perform the following tasks: forward, backward, turn left, and turn right. Voice commands are effectively communicated through Bluetooth innovation, and required actions are performed successfully. To achieve our goal, we used an Arduino ATmega328P board with its accessories such as the Motor Driver, motors, ultrasonic sensors, a Bluetooth module from the Arduino board.

Keywords : ROBOT ; Arduino ATmega328P ; Smartphone.

Listes des acronymes et abréviations

DDL : degrés de libertés

MCI : Modèle cinématique inverse

MCD : Modèle cinématique direct

SMA : système mécanique articulé

IDE : Environnement de Développement Intégré

MIT : Massachusset Institute of Technology

AOT : Ahead Of Time

JIT : Just-In-Time

$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$

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

$\dot{\mathbf{q}}$: Vitesse des coordonnées généralisées.

\mathbf{J} : c'est une matrice Jacobéenne.

Table des matières

INTRODUCTION GENERALE.....	1
CHAPITRE 1 GENERALITES SUR LA ROBOTIQUE MOBILE.....	3
1.1 INTRODUCTION.....	3
1.2 DEFINITION.....	3
1.2.1 La robotique.....	3
1.2.2 Robot.....	4
1.3 STRUCTURE D'UN ROBOT.....	6
1.4 MODELE DU ROBOT.....	6
1.4.1 Modèle cinématique.....	6
a. <i>Modèle cinématique direct (MCD)</i>	7
b. <i>Modèle cinématique inverse (MCI)</i>	7
1.4.2 Modèle dynamique.....	7
1.5 TYPES DE ROBOTS.....	8
1.5.1 Robots manipulateurs.....	8
1.5.2 Robots mobiles.....	9
a. <i>Architecture des robots mobiles</i>	10
b. <i>Classification des robots mobiles</i>	11
1.6 AVANTAGES ET INCONVENIENTS DES ROBOTS.....	14
1.6.1 Les avantages [16].....	14
1.6.2 Les inconvénients.....	15
1.7 CONCLUSION.....	15
CHAPITRE 2 PRESENTATION DE LA CARTE ARDUINO.....	16
2.1 INTRODUCTION.....	16
2.2 ARDUINO.....	16

2.2.1	Début de Arduino.....	17
2.2.2	Fonction de la carte Arduino.....	17
2.2.3	Définition de la carte Arduino.....	18
2.2.4	Microcontrôleur ATmega328 de la carte Arduino	18
2.3	FAMILLES DE LA CARTE ARDUINO.....	19
2.4	TYPES DE CARTE ARDUINO	20
2.5	CHOIX D'ARDUINO	21
2.6	ARDUINO UNO	21
2.6.1	Composition de carte Arduino UNO	22
a.	<i>Alimentation de la carte Arduino</i>	23
b.	<i>Programmation</i>	24
2.7	BLUETOOTH.....	25
2.7.1	Module Bluetooth Arduino	26
2.7.2	Différence entre HC-05 et HC-06	27
2.8	SERVOMOTEUR.....	27
2.8.1	Fonctionnement des servomoteurs	28
2.8.2	Avantages et inconvénients [15].....	30
a.	<i>Avantages</i>	30
b.	<i>Inconvénients</i>	30
2.9	SHIELD MOTEUR L293D.....	30
2.10	CAPTEUR A ULTRASON HC-SR04.....	31
2.10.1	Caractéristique du HC-SR04	32
2.10.2	Configuration des broches du HC-SR04.....	32
2.11	MINI MOTEUR ET ROUES.....	32
2.11.1	Mini moteur motoréducteur 6V	32
2.11.2	Roues	33

2.12	CONCLUSION	34
CHAPITRE 3 SIMULATION ET REALISATION DU ROBOT : PARTIE MATERIELLE		
ET LOGICIELLE		35
3.1	INTRODUCTION	35
3.2	ÉLABORATION DU PROTOTYPE	35
3.3	LE SYSTEME D'EXPLOITATION ANDROID	37
3.3.1	Définition.....	37
3.3.2	Logo.....	37
3.3.3	L'architecture d'Android	38
3.4	PRESENTATION DE MIT APP INVENTOR	39
3.4.1	Conception d'APP INVENTOR	39
a.	<i>L'interface graphique.....</i>	<i>Erreur ! Signet non défini.</i>
b.	<i>Blocs de programmation</i>	40
c.	<i>L'émulateur</i>	40
3.4.2	Concevoir un projet	40
3.5	FLUTTER	42
3.5.1	Langage de programmation DART	43
3.5.2	Caractéristiques et avantages du Flutter	44
3.6	CONCEPTION DE L'APPLICATION.....	45
3.7	SIMULATION DU ROBOT	47
3.7.1	ISIS PROTEUS	47
3.8	REALISATION DU ROBOT MOBILE	49
3.8.1	Assemblage électrique et connexion au Bluetooth.....	49
3.8.2	Assemblage global	49
CONCLUSION GENERALE		53

ANNEXES.....	54
BIBLIOGRAPHIE	58

Liste des figures

Figure 1.1 : robot manipulateur mobile

Figure 1.2 : robot manipulateur mobile

Figure 1.3 : architecture robot manipulateur

Figure 1.4 : robots mobiles

Figure 1.5 : architecture d'un robot mobile

Figure 1.6 : robots mobiles a chenilles

Figure 1.7 : robots mobiles marcheurs

Figure 1.8 : robots mobiles rampant

Figure 2.1 : interagir de la carte Arduino avec le monde réel

Figure 2.2 : schéma fonctionnelle d'un microcontrôleur

Figure 2.3 : microcontrôleur ATmega328p

Figure 2.4 : carte Arduino UNO

Figure 2.5 : alimentation de la carte Arduino UNO

Figure 2.6 : environnement de développement interactif pour la carte Arduino

Figure 2.7 : logo de Bluetooth

Figure 2.8 : module Bluetooth Arduino

Figure 2.9 : caractéristique et alimentation du module Bluetooth

Figure 2.10 : composition d'un servomoteur

Figure 2.11 : fonctionnement d'un servomoteur

Figure 2.12 : signal de commande servomoteur

Figure 2.13 : shield moteur L293D

Figure 2.14 : capteur ultrason

Figure 2.15 : mini moteur motoréducteur

Figure 2.16 : la roue tamya

Figure 3.1 : schéma général du prototype

Figure 3.2 : graphique synoptique du prototype

Figure 3.3 : logo android

Figure 3.4 : architecture android

Figure 3.5 : logo app Inventor

Figure 3.6 : conception d'une application android par APP INVENTOR

Figure 3.7 : site MIT INVENTOR

Figure 3.8 : nouveaux projet sur APP INVENTOR

Figure 3.9 : nom du projet sur APP INVENTOR

Figure 3.10 : interface d'utilisateur sur APP INVENTOR

Figure 3.11 : logo flutter

Figure 3.12 : exemple de développement d'application Android sur flutter

Figure 3.13 : interface du code source

Figure 3.14 : code source utilisant MIT APP INVENTOR

Figure 3.15 : code source utilisant flutter

Figure 3.16 : l'interface d'utilisateur de l'application

Figure 3.17 : logo logiciel Proteus

Figure 3.18 : l'interface d'utilisateur Proteus

Figure 3.19 : schéma du circuit sur Proteus

Figure 3.20 : schéma du robot mobile

Figure 3.21 : montage électrique du robot mobile

Figure 3.22 : caractéristique et alimentation du module Bluetooth

Figure 3.23 : photos finale du robot mobile

Liste des tableaux

Tableau 1.1 : classes de robots mobiles a roues

Tableau 2.1 : synthèse des caractéristique Arduino UNO

Introduction générale

Depuis 1970, le monde scientifique et industriel s'est toujours focalisé sur l'étude et la réalisation de machines et de robots mobiles contrôlés et autonomes. Le but de l'homme est de s'affranchir du travail dans ce qui est exigeant, fatigant, inintéressant, donc de se simplifier la vie. Les robots mobiles pourront se déplacer n'importe où, notamment dans un environnement hostile auquel l'homme ne peut pas accéder, et effectuer des tâches à la place de l'homme, notamment dans un environnement domestique [1].

La robotique a deux principaux domaines d'intérêt, la robotique de manipulation (robotique industrielle) et la robotique mobile. L'un des problèmes majeurs de la robotique mobile est la planification des mouvements. Autour de ce problème de planification de mouvement de nombreuses études ont été menées afin de développer des méthodes générales de guidage de robots.

Les robots de première génération étaient de simples automates capables d'effectuer des tâches répétitives dans des environnements bien connus. La seconde génération voit apparaître des robots équipés de capteurs leur permettant de percevoir les changements de l'environnement et d'agir de manière simple. Depuis, les robots se sont largement développés et généralisés. Ils sont dotés de moyens de perception et de prise de décision leur permettant d'appréhender et d'évoluer dans différents types d'environnement [2].

Certains sont même utilisés pour l'exploration de planètes lointaines (Lune, Mars). La dernière génération de robots devrait être dotée d'un haut degré d'autonomie, elle fait

l'objet de recherches intensives. On s'est de plus en plus rendu compte que pendant de nombreuses années encore, la coopération entre l'homme et la machine est inévitable.

C'est dans ce contexte que se situe notre projet de fin d'études intitulé "Robot à commande vocale via Bluetooth et Smartphone".

Notre objectif principal est de contrôler un ROBOT à l'aide de la voix via Bluetooth pour permettre aux utilisateurs de contrôler ce ROBOT via un Smartphone Android.

Pour atteindre notre objectif, nous avons utilisé une carte Arduino ATmega328P, ainsi que d'autres équipements tels que le Motor Driver, des moteurs, des capteurs (Ultrason), un module Bluetooth, etc.

Notre projet est structuré autour de deux parties :

- La partie contrôle : c'est l'ensemble des composants mécaniques, électroniques et informatiques composant le ROBOT, l'essentiel de cette partie est la carte Arduino connectée au Shield.
- La partie commande : c'est une application Android qui permet à l'utilisateur de contrôler le ROBOT à la voix avec une connexion Bluetooth. Ces deux parties sont reliées par un module Bluetooth qui assure la communication.

Ainsi que notre travail est divisé en trois chapitres :

- **Le premier chapitre** présente une introduction à la robotique : définitions, historique, capteurs et différentes techniques de commande.
- **Le deuxième chapitre** est une explication de la carte de commande Arduino.
- **Le troisième chapitre** présente principalement les deux parties constitutives du système de commande : la partie matérielle (Hardware) et la partie logicielle (Software).

Enfin nous terminons ce manuscrit par une conclusion générale résumant l'ensemble de notre travail ainsi que ses perspectives.

Chapitre 1 Généralités sur la robotique mobile

1.1 Introduction

La robotique a été un contributeur majeur au développement de la société humaine pendant de nombreuses années. C'est un domaine qui nécessite la synergie de divers domaines scientifiques tels que le génie mécanique, le génie électrique et électronique, le génie du contrôle, le génie informatique, le génie des capteurs, etc. Les robots et autres fonctions automatisées des machines doivent coexister avec les humains.

Dans cette relation symbiotique, les besoins et les préférences humaines doivent d'abord être respectés, intégrés et mis en œuvre.

À cette fin, les robots modernes, en particulier les robots mobiles à roues ou à pattes, intègrent et mettent en œuvre des cycles perceptifs d'actions de manière ciblée et rentable.

Les robots mobiles sont désormais considérés comme automatisant de nombreuses tâches. Celles-ci sont particulièrement diverses : nettoyage, transport dans des ateliers automatisés, agriculture, exploitation minière, assistance aux personnes handicapées, exploration de milieux difficiles en sont quelques exemples.

Le travail de la robotique a pour but de concevoir et de construire des machines capables d'évoluer et d'interagir avec l'environnement physique pour effectuer les différentes tâches qui les génèrent.

1.2 Définition

1.2.1 La robotique

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots [3].

De cette définition découlent deux interprétations : la première serait de voir le robot comme une machine, qui dispose de capteurs, d'un système logique et d'actionneurs. Il est matériel. La seconde suggère qu'un robot peut aussi être virtuel [4].

La robotique actuelle trouve des applications dans différents domaines:

- la robotique industrielle,
- la robotique domestique,
- la robotique médicale,
- la robotique militaire,
- la robotique sociale,
- la robotique scientifique, par exemple pour l'exploration de l'espace [5,6] (aérobot), des fonds marins (robots sous-marins autonomes), dans les laboratoires d'analyse (robotique de laboratoire), etc., ou encore
- la robotique de transport (de personnes et de marchandises), avec par exemple ROPITS (Robot for Personal Intelligent Transport System) [7], Robosoft [8], RoboCourier [9], etc.

Le mot robotique vient de robot. Selon l'Oxford English Dictionary, le terme robotique a été inventé pour la première fois par l'auteur russo-américain **Isaac Asimov** dans son roman de science-fiction **LIAR** (Menteur), publié dans AmazingScience Fiction en mai 1941. Dans certains des autres écrits d'Asimov, il prétend avoir utilisé le mot robotique pour la première fois dans sa nouvelle Runaround (Shocking Science Fiction, mars 1942). Cependant, le menteur d'origine! Cinq mois plus tôt que Runaround, le premier est donc considéré comme à l'origine du terme.

Asimov ignorait initialement que le terme était devenu populaire. Il a supposé que le terme existait déjà, similaire à la «mécanique» (comme «positronique» et «électronique»), et à d'autres termes similaires pour les branches des sciences appliquées [10].

1.2.2 Robot

Un robot est un assemblage complexe de pièces mécaniques, électromécaniques ou électroniques. L'ensemble est piloté par une unité centrale appelée « système embarqué » : une simple séquence d'automatisation, de logiciel informatique ou d'intelligence artificielle selon le degré de complexité des tâches à effectuer. Lorsque

les robots autonomes sont mobiles, ils disposent également d'une source d'énergie embarquée : généralement une batterie de stockage électrique ou un générateur électrique couplé à un moteur à essence pour les plus énergivores.

Le terme robot apparaît pour la première fois dans la pièce de théâtre (de science-fiction) R.U.R. (Rossum's Universal Robots), écrite en 1920 par l'auteur Karel Čapek. Le mot a été créé par son frère Josef à partir du mot tchèque "robota" qui signifie "travail, besogne, corvée".

Les premiers robots industriels sont apparus, malgré leur coût élevé, au début des années 1970. Ils étaient destinés à effectuer certaines tâches répétitives, éprouvantes ou toxiques pour un opérateur humain : peindre ou souder des carrosseries automobiles. Aujourd'hui, l'évolution de l'électronique et de l'informatique permet de développer des robots plus précis, plus rapides ou avec une meilleure autonomie. Industriels, militaires ou chirurgiens spécialistes rivalisent d'inventivité pour développer des robots assistants les aidant à effectuer des tâches délicates ou dangereuses. Parallèlement, des robots à usage domestique apparaissent : aspirateurs, tondeuses, etc.

L'usage du terme « robot » a été galvaudé pour prendre des acceptions plus larges : distributeur automatique, appareil électromécanique à forme humaine ou animale, logiciel faisant office d'adversaire sur les plateformes de jeu, bot informatique.

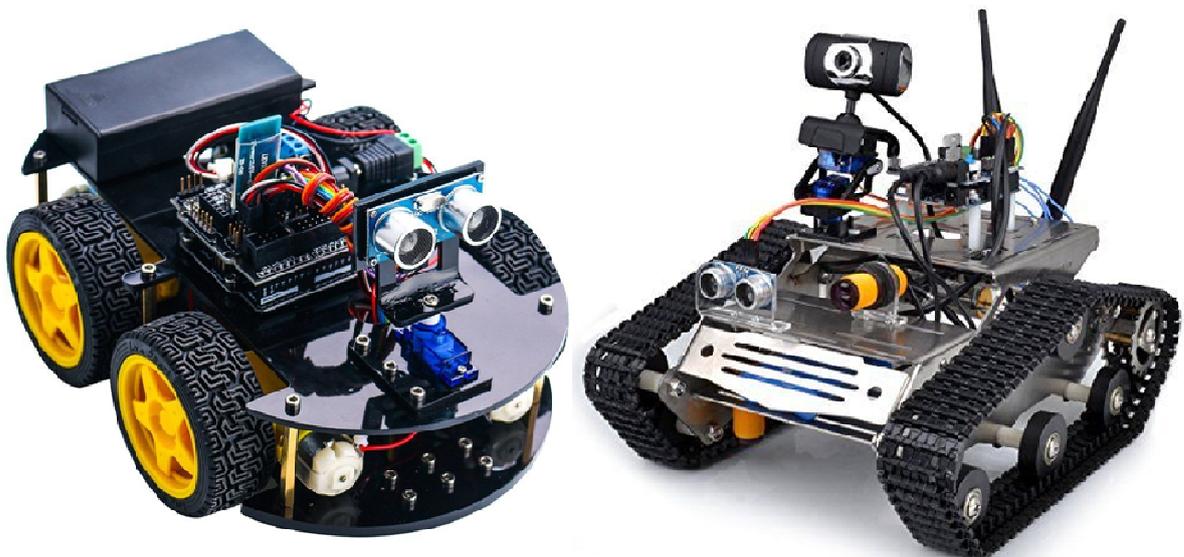


Figure 1.1. Robots manipulateurs mobiles.

1.3 Structure d'un robot

Un robot est un appareil capable d'effectuer une ou plusieurs fonctions. Il se compose de sept parties principales : le socle à un ou plusieurs bras articulés, équipé de systèmes de préhension (brides, ventouses, etc.), la source d'énergie, les actionneurs, les systèmes de transmission, les capteurs internes, les capteurs externes (position, température, etc.) et le système de traitement de l'information.

Cette énergie peut être utilisée à des fins pneumatiques, hydrauliques ou électriques, celle-ci étant privilégiée par les utilisateurs et constructeurs de robots, pour déplacer des charges inférieures à 1000 kg. Les actionneurs sont des dispositifs capables de modifier le fonctionnement du robot, en gérant les transferts d'énergie à partir des signaux émis par le système d'automatisation. Ceux-ci prennent généralement la forme de systèmes d'asservissement (voir servomécanisme).

Les capteurs intérieurs et extérieurs permettent de mesurer des grandeurs mécaniques, thermiques, électriques ou chimiques (température, position, concentration chimique, etc.). Les valeurs mesurées sont transformées en un signal électrique et transmises à un autre appareil (un appareil électronique) capable de lire, d'enregistrer ou de contrôler les grandeurs mesurées.

1.4 Modèle du robot

1.4.1 Modèle cinématique

La cinématique complète la modélisation géométrique en établissant les relations entre les vitesses des paramètres articulaires (\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. La cinématique des robots traite de la configuration des robots dans leur espace de travail, des 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 une structure cartésienne, cylindrique, sphérique ou articulée, et un robot mobile peut en avoir une, deux, trois ou plus, avec ou sans roues contraintes dans leur mouvement.

L'étude de la cinématique est un préalable fondamental pour l'étude de la dynamique, des caractéristiques de stabilité et du contrôle des robots. Le développement de nouvelles structures spécialisées et de la cinématique robotique fait toujours l'objet de recherches continues, vers la fin de la construction de robots capables d'effectuer des tâches plus sophistiquées et complexes dans des applications industrielles et sociétales [12].

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 articulaires (\dot{q}_i), $i = 1$ à n .

Le MCD est de la forme [13] : $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$

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

$\dot{\mathbf{q}}$: Vitesse des coordonnées généralisées.

\mathbf{J} : c'est une matrice Jacobéenne.

b. Modèle cinématique inverse (MCI)

Le but de ce modèle est de trouver les vitesses articulaires qui génèrent la vitesse de point final souhaitée. Sa détermination consiste à inverser la matrice Jacobéenne, ce qui n'est pas toujours possible. Le modèle cinématique inverse est de la forme :

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{x}} \text{ (si } \det(\mathbf{J}(\mathbf{p})) \neq 0 \text{).}$$

1.4.2 Modèle dynamique

On s'intéresse dans ce modèle aux efforts des actionneurs produits par les mouvements du système mécanique articulé (SMA). Ce dernier est constitué d'un ensemble de corps solides articulés entre eux conférant à l'organe terminal (le préhenseur ou l'outil) les degrés de libertés (ddl) souhaités pour générer un mouvement.

Il s'agit d'établir les équations différentielles non linéaires qui lient les forces d'actionnement $\Gamma_i(t)$ aux variables conjointes $q_i(t)$, aux vitesses conjointes $\dot{q}_i(t)$ et les accélérations articulaires $\ddot{q}_i(t)$.

L'ensemble de ces équations constitue ce qu'on appelle communément le modèle dynamique du manipulateur. Pour cela il existe plusieurs formalismes de modélisation tels que le formalisme de Newton-Euler, le formalisme de Lagrange-Euler, le principe d'Alembert, etc. [12].

1.5 Types de robots

Il existe deux grandes familles de robots :

- Robots manipulateurs (fixes).
- Robots mobiles.

1.5.1 Robots manipulateurs

Un robot manipulateur est un bras avec un certain nombre de segments articulés. Il est destiné à la manipulation ou au déplacement de matériaux, d'outils et de pièces sans contact humain direct. En même temps, les robots sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement sécurisé. Par conséquent, les robots manipulateurs sont utilisés dans des applications industrielles pour effectuer efficacement des tâches telles que l'assemblage, le soudage, le traitement de surface et le forage.

Les robots manipulateurs se présentent sous de nombreuses formes, qui sont divisées en cinq grandes catégories :

- Robots cylindriques.
- Robots rectilignes.
- Robots sphériques.
- Robots articulés.
- Robots SCARA.

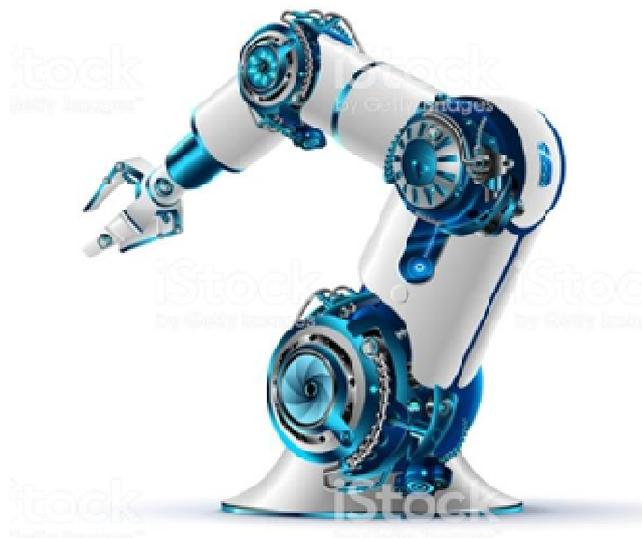


Figure 1.2.Robot manipulateur [14].

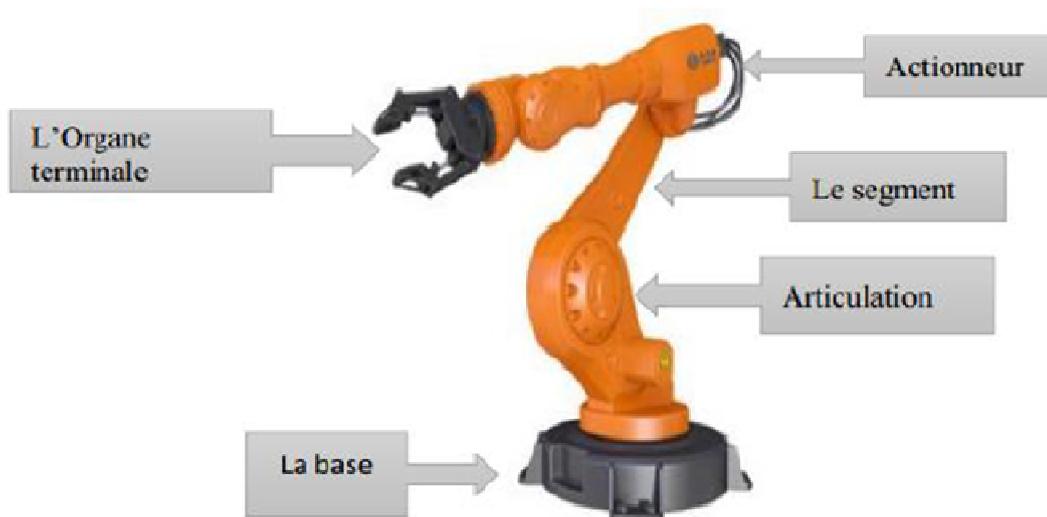


Figure 1.3.Architecture d'un robot manipulateur [14].

1.5.2 Robots mobiles

Un robot mobile est un robot qui peut se déplacer dans son environnement de façon autonome. Pour y parvenir, le robot doit pouvoir naviguer. Par conséquent, la portée requise et la précision de navigation dépendent de la taille du robot et du type de tâche à effectuer.

L'utilisation de robots mobiles est désormais couramment envisagée pour l'automatisation de nombreuses tâches. Celles-ci sont particulièrement diverses : le nettoyage, le transport dans des ateliers automatisés, l'agriculture, l'exploitation minière, l'assistance aux personnes handicapées, l'exploration de milieux hostiles, etc. L'architecture mécanique d'un robot mobile dépend en grande partie des spécificités de la mission à réaliser et de l'environnement de travail. Ces données conditionnent, entre autres, le choix d'un système de locomotion approprié.



Figure 1.4. Robots mobiles.

a. Architecture des robots mobiles

L'architecture des robots mobiles est illustrée à la **figure 1.5**. Il se compose de quatre éléments à savoir :

1. La structure mécanique et la motricité.
2. Les organes de sécurité.
3. Le système de traitement des informations et gestion des tâches.
4. Le système de localisation.

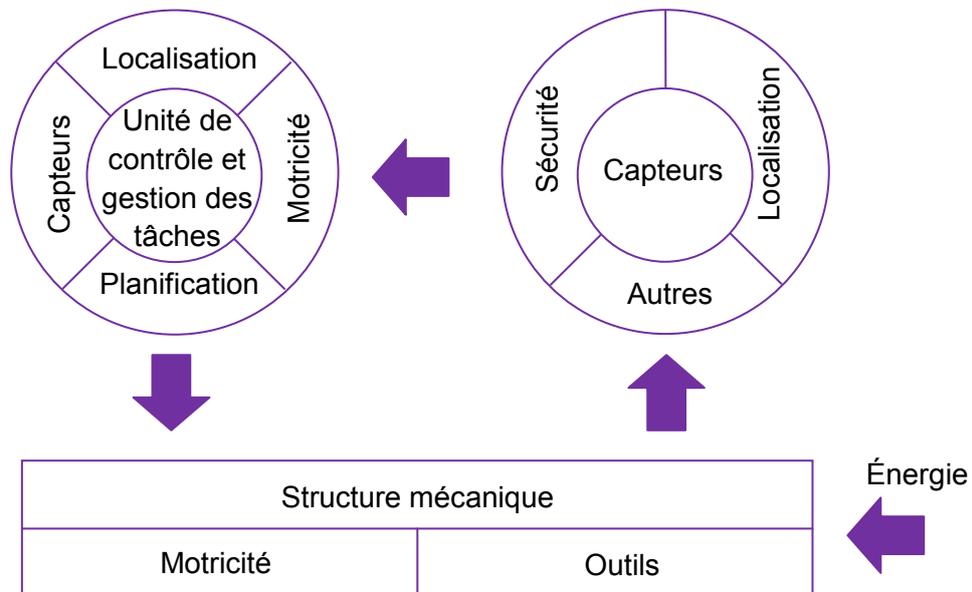


Figure 1.5. Architecture d'un robot mobile [2].

b. Classification des robots mobiles

Selon le système de locomotion, on distingue quatre types de robots mobiles :

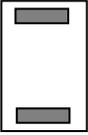
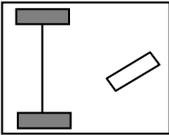
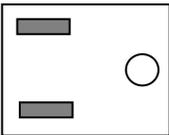
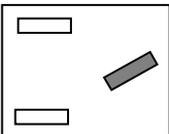
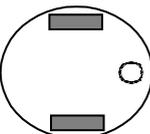
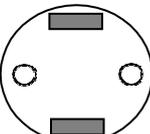
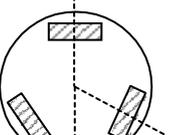
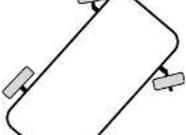
1. Robots mobiles à roues

La mobilité des roues est la structure mécanique la plus courante. Ce type de robot permet un mouvement avec accélération et vitesse tout en nécessitant un sol relativement plat. Il existe plusieurs catégories de robots à roues, déterminées essentiellement par la position et le nombre de roues utilisées.

- Robot mobile de type uni-cycle : Propulsé par deux roues indépendantes et disposant éventuellement d'un certain nombre de roues folles assurant sa stabilité.
- Robot mobile de type tricycle : Composé de deux roues fixes de même axe et d'une roue orientable centrée placée sur l'axe longitudinal du robot. Le mouvement est communiqué au robot par deux actions (vitesse longitudinale et orientation de la roue orientable).
- Robot mobile de type voiture : Un robot mobile de type voiture s'apparente à un tricycle, il est constitué de deux roues fixes placées sur le même axe et de deux roues orientables centrées également placées sur le même axe. Le robot mobile de type voiture est cependant plus stable puisqu'il dispose d'un point d'appui supplémentaire. Toutes les autres propriétés de la voiture robot sont identiques au tricycle robot, le second pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ce de manière à laisser le centre de rotation inchangé.

- Robot mobile omnidirectionnel : Un robot mobile omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est généralement constitué de trois roues déportées orientables disposées en triangle équilatéral. L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais cela se fait au prix d'une complexité mécanique beaucoup plus grande.

Tableau 1.1 : Classes de robots mobiles à roues.

	<p>Uni-cycle, avec deux roues motrices différentielles (Exemple : Cyepersonale robot).</p>
	<p>Tricycle, deux roues motrices en arrière, et une roué dirigée à l'avant (Exemple : Piaggiominitrucks).</p>
	<p>Uni-cycle, deux roues motrices différentielles en arrière, et une roue libre à l'avant (Exemple : Pygmalion et Alice).</p>
	<p>Tricycle, deux roues libres en arrière, une roue de traction dirigée à l'avant (Exemple : Neptune, Hero-1).</p>
	<p>Uni-cycle, deux roues motrices différentielles centrées, et un point de contact a l'avant (Exemple : Nomad Scout, smartRob).</p>
	<p>Uni-cycle, deux roues motrices différentielles centrées, et deux points de contact avant et arrière (Exemple: EPFLKhepera, HyperbotChip).</p>
	<p>Omnidirectionnel, trois roues suédoises motorisées (Exemple : Tribolo EPFL, Palm PilotRobotKit).</p>
	<p>Voiture - stable et complexité, - mécanique modérée.</p>

2. Robots mobiles à chenilles

Les robots mobiles chenillés illustrés à la **figure 1.6** présentent l'avantage d'une bonne adhérence au sol et la capacité de franchir les obstacles. L'emploi de ces robots est orientée vers une utilisation sur des terrains accidentés ou de mauvaise qualité en termes d'adhérence (présence de boue, d'herbe, etc.).

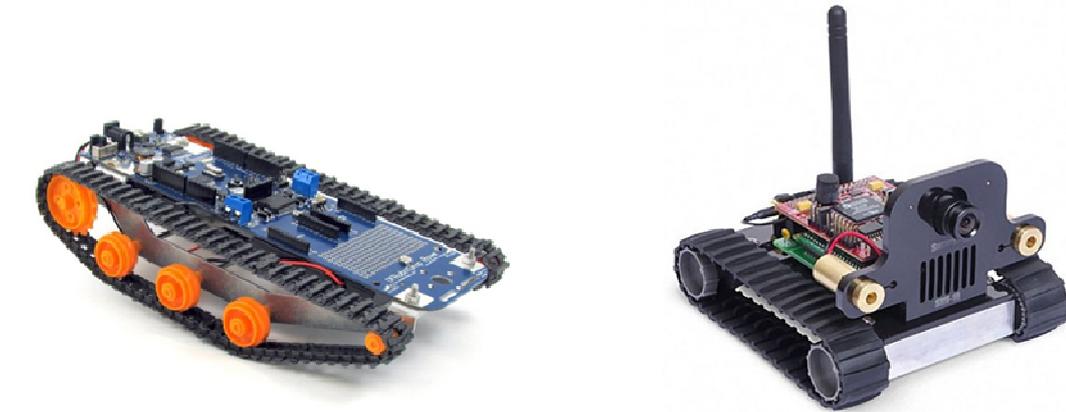


Figure 1.6. Robots mobiles à chenilles.

3. Robots mobiles marcheurs

Les robots mobiles marcheurs donnés à la **figure 1.7** sont destinés à effectuer diverses tâches dont l'accès à un site est difficile et dangereux pour l'être humain.



Figure 1.7. Robots mobiles marcheurs.

Leur structure est dotée de plusieurs degrés de liberté ce qui permet un rapprochement avec les robots manipulateurs. Une distinction est faite entre les robots marcheurs (figure à deux pattes (humanoïdes), à quatre pattes (type cheval) et à six pattes (type araignée)).

4. Robots mobiles rampants

Le rampant (crawling) est une solution de locomotion pour un environnement de type "tunnel" qui conduit à la réalisation de structures filiformes. Le système est composé d'un ensemble de modules ayant chacun plusieurs mobilités. Ici aussi les techniques utilisées dérivent des modes de locomotion des animaux et des insectes comme le montre la **figure (1.8)**.

Les applications de ce type de robot sont très spécialisées et ses architectures sont généralement spécifiques à l'application visée [15].

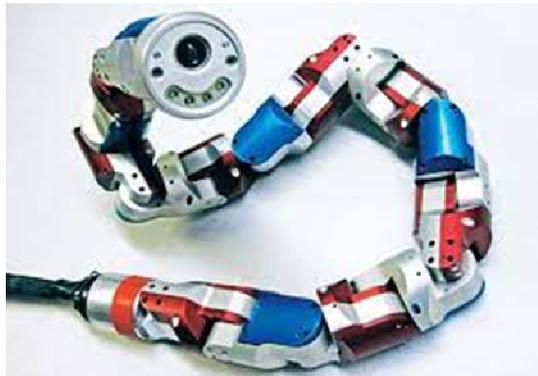


Figure 1.8. Robot mobile rampant.

1.6 Avantages et inconvénients des robots

1.6.1 Les avantages [16]

- La robotique et l'automatisation peuvent dans de nombreuses situations augmenter la productivité, la sécurité, l'efficacité, la qualité et la cohérence des produits.
- Les robots peuvent travailler dans un environnement dangereux, sans avoir besoin d'assistance vitale ou de problèmes de sécurité.
- Les robots n'ont pas besoin d'éclairage, de climatisation, de ventilation ou de protection contre le bruit.
- Les robots fonctionnent en continu, sans fatigue ni ennui, et ne nécessitent pas d'assurance médicale ou de vacances.
- Les robots sont d'une précision répétable à tout moment.
- Les robots peuvent être beaucoup plus précis que les êtres humains. La précision linéaire d'un robot est généralement de 10 à 20 microns.

- Ils s'améliorent constamment.

1.6.2 Les inconvénients

Ils n'ont pas la capacité de réagir à l'inattendu à moins que les situations ne soient comprises et que les réponses ne soient intégrées au système. Des mesures de sécurité sont nécessaires pour s'assurer qu'elles ne blessent pas les opérateurs ou n'endommagent pas les machines. On peut également citer d'autres inconvénients tels que [8] :

- Absence de prise de décision.
- Consommation d'énergie.
- Ils peuvent endommager d'autres appareils et blesser des personnes.
- Les robots sont chers en raison du coût initial de l'équipement, de l'installation, du besoin de périphériques, du besoin de formation et du besoin de programmation (les robots ont besoin d'humains).
- Les capacités du robot sont limitées.

1.7 Conclusion

L'utilisation de la robotique prend donc de plus en plus d'importance et les avancées technologiques et l'intelligence artificielle permettent toujours des améliorations, que ce soit dans l'architecture et la composition ou dans la fonctionnalité et l'autonomie.

Dans ce chapitre nous avons présenté une introduction aux deux principales catégories de robots, ainsi que d'autres spécificités de la robotique. Nous avons ensuite discuté de leurs avantages et inconvénients. Nous avons pu observer l'évolution actuelle des robots proposés par différents constructeurs, permettant de présenter une alternative aux machines-outils en utilisant une commande adéquate.

Le prochain chapitre suivant s'intéresse à la présentation des cartes Arduino et du système Android.

Chapitre 2 Présentation de la carte Arduino

2.1 Introduction

Avant d'aborder le troisième chapitre qui est le cœur de notre projet, nous allons présenter au cours de ce chapitre un ensemble d'éléments nécessaires à la fabrication de notre robot manipulateur.

Dans la section de conception matérielle, nous fournirons un ensemble d'informations sur la carte de contrôle Arduino, telles que sa définition, son principe de fonctionnement, le type et la série de la carte Arduino, etc. Le module Arduino Bluetooth, qui est l'outil de communication avec notre robot et ses caractéristiques, ainsi que le servomoteur et son fonctionnement sont également décrits.

Dans la section logicielle, nous présentons des logiciels tels que "IDE" pour programmer l'Arduino et "App Inventor" pour générer des applications Android.

Le circuit de commande de notre système est entièrement basé sur un circuit programmable appelé "Arduino", l'avantage de ce dernier est de minimiser la surface des cartes électroniques et donc de minimiser la quantité de composants électroniques utilisés.

2.2 Arduino

Arduino est une plateforme interactive de prototypage d'objets à usage créatif composée d'une carte électronique et d'un environnement de programmation. Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par 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 des relations homme/machine ou environnement/machine [17].

2.2.1 Début de Arduino

Le projet Arduino est né à l'hiver 2005. Massimo Banzi enseigne à l'école de design d'Ivrea (School of Design Ivrea) en Italie, où ses étudiants se plaignent souvent de solutions inefficaces en récompense de la réalisation de leurs projets robotiques. Banzi en a discuté avec David Cuartielles, ingénieur espagnol spécialisé dans les microcontrôleurs.

Le groupe décide de fabriquer leur propre carte en incluant dans leur histoire un des élèves de Banzi, David Mellis, qui sera chargé de concevoir le langage de programmation qui accompagne la carte. Dans deux jours David écrira le code, trois jours de plus et la carte était terminée. Ils ont décidé de l'appeler Arduino (un endroit fréquenté par des étudiants près de l'école).

2.2.2 Fonction de la carte Arduino

Arduino est un circuit imprimé open source. Ce mode de collaboration possède un fort potentiel et fait clairement avancer les choses. Sous ce nom Arduino se cachent non seulement du matériel mais aussi un logiciel. On parle alors de Physical Computing, qui désigne la construction de systèmes interactifs permettant de connecter le monde physique à celui des ordinateurs. Le monde réel est considéré comme un système analogique, alors que les ordinateurs agissent dans un environnement numérique ne connaissant que les états logiques 0 et 1. Il faut donc seulement établir une liaison entre ces deux mondes.

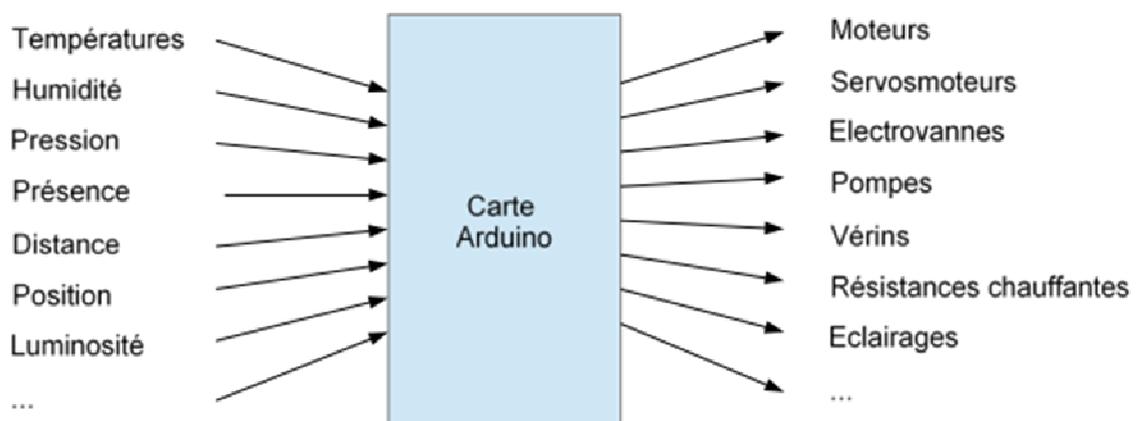


Figure 2.1. Interagir de la carte Arduino avec le monde réel.

2.2.3 Définition de la carte Arduino

La carte Arduino est basée sur un circuit intégré (un mini-ordinateur aussi appelé microcontrôleur) associé à des entrées et des sorties qui permettent à l'utilisateur de connecter différents types d'éléments externes. Les caractéristiques de la carte Arduino sont les suivantes :

- Plate-forme de développement et de prototypage Open Source.
- Stockage et exécution d'un programme.
- Shields (cartes d'extension) avec diverses fonctions qui s'affichent sur la carte Arduino.
- Relais, commande de moteurs, lecteur carte SD, etc.
- Ethernet, WIFI, GSM, GPS, etc.
- Afficheurs LCD, Écran TFT, etc.
- IDE (Environnement de Développement Intégré) multi OS.
- Édition du programme 1718 Arduino et système Androïde.
- Compilateur de programme.
- Transfert du programme dans la carte via le port USB.

2.2.4 Microcontrôleur ATmega328 de la carte Arduino

En générale, un microcontrôleur se divise en trois parties :

- Unité centrale (CPU),
- Mémoires (ROM et RAM),
- Ports d'entrée et de sortie.

En plus, un microcontrôleur est doté d'une horloge interne qui permet de piloter l'unité centrale. Les éléments qui composent un microcontrôleur sont comparables aux périphériques d'un ordinateur. La différence réside dans le fait que les trois parties citées précédemment sont intégrées au microcontrôleur. Elles se trouvent toutes dans le même boîtier, ce qui est plus simple et plus compact. La figure 2.2 montre un schéma fonctionnel de notre microcontrôleur [18].

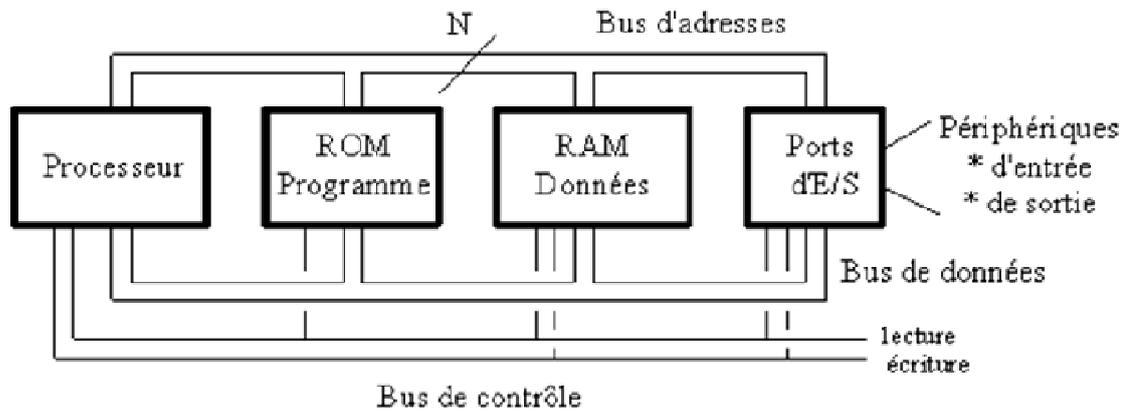


Figure 2.2. Schéma fonctionnel d'un microcontrôleur.

Le microcontrôleur présent sur notre carte Arduino est de type ATmega328. Ce dernier est montré sur la figure 2.2. Avec ses dimensions réduites, il dispose néanmoins d'une grande puissance de calcul. En fait, il suffit de le souder à une carte et de le mettre sous tension pour l'utiliser. Certains composants sont à rajouter comme par exemple, des stabilisateurs de tension, des connexions pour la programmation, et autres. Mais il est cependant sous cette forme déjà (presque) prêt à l'emploi [18].

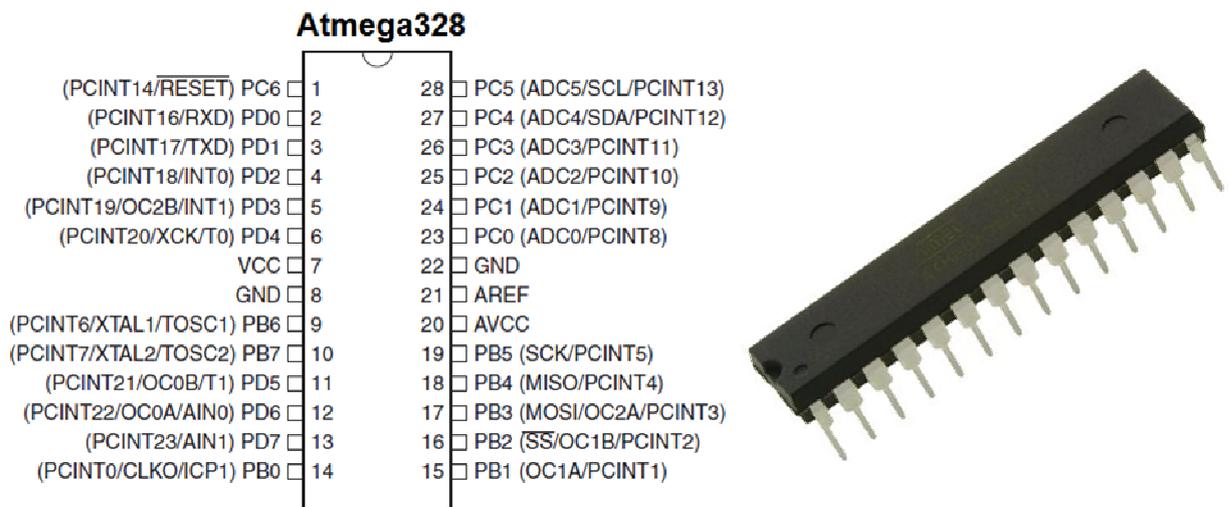


Figure 2.3. Microcontrôleur ATmega328.

2.3 Familles de la carte Arduino

Il existe trois familles de cartes Arduino selon le fabricant [19] :

1. Les "officiels" qui sont fabriqués en Italie par le fabricant officiel (Smart Projects).
2. Les "compatibles" qui ne sont pas fabriqués par (Smart Projects), mais qui sont entièrement compatibles avec les Arduino officiels.
3. Les "autres" fabriqués par diverses sociétés et commercialisés sous un nom différent (Freeduino, Seeduino, Femtoduino, ...).

2.4 Types de carte Arduino

Actuellement, il existe plus de 20 versions du module Arduino, parmi lesquelles on trouve [20] :

- L'Arduino NG, doté d'une interface USB pour la programmation et l'utilisation d'un microcontrôleur ATmega8.
- L'Arduino Mini, est une version miniature de l'Arduino utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, est une petite carte basée sur les microcontrôleurs comme ATmega328P (ou ATmega628).
- Le LilyPad Arduino, est une technologie e-textile portable développée par Leah «Buechley» et soigneusement conçue par «Leah et SparkFun».
- L'Arduino NG plus, doté d'une interface USB pour la programmation et l'utilisation d'un microcontrôleur ATmega168.
- L'Arduino Bluetooth, dépend principalement du microcontrôleur comme ATmega168 et cette carte est également appelée Arduino BT.
- L'Arduino Mega, utilise le microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilise le microcontrôleur ATmega328.
- L'Arduino Mega2560, utilise le microcontrôleur ATmega2560, et possède une mémoire de 256 KBS. Il incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, utilise un microcontrôleur avec l'USB qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Esplora : ressemble un contrôleur visuel de jeu avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Parmi ces types, nous avons choisi une carte Arduino UNO (carte de base). L'intérêt principal de cette carte est de faciliter la mise en œuvre d'une telle commande qui sera détaillée plus loin.

2.5 Choix d'Arduino

Il existe de nombreux microcontrôleurs disponibles sur le marché tels que BX-24 Phidgets, Basic Stamp et Parallax de Netmedia Raspberry Pi et tous ont de puissantes capacités et ont la capacité de contrôler différents composants électroniques. Bien sûr, avec un taux de préférence différent, mais que distinguer l'Arduino est un ensemble de choses qui font la différence entre lui et les autres le plus important :

Le choix d'Arduino repose sur les points suivants :

- **Simplicité** : la carte Arduino est conçue pour répondre aux besoins de tout le monde (chercheurs, enseignants, étudiants et même amateurs).
- **Prix** : Arduino est moins cher que les autres cartes du même type.
- **Auto assemblage** : nous pouvons télécharger gratuitement la fiche technique privée Arduino sur le site officiel et acheter les pièces et l'installer nous-mêmes.
- **Multiplateforme** : Le programme Arduino a la capacité de fonctionner sous Windows, Mac OS, Linux.
- **Un environnement logiciel simple et convivial** : l'environnement de programmation conçu pour être facile pour les débutants.
- **Open Source** :
 - ✓ Le logiciel Open Source est écrit en C ++ et il est de plus téléchargeable. Les programmeurs peuvent le modifier en fonction de leurs besoins.
 - ✓ le matériel Open Source est principalement constitué de microcontrôleurs ATmega8 et ATmega168 et les schémas sont publiés sous une licence Créative Commons, permettant aux concepteurs de circuits électroniques de concevoir leurs propres circuits [21].

2.6 Arduino UNO

La version de base du système est basée sur un microcontrôleur Atmega328. Cette carte délivre un maximum de 14 points d'entrée/sortie digitaux et de 6 entrées analogiques. Elle existe avec un processeur en version DIL sur support ou SMD soudée. Outre le processeur, la principale évolution consiste en un remplacement du

chipset de l'interface de programmation USB de la puce d'interface USB/Série FTDI en une puce Atmel8U2 ou sur les dernières versions 16U2. Le tableau ci-dessous présente les principales caractéristiques de la carte Arduino Uno [21].

Tableau 2.1 : Synthèse des caractéristiques [21].

Catégorie	Valeur
Microcontrôleur	ATmega328
Fréquence d'horloge	16MHz
Tension de service	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limites)	6-20V
Ports numériques	14 entrées/sorties
Ports analogiques	6 entrées
Courants maxi par broche d'E/S (cc)	40mA
Courant maxi par broche 3.3V	50mA
Mémoire	32ko Flash, 2ko SRAM, 1ko EEPROM
Interface	USB

2.6.1 Composition de carte Arduino UNO

Elle se compose de :

- Microcontrôleur ATmega.
- Tension de service 5 V.
- 14 entrées et sorties numériques (6 sorties commutables en MLI).
- 6 entrées analogiques (résolution 10 bits).
- 32 Ko de mémoire flash (0,5 Ko occupé par le chargeur d'amorçage ou bootlader).
- 2 Ko de SRAM.
- 1 Ko d'EEPROM.
- Fréquence d'horloge 16 MHz.
- Interface USB.

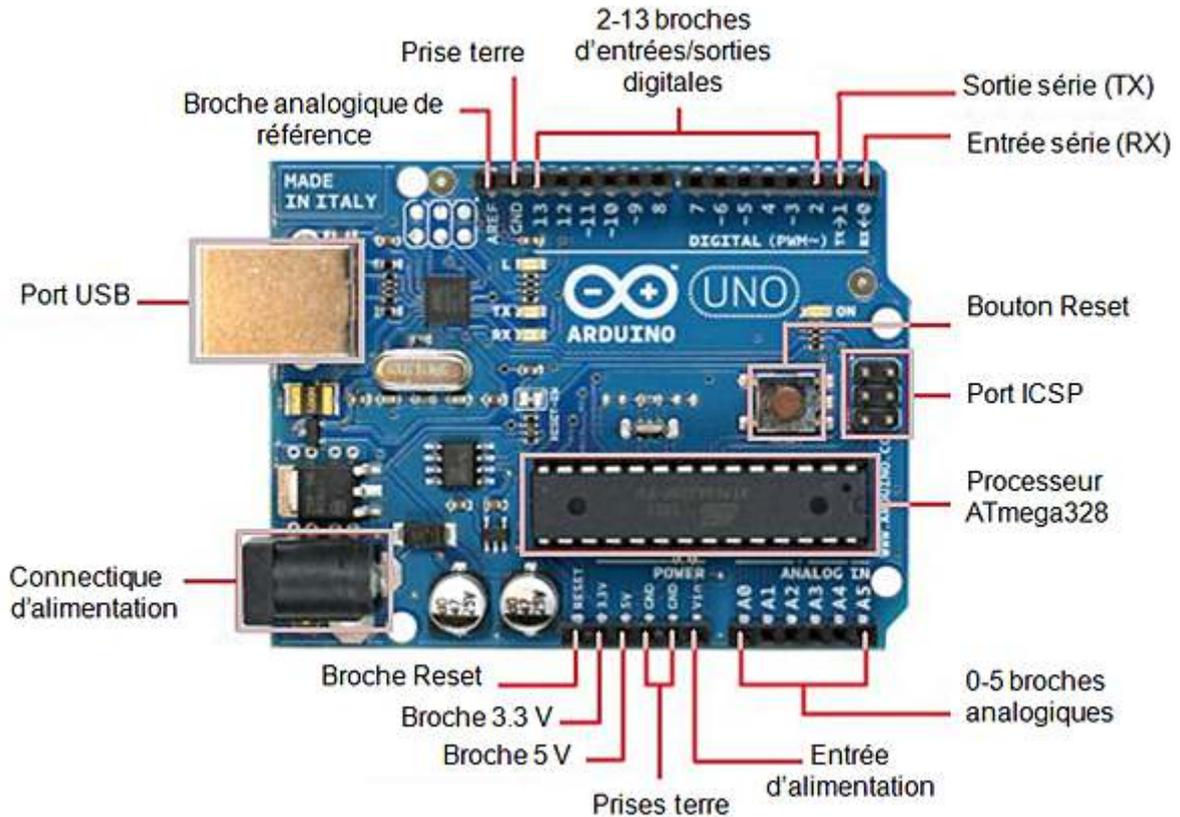


Figure 2.4. Carte Arduino UNO.

a. Alimentation de la carte Arduino

La carte Arduino Uno peut être alimentée soit via la connexion USB (qui fournit du 5V jusqu'à 500mA) soit à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non USB) peut être soit un adaptateur secteur (généralement capable de fournir 3V à 12V à 500mA) soit des piles (ou des batteries rechargeables). L'adaptateur secteur peut être connecté en branchant une fiche centrale positive de 2,1 mm dans le connecteur jack de la carte. Les fils d'une batterie ou d'un bloc-batterie peuvent être insérés dans les connecteurs à broches de la carte étiquetés GND (masse ou 0 V) et Vin (tension d'entrée positive) du connecteur d'alimentation. Cependant, si la carte est alimentée avec moins de 7 V, la broche 5 V peut fournir moins de 5 V et la carte peut être instable. Si plus de 12V est utilisé, le régulateur de tension de la carte pourrait chauffer et endommager la carte. De plus, la plage idéale recommandée pour alimenter la carte Uno se situe entre 7V et 12V [21].

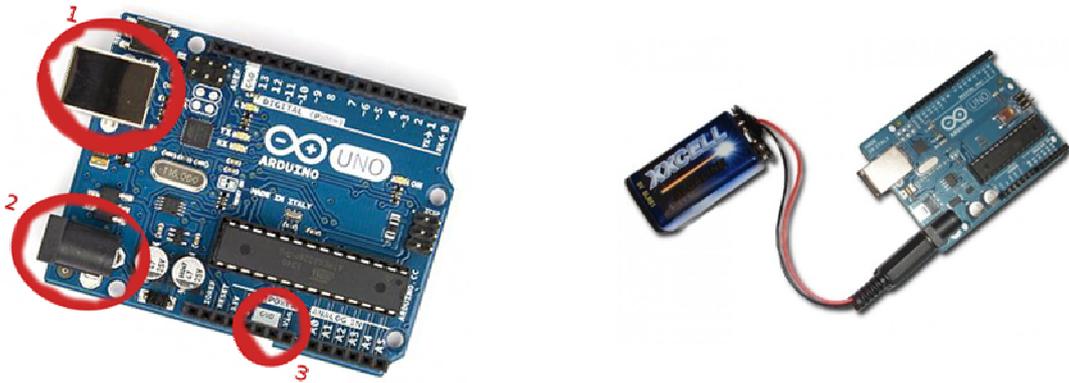


Figure 2.5.Alimentation de la carte Arduino UNO.

b. Programmation

L'environnement de développement interactif ou IDE est un programmeur utilisé pour compiler et interpréter des codes pour développer des programmes utilisés sur un plateau Arduino. IDE est une application écrite en Java inspirée du langage Processing. L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

```

sketch_apr21a | Arduino 1.8.13
Fichier Édition Croquis Outils Aide
sketch_apr21a $
void setup() {
  // put your setup code here, to run once:
  pinMode(13, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);}
}

```

Figure 2.6.Environnement de développement interactif pour la carte Arduino.

Le programme est lu par le microcontrôleur de haut en bas. Une variable doit être déclarée avant d'être utilisée par une fonction. La structure minimale se compose de :

- Un en-tête : déclaration des variables, des constantes, indication de l'utilisation des bibliothèques etc.
- Un setup (configuration) : cette partie est lue une seule fois, elle comprend les fonctions à réaliser au démarrage (utilisation des broches comme entrées ou sorties, mise sous tension du midi, du port série de l'I2C etc.).
- Une loop (boucle) : cette partie est lue en boucle, c'est ici que les fonctions sont exécutées. En plus de cette structure minimale, on peut ajouter des "sous-programmes" ou "routines" qui peuvent être appelés à tout moment dans la boucle, très utiles pour produire des morceaux de code répétitifs.
- Des « Callbacks », ce sont des fonctions qui sont rappelées automatiquement depuis une bibliothèque (ou bibliothèque).

2.7 Bluetooth

Bluetooth est une norme de télécommunication permettant l'échange bidirectionnel de données sur de courtes distances en utilisant les ondes radio UHF sur la bande de fréquence 2,4 GHz. Son but est de simplifier les connexions entre les appareils électroniques à proximité en éliminant les connexions filaires. Il peut remplacer, par exemple, les câbles entre les ordinateurs, les tablettes, les haut-parleurs, les téléphones portables entre eux ou avec les imprimantes, les scanners, les claviers, les souris, les contrôleurs de jeux vidéo, les téléphones portables, les assistants personnels, les systèmes mains libres pour microphones ou écouteurs, les autoradios, appareils photo numériques, lecteurs de codes-barres et bornes publicitaires interactives.



Figure 2.7. Logo de Bluetooth.

Le nom "Bluetooth" est directement inspiré du surnom anglicisé du roi viking danois Harald à la dent bleue (en danois Harald Blåtand, en anglais Harald Bluetooth). Ce nom a été proposé en 1996 par Jim Kardach d'Intel, un ingénieur travaillant alors sur le développement d'un système qui permettrait aux téléphones portables de communiquer avec les ordinateurs [23][24].

2.7.1 Module Bluetooth Arduino

Le module Bluetooth Arduino est une plate-forme Arduino populaire avec une connexion série Bluetooth au lieu d'une connexion USB. Il a une très faible consommation d'énergie, une très courte portée (sur un rayon d'une dizaine de mètres), un faible débit et il est compact. Il existe deux types de module Bluetooth : HC-05 et HC-06. Tous les deux sont compatibles avec Arduino et les deux modules peuvent être utilisés en mode COMMAND, pour les programmer avec des "commandes AT", ou en mode DATA, pour échanger des données [25].

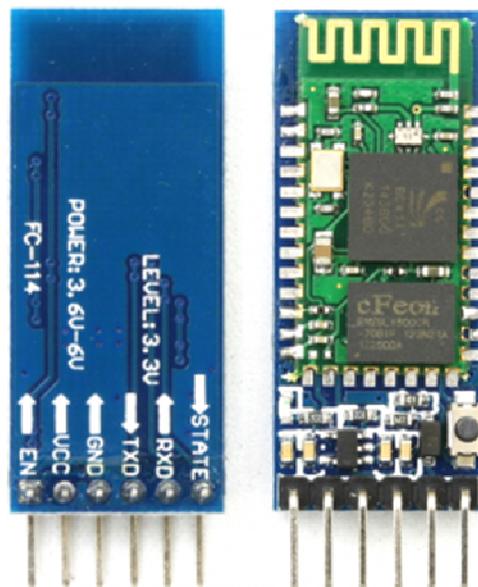


Figure 2.8. Module Bluetooth Arduino.

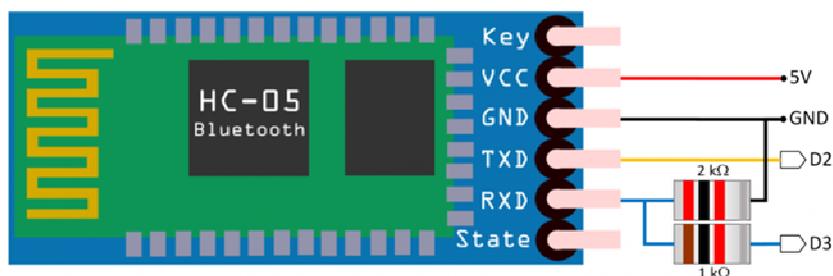


Figure 2.9. Caractéristique et alimentation du module Bluetooth.

2.7.2 Différence entre HC-05 et HC-06

La différence entre le HC-05 et le HC-06 est que le HC-05 a un micrologiciel "complet" doté de nombreuses commandes AT, et peut être à la fois maître et esclave. En revanche, le firmware du HC-06 ne peut être qu'esclave, avec des commandes AT très limitées.

En d'autres termes:

- Le module HC-05 peut établir une connexion avec d'autres modules. Par exemple, un robot étant maître et se connectant au module Bluetooth esclave, ou en mode esclave pour créer une connexion sans fil avec un ordinateur portable.
- Le module HC-06 ne peut être qu'esclave. Cela le rend uniquement utilisé pour connecter un ordinateur portable en tant que maître à un robot avec un module esclave, par exemple, pour un pont série sans fil [20].

Pour notre travail, nous avons utilisé le module Bluetooth HC-06.

2.8 Servomoteur

Un servomoteur est un système qui vise à produire un mouvement précis en réponse à une commande extérieure. C'est un actionneur (système produisant une action) qui intègre électronique, mécanique et automatisme. Un servomoteur est capable d'attendre des positions prédéterminées dans les consignes qui lui sont données, puis de les maintenir [26].

Un servomoteur est composé de :

- Un micro moteur à courant continu.
- Un réducteur en sortie de ce moteur diminuant la vitesse et augmentant le couple.
- Un potentiomètre (agissant comme un diviseur résistif) qui génère une tension variable, proportionnelle à l'angle de l'axe de sortie.
- Un axe dépassant hors du boîtier avec différents bras ou roues de fixation.
- Un dispositif électronique d'asservissement.

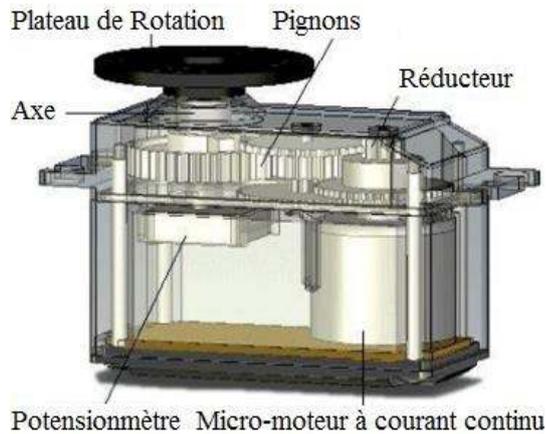


Figure 2.10. Composition d'un servomoteur [26].

Dans l'univers Arduino, le servomoteur est principalement utilisé dans les applications robotiques. Le servomoteur possède trois fils de connexion pour pouvoir fonctionner, deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande :

- Rouge : pour l'alimentation positive (4.5V à 6V en général).
- Noir ou marron : pour la masse (0V).
- Orange, jaune, blanc, ... : entrée du signal de commande [26].

2.8.1 Fonctionnement des servomoteurs

C'est un petit circuit électronique permet de contrôler un moteur à courant continu en fonction de la position d'un potentiomètre intégré au servomoteur. La sortie du moteur à courant continu est mécaniquement liée à une série d'engrenages qui augmente la force (couple) du servomoteur en réduisant sa vitesse de rotation. Lorsque le moteur tourne, les engrenages s'animent, le bras bouge et entraîne avec lui le potentiomètre. Le circuit électronique ajuste en permanence la vitesse du moteur afin que le potentiomètre (et par extension le bras) reste toujours au même endroit.

Le servomoteur à l'avantage d'être asservi en position angulaire, cela signifie que l'axe de sortie du servomoteur respectera la consigne d'instruction envoyée à son entrée, même si un obstacle intervient pour faire changer l'orientation de sa trajectoire, le servomoteur essaiera de conserver la position. Pour un ajustement précis de la position, le moteur et son réglage sont équipés d'un système de mesures qui détermine la position courante.

Il suffit de donner une consigne au servomoteur ("reste à 45°" par exemple) et le servomoteur fera son maximum pour rester au plus près de cette consigne.

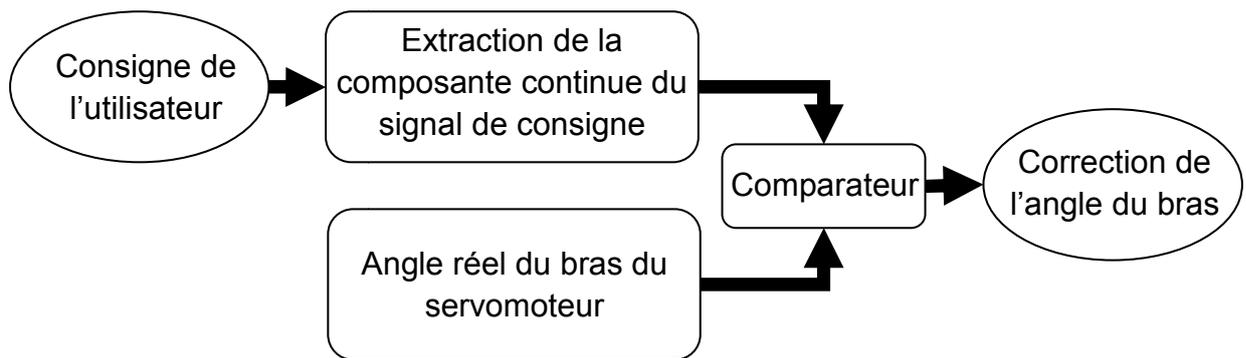


Figure 2.11. Fonctionnement d'un servomoteur.

Cette consigne est transmise au moyen d'un signal numérique (plus précisément une impulsion). Pour que le servomoteur reste à une position donnée, une impulsion d'une durée comprise entre 1 et 2 millisecondes doit être émise toutes les 20 millisecondes (soit à une fréquence de 50 Hz).

- Une impulsion de 1 ms correspond à un angle de 0°.
- Une impulsion de 2 ms correspond à un angle de 180°.
- Une impulsion de 1,5 ms correspond à un angle de 90°.

La plupart des servomoteurs fonctionnent sur 5 volts, mais certains fonctionnent sur 3.3 volts.

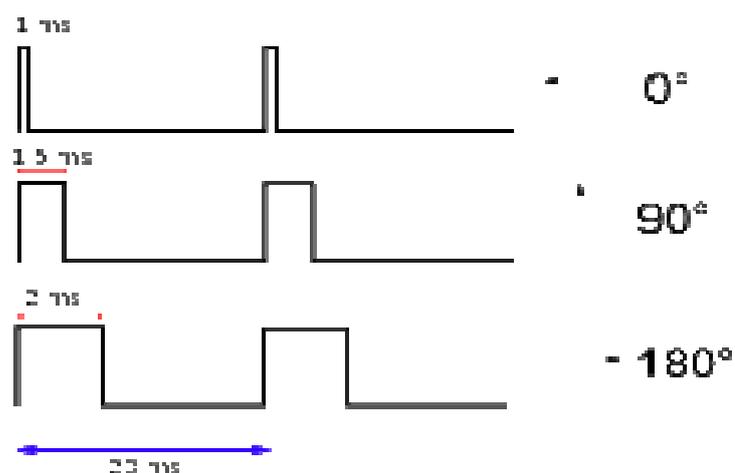


Figure 2.12. Signal de commande servomoteur.

2.8.2 Avantages et inconvénients [15]

a. *Avantages*

- Aucune connaissance en programmation n'est requise dans la plupart des cas.
- Projet pour un petit budget.
- Projet de base dans lequel la précision n'est pas importante.
- Robot de petite taille (les servomoteurs intelligents ne sont pas encore disponibles en micro format).

b. *Inconvénients*

- Communication unidirectionnelle.
- Impossible d'obtenir de rétroaction sur la position réelle.
- Servomoteurs standards connectés à un contrôleur nécessitent beaucoup de câblage.

2.9 Shield moteur L293D

Le composant L293D est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur. Ce composant se trouve facilement sur et ne coûte pas très cher.

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux. Il est également possible de constituer deux pont-H afin de piloter deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.

Il est important de noter que le L293D peut délivrer un maximum de 600mA, ce que le choix des moteurs doit donc être fait en conséquence.

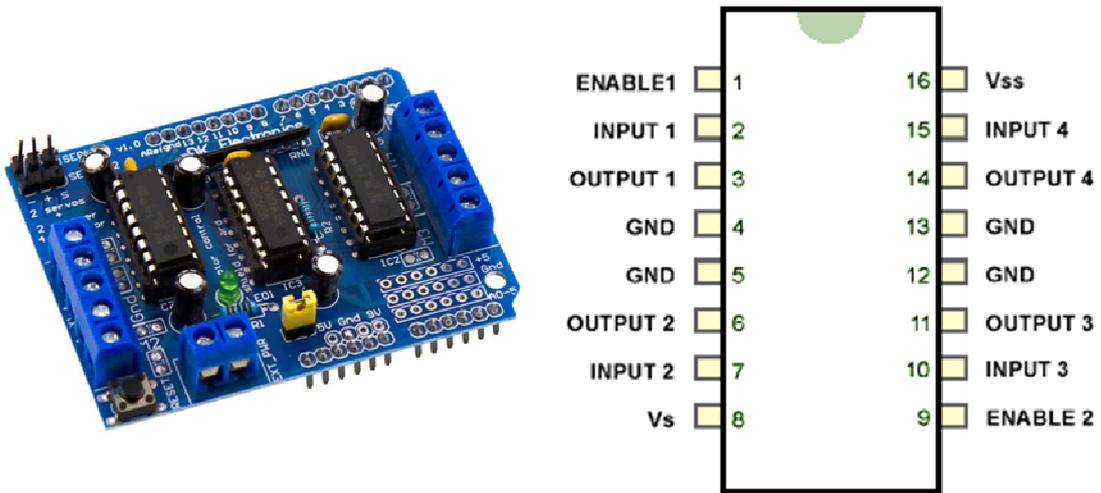


Figure 2.13.Shield moteur L2934.

Les caractéristiques techniques du L293D sont :

- Nombre de pont-H : 2
- Courant Max en régime continu : 600mA (x2)
- Courant de pointe Max < 2ms : 1200mA
- VS Max Alim moteur : 36v
- VSS Max Alim logique : 7v
- Nombre de broches : 16 DIP
- Perte de tension : 1.3v

2.10 Capteur à ultrason HC-SR04

Le capteur HC-SR04 utilise des ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec une grande précision et des mesures stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou les matériaux sombres, bien que des matériaux tels que les vêtements puissent être difficiles à détecter.

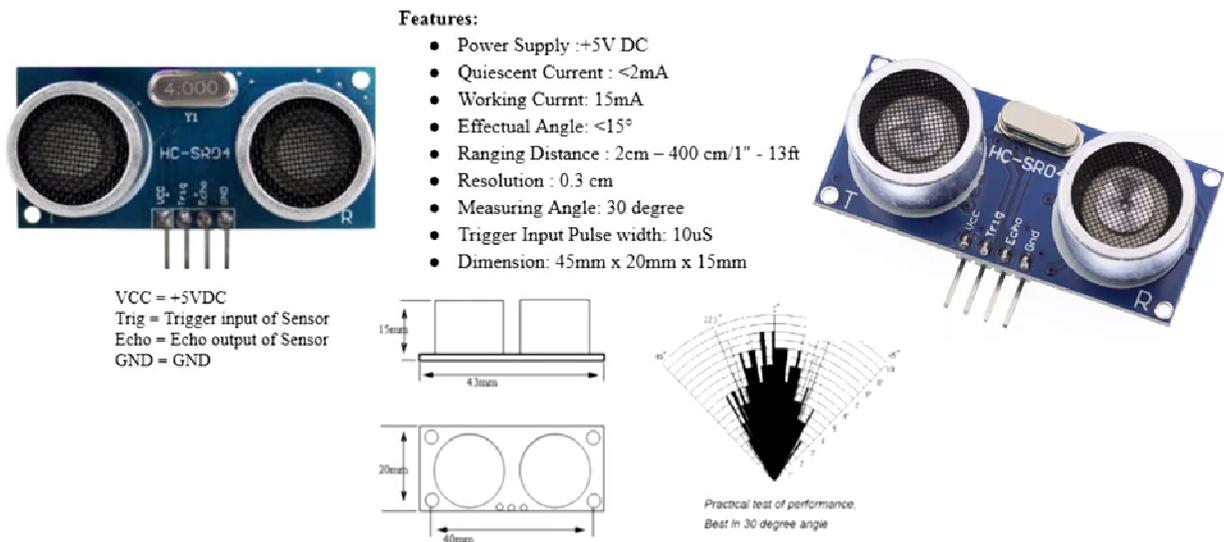


Figure 2.14. Capteur ultrason.

2.10.1 Caractéristique du HC-SR04

- Plage de mesure : 2 cm à 400 cm.
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : -7.5° à $+7.5^\circ$.
- Signal de détection : impulsion de 10us TTL.
- Signal d'écho : sortie TTL PWL.
- Dimensions : 45 mm x 20 mm x 15 mm.

2.10.2 Configuration des broches du HC-SR04

- Vcc = Alimentation +5 V CC.
- Trig = Entrée de déclenchement de la mesure (Trigger input).
- Echo = Sortie de mesure donnée en écho (Echo output).
- GND = Masse de l'alimentation.

2.11 Mini moteur et roues

2.11.1 Mini moteur motoréducteur 6V

Il s'agit d'un micro motoréducteur (moteur + réducteur) 6V, à 180 tours/min avec transmission d'axe dans lequel l'axe de l'arbre de sortie s'étend parallèlement à l'axe

du moyen moteur. Ce moteur, de petite taille, est simple à installer. Il est très approprié pour construire une plate-forme de 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.



Figure 2.15. Mini moteur motoréducteur.

2.11.2 Roues

Pour notre application, nous avons choisi des roues de marque Tamiya qui s'adaptent mieux aux moteurs utilisés.

Caractéristiques

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



Figure 2.16. La roue Tamiya.

2.12 Conclusion

Dans ce chapitre, nous avons fait une présentation de la carte de commande l'Arduino. Nous avons évoqué les différentes familles et les différents types de cette dernière tout en donnant les raisons pour lesquelles elle a été choisie. Ensuite, nous avons présenté le matériel dont nous avons besoin pour notre réalisation (servomoteur, capteur Bluetooth, l'ultrason, etc.). Des définitions des caractéristiques techniques de la carte Arduino a fait aussi l'objet de ce chapitre.

Le chapitre suivant sera consacré à la réalisation de notre robot, en commençant par la présentation du logiciel de travail (programmation et simulation), la programmation du robot et de l'application mobile, pour finir par la réalisation et l'assemblage.

Chapitre 3 Simulation et réalisation du robot :

partie matérielle et logicielle

3.1 Introduction

Dans ce chapitre, nous allons présenter les différentes étapes de réalisation de notre robot. La fabrication commence par une structure mécanique avec quatre moteurs à courant continu, ainsi que l'assemblage mécanique et électrique de la carte avec les servomoteurs et le module Bluetooth HC-05. Ensuite le développement d'une application Android basée sur le logiciel MIT APP INVENTOR, qui permet de connecter et contrôler notre robot via tous les téléphones Android.

3.2 Élaboration du prototype

Le robot à commande vocale Arduino est interfacé avec un module Bluetooth HC-05. Nous pouvons donner des commandes vocales spécifiques au robot via une application Android installée sur le téléphone. Côté réception, un module émetteur-récepteur Bluetooth reçoit les commandes et les transmet à l'Arduino, ce qui permet de contrôler le robot.

Le robot à commande vocale sans fil d'Arduino se compose de deux parties: émission et réception. La partie émission est composée du Bluetooth et de l'application Android du Smartphone. Du côté réception, la partie comprend une carte ArduinoUNO en tant que processeur, un module Bluetooth HC-05 en tant que module de communication sans fil, un circuit intégré L293D pour piloter les moteurs d'entraînement, une paire de moteurs à engrenages CC 200 tr/min conçus pour le déplacement du robot et d'une batterie 9V.

Le schéma fonctionnel du robot à commande vocale sans fil Arduino est illustré ci-dessous.

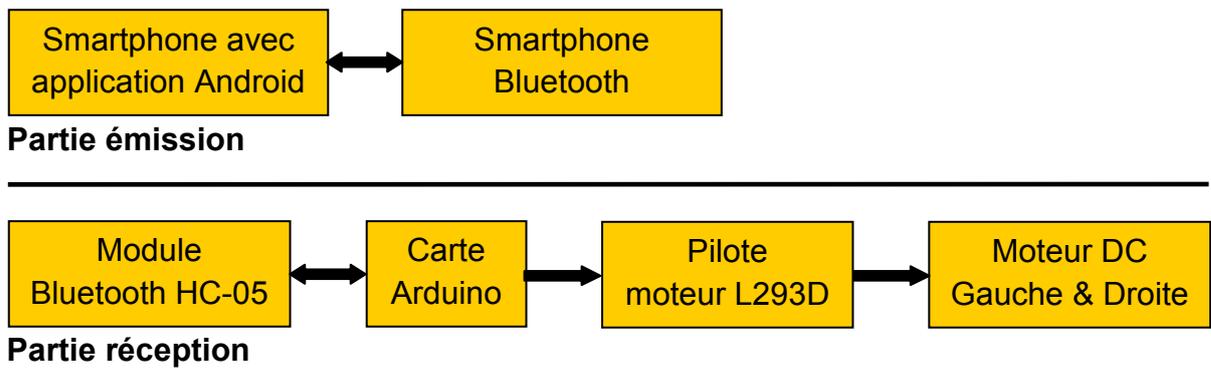


Figure 3.1. Schéma général du prototype.

Au début, le Bluetooth HC-05 doit être couplé ensuite connecté. Une fois l'application Android est en cours d'exécution sur le Smartphone, les commandes vocales de l'utilisateur sont captées par le microphone du Smartphone. Ces commandes sont traitées et la conversion parole-texte est effectuée dans l'application à l'aide de la technologie de reconnaissance vocale de Google. Le texte est ensuite envoyé au récepteur via Bluetooth.

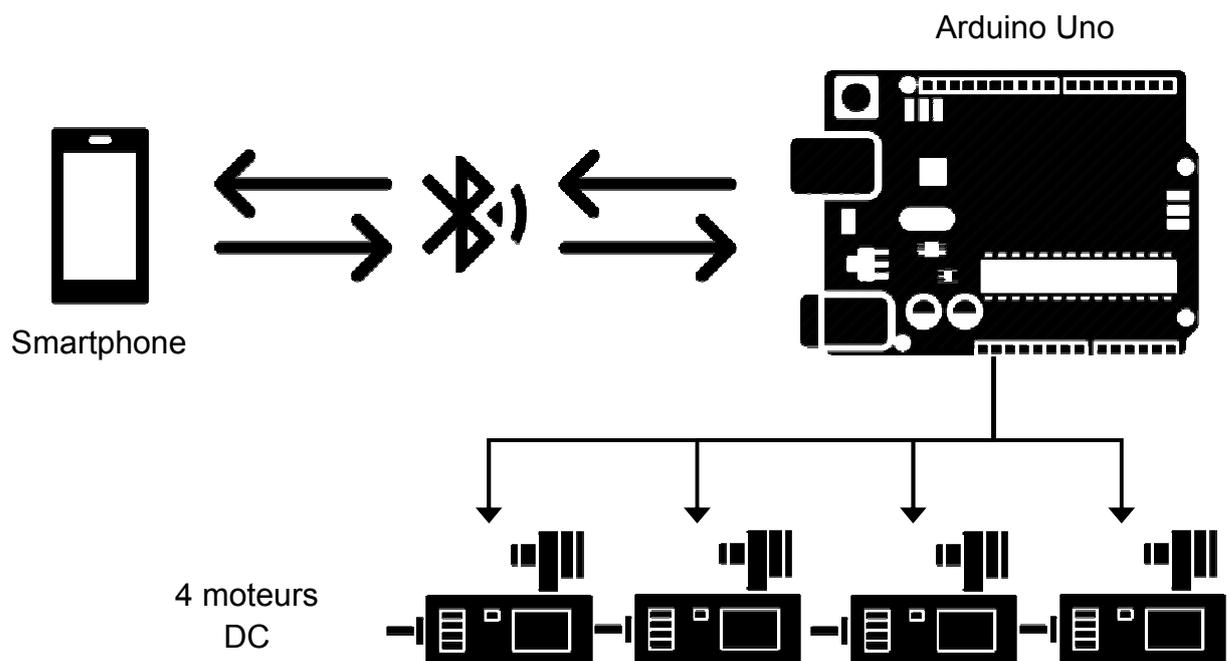


Figure 3.2. Graphique synoptique du prototype.

Le graphique synoptique est donné par la figure 3.2. Les broches TX, RX d'Arduino sont connectées aux broches Rx, Tx du module Bluetooth. Le module Bluetooth est alimenté en 5V. De même, le moteur CC gauche est connecté aux broches 3 et 6 de L293D et le moteur CC droit aux broches 14 et 11 de L293D. Les broches numériques Arduino 2, 3, 4, 5 sont connectées respectivement aux L293D 2, 7, 10, 15.

Les broches 2, 5, 12, 13 du circuit intégré L293D sont des broches GND et 9, 1, 16 sont 5V. Mais la broche 8 de L293D est directement alimentée en 9V.

3.3 Lesystème d'exploitationAndroid

3.3.1 Définition

Android est une plate-forme complète pour les appareils mobiles (téléphone, PDA, netbook, tablettes, etc.). La plateforme Android est un OS (Operating System) basé sur un noyau Linux, sous licence open source. Il se compose d'un système d'exploitation, de bibliothèques (middleware) et d'un ensemble d'applications : un client de messagerie, un navigateur, un agenda, etc.

Les services proposés par Android facilitent l'exploitation des réseaux de télécommunications GSM, Bluetooth, WIFI et UMTS, la manipulation des médias, notamment vidéo, audio et images JPEG ainsi que d'autres formats, l'exploitation des capteurs tels que les capteurs de mouvement, appareil photo, boussole et récepteur GPS, fonctionnement de l'écran tactile, stockage de la base de données, affichage de la page Web, application multitâche et envoi de messages.

3.3.2 Logo

Le personnage nommé Bugdroid est le petit robot vert utilisé par Google pour présenter Android. Ce personnage est sous licence « Creative Commons by (3.0) » et peut donc être utilisé librement.

Le site Engadget annonce que Bugdroid, le logo Android, est en fait un personnage d'un jeu Atari des années 1990 : Gauntlet : The Third Encounter.

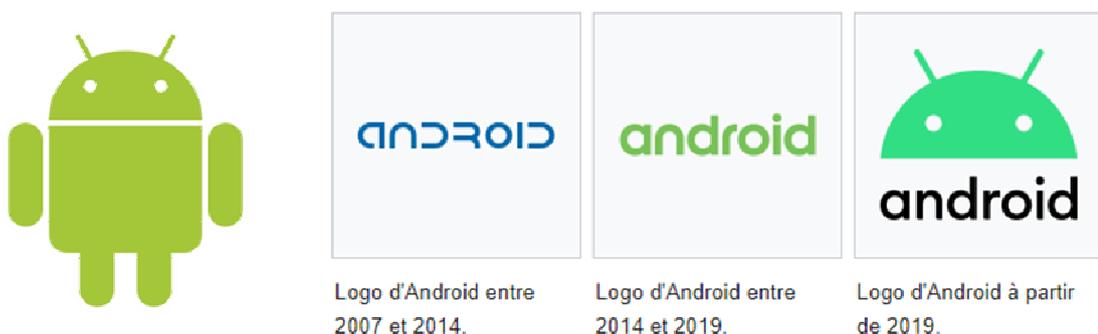


Figure 3.3.Logo Android.

3.3.3 L'architecture d'Android

L'environnement de développement s'appuie sur une architecture basée sur le noyau Linux. La plate-forme Android est composée de cinq couches principales, ce sont le noyau Linux, les bibliothèques et runtime Android, le framework d'application et la couche application.

Android bénéficie d'une architecture en couches complètes qui en fait une plate-forme riche, dédiée aux appareils mobiles. Il est basé sur le noyau Linux qui permet de faire le pont entre la partie matérielle et la partie logicielle et de profiter des services système de base tels que la sécurité, la gestion de la mémoire, la gestion des processus, etc.

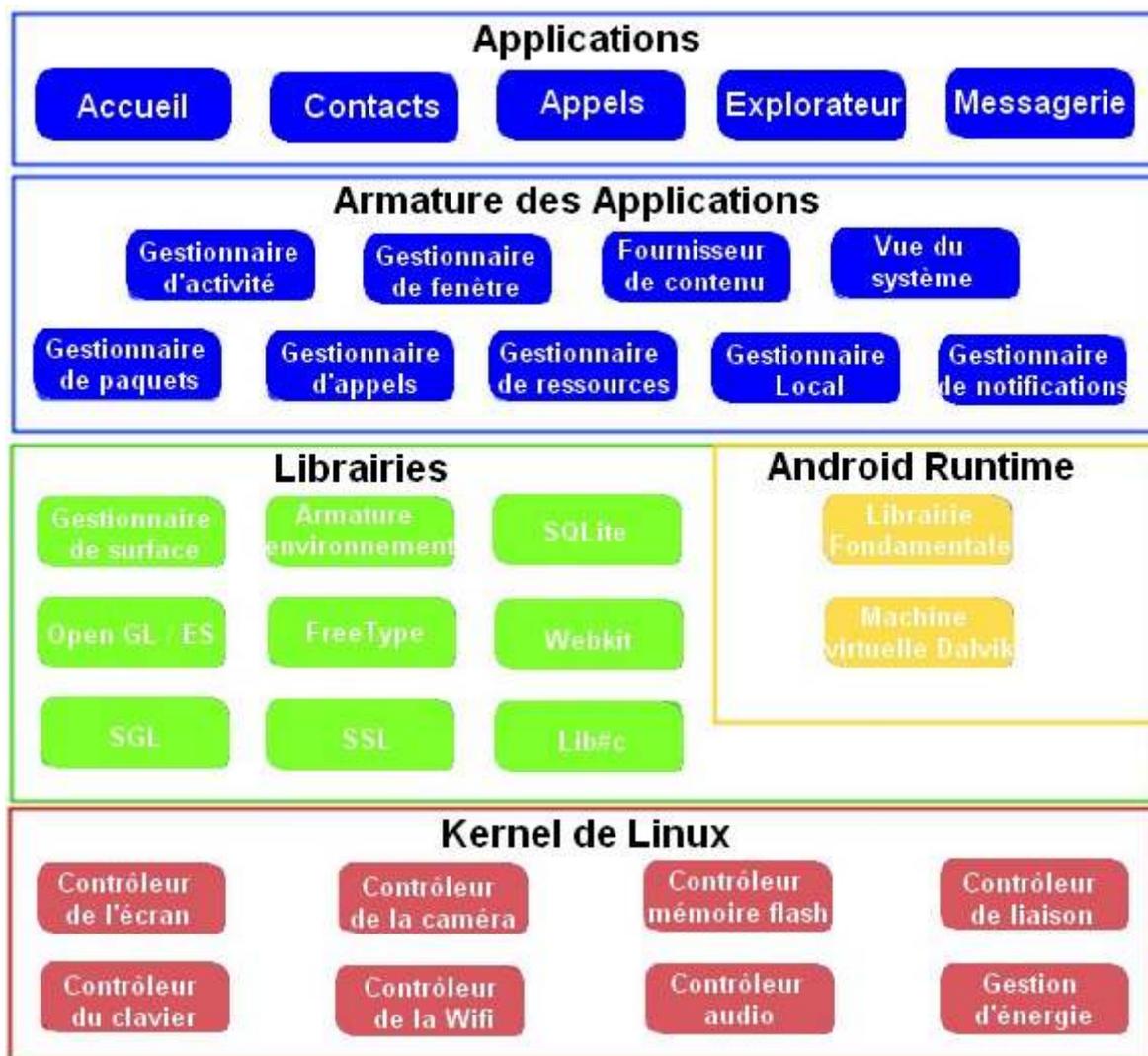


Figure 3.4. Architecture d'Android.

C'est sur cette couche que vient se greffer le Runtime Android, incluant la machine virtuelle java et ses bibliothèques. Ensuite elle vient la plate-forme logicielle, également appelée Development Framework, écrite en java et permettant la mutualisation des ressources entre les applications Java. Elle permet aux développeurs de produire des applications diverses et innovantes à l'aide d'un ensemble d'API.

Enfin, et à un niveau supérieur se trouve un ensemble d'applications sous forme de packages apk. Les applications fournies par Android sont telles qu'un navigateur Web, un client de messagerie, un calendrier, un gestionnaire de contacts, etc.

3.4 Présentation de MIT APP INVENTOR

APP INVENTOR est une application web développée par Google Labs. Cette application a été reprise par le MIT (Massachusetts Institute of Technology) après avoir été abandonnée par Google. Elle permet le développement d'applications Android basées sur l'assemblage de blocs (langage Scratch). Le programme s'effectue en ligne, à l'aide de n'importe quel navigateur et sous différents environnements logiciels (Mac, Linux ou Windows) [27].

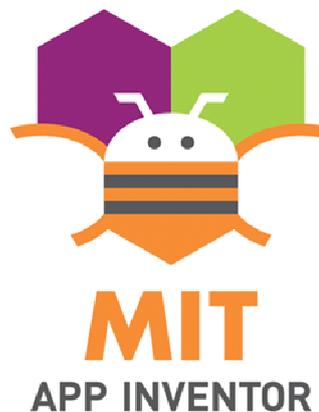


Figure 3.5. Logo APP INVENTOR.

3.4.1 Conception d'APP INVENTOR

L'application se compose de trois fenêtres, l'une pour concevoir l'interface graphique, la seconde pour assembler les blocs de programmation et la dernière pour l'émulateur :

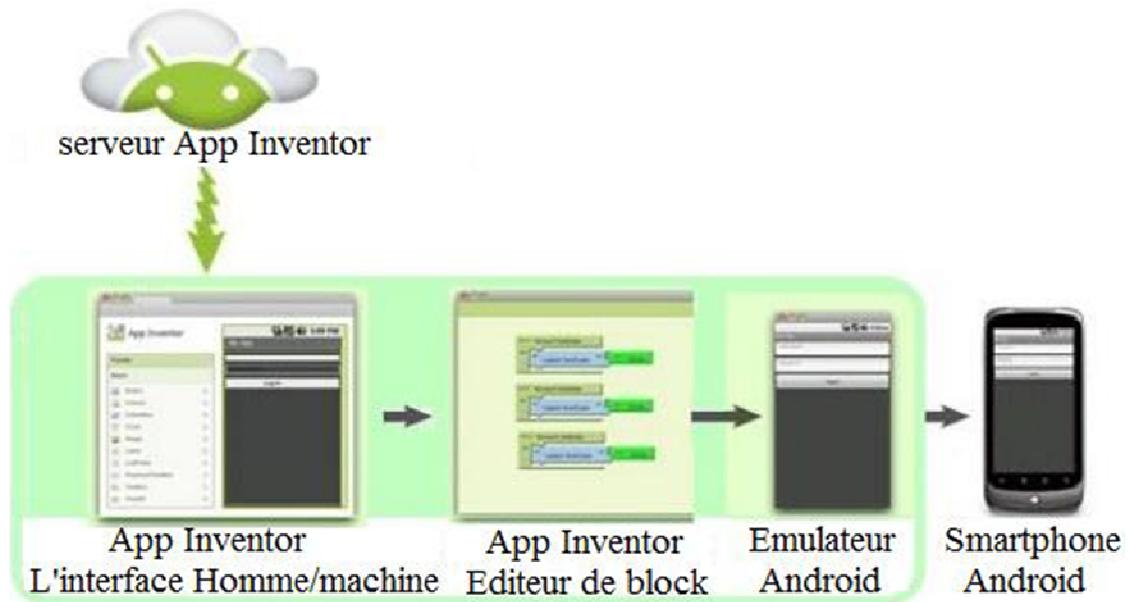


Figure 3.6. Conception d'une application Android par APP INVENTOR.

Dans cette partie, nous avons divers blocs graphiques tels que des boutons, des champs de texte, des images, des sliders, ainsi que des layouts. Ces derniers sont utilisés pour délimiter les zones de notre application. Nous avons également des sensors pour exploiter les capteurs du téléphone et certains autres outils multimédias tels que le GPS et les fonctions de communication Bluetooth.

c. Blocs de programmation

On retrouve dans la partie bloc toutes les fonctions mathématiques, logiques ou événementielles, relatives aux objets utilisés dans la partie graphique, ce qui permet de profiter de la simplicité de la programmation orientée objet sous forme de blocs prêts à être programmés, qu'il ne leur reste plus qu'à organiser selon le programme souhaité.

d. L'émulateur

Cette étape nous permet de tester l'application. L'émulateur peut remplacer un terminal réel afin de vérifier le bon fonctionnement du programme [28].

3.4.2 Concevoir un projet

Pour effectuer cette tâche il faut ouvrir un compte google à partir d'un navigateur internet. Ensuite, accéder au site Web de MIT App Inventor : <http://appinventor.mit.edu/>.

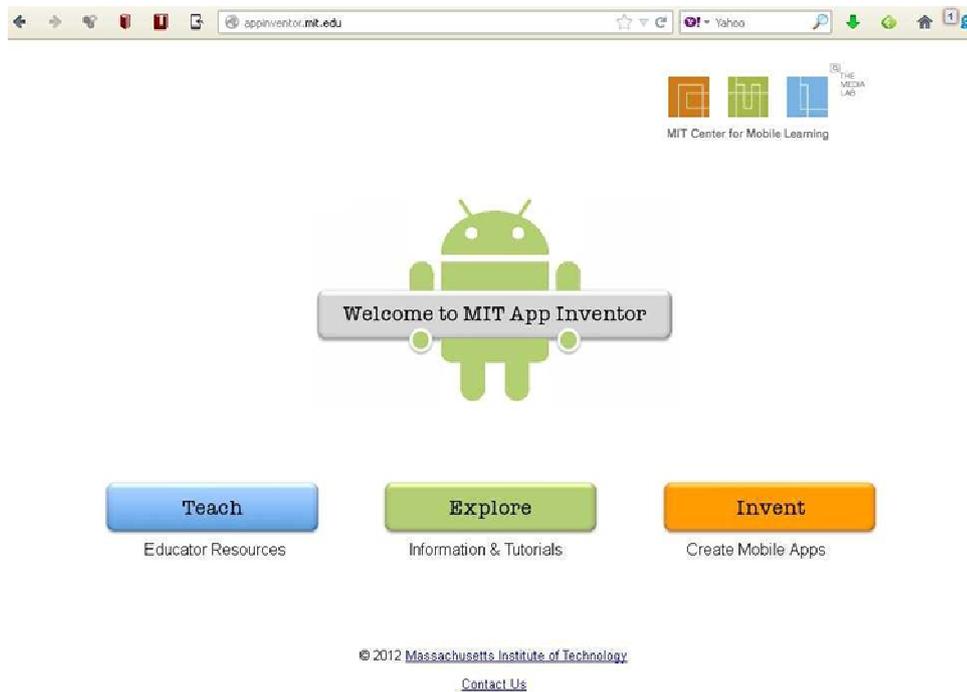


Figure 3.7. Site MIT APP INVENTOR.

Les étapes à suivre pour concevoir un projet sont :

- Cliquer sur le bouton « **invent** ».

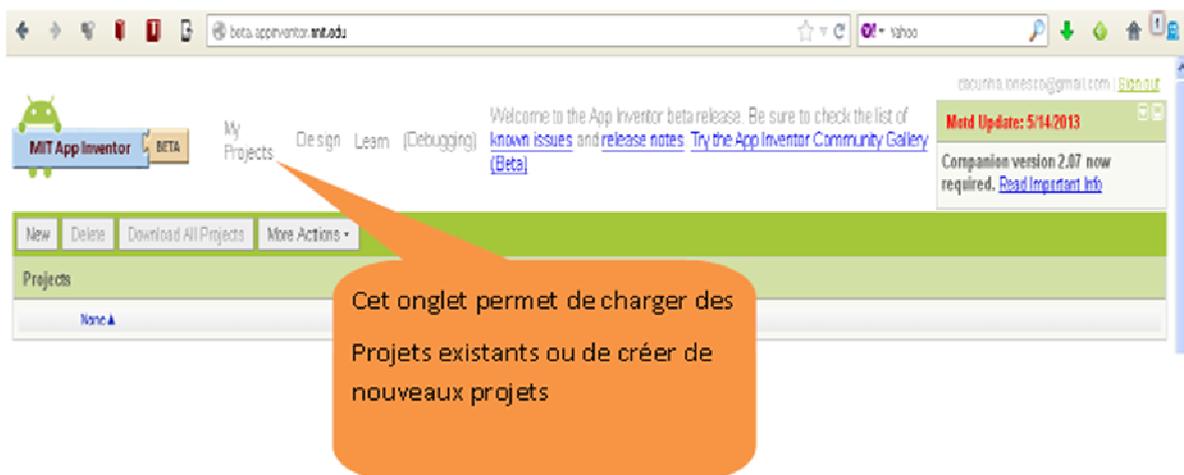


Figure 3.8. Nouveau projet sur APP INVENTOR.

- Cliquer ensuite sur « **new** » pour avoir un nouveau projet et nommer-le.



Figure 3.9. Nom du projet APP INVENTOR.

- L'interface d'utilisateur.

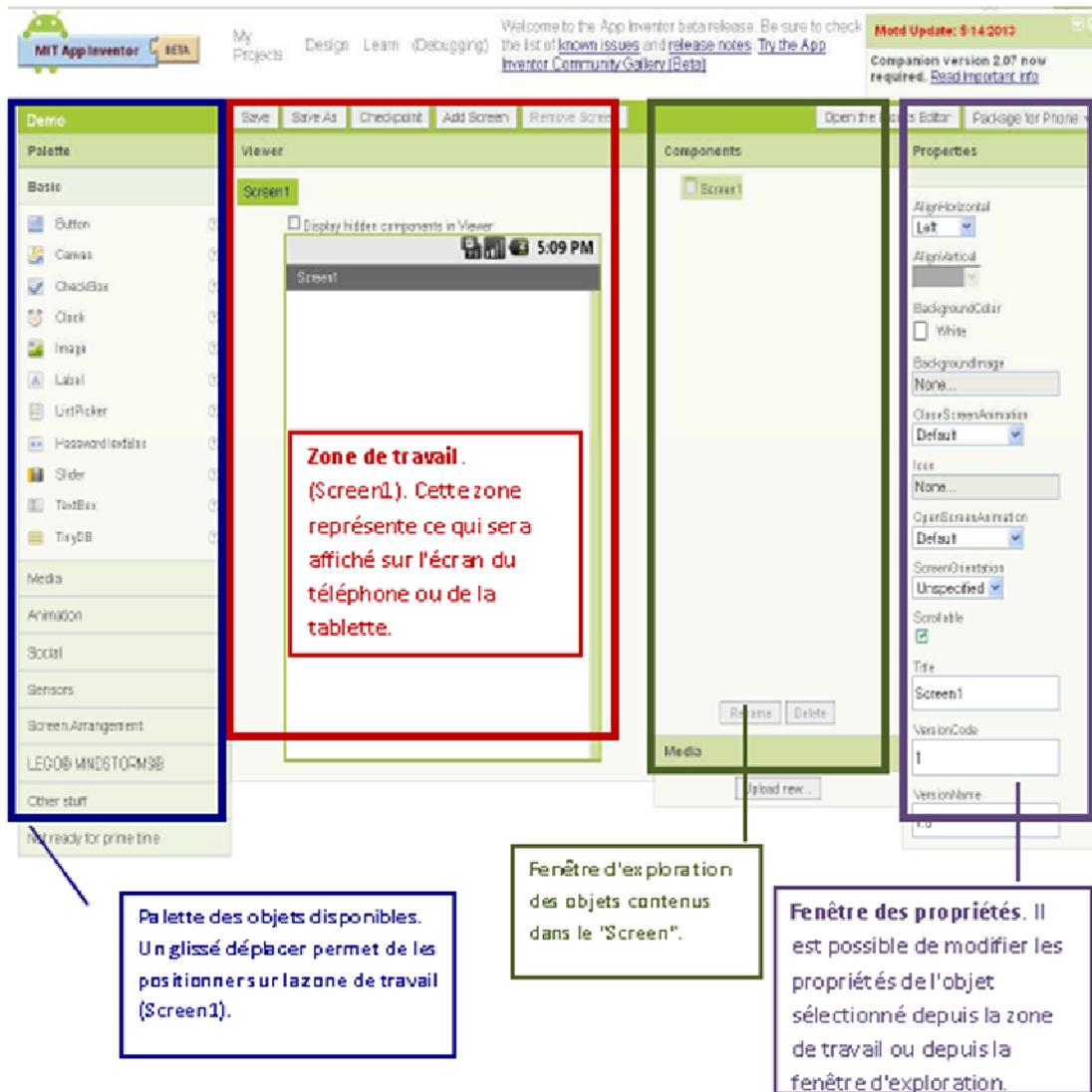


Figure 3.10. L'interface utilisateur APP INVENTOR.

3.5 Flutter

Flutter est un framework développé par Google. De ce fait, les ingénieurs ont pu observer les points forts et les faiblesses de chaque outil existant pour n'en extraire que la quintessence.

En fait, c'est l'une des briques essentielles de Fuchsia/Andromeda. En effet ce framework est utilisé pour tout ce qui est interface utilisateur. Aujourd'hui, Flutter est surtout connu pour sa capacité à concevoir des applications multiplateformes natives pour Android et iOS (Windows/Mac/Linux sont également pris en charge).

L'équipe de Flutter vient principalement du web (plus précisément de Chrome) et a essayé d'adapter sa philosophie au monde du mobile. Ils se sont appuyés pour cela sur Skia, le moteur qui fait partie intégrante de Chrome ou encore de toute la gestion de texte d'Android.



Figure 3.11. Logo Flutter.

3.5.1 Langage de programmation DART

DART est un langage de programmation conçu par Google. Depuis l'émergence de Flutter, ce langage a retrouvé des couleurs et a même connu une version 2.0.

Si Google a opté pour Dart, c'est parce qu'il propose deux modes de fonctionnement. Le premier, appelé AOT (pour Ahead Of Time), permet de générer une application native pour chaque plateforme. L'avantage de Flutter, par rapport à ses concurrents, est donc fort, puisque le code sera optimisé directement pour l'architecture sur laquelle il fonctionnera.

Le deuxième mode de fonctionnement est appelé JIT (Just-In-Time) et offre la fonctionnalité Hot Reload pendant le développement. La conception d'une application Android native prend beaucoup de temps entre chaque build.

L'idée du Hot Reload dans Flutter est de corriger ce problème en ne prenant que quelques millisecondes (voire secondes dans le pire des cas) entre chaque modification. Le développement de son application est alors beaucoup plus rapide.

En plus de ces deux modes, Dart est également très efficace dans la gestion de l'allocation et du "garbage collector", c'est-à-dire l'élimination des objets en mémoire qui ne sont plus utilisés.

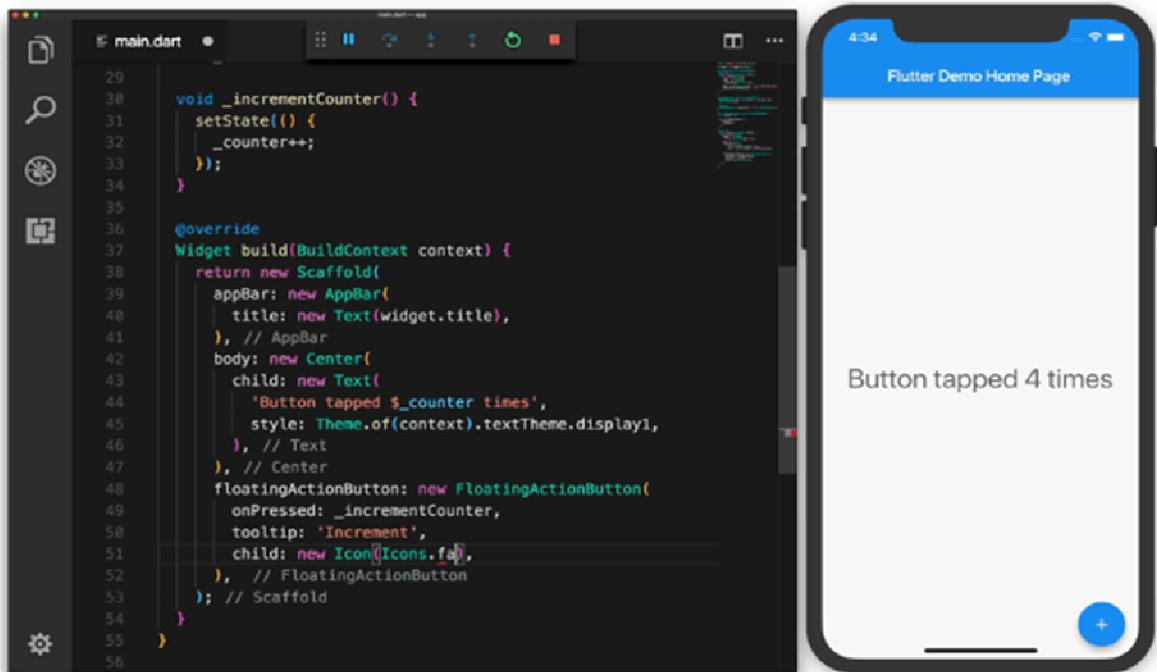


Figure 3.12. Exemple de développement d'application Android sur Flutter.

3.5.2 Caractéristiques et avantages du Flutter

Les caractéristiques et les avantages de l'utilisation de Flutter sont les suivants :

- Base de code unique pour Android et iOS.
- Fonction de rechargement à chaud (hot reload).
- Open-source par Google.
- Programmation Dart.
- Très rentable.
- Développement plus rapide.
- Excellente expérience utilisateur.
- Gamme de widgets.
- Productivité améliorée.
- Sécurisé.

3.6 Conception de l'application

Cette section développe comment programmer la connexion Bluetooth entre l'application et le module Bluetooth HC-05 relié au robot :

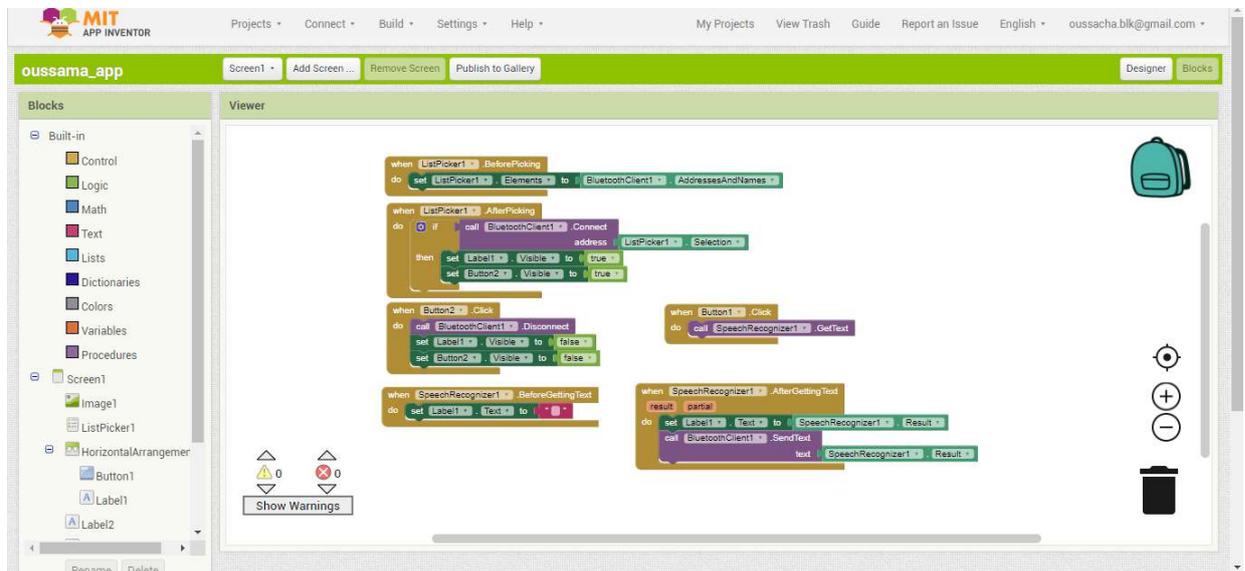


Figure 3.13. Interface du code source.

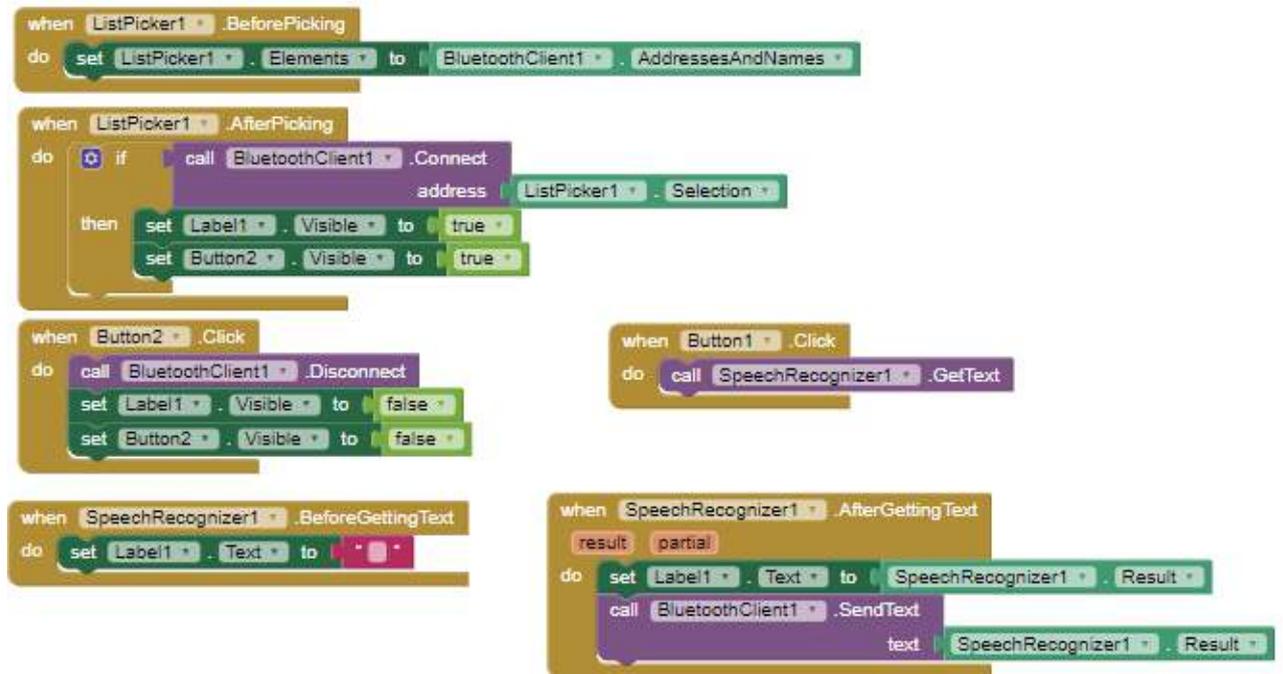


Figure 3.14. Code source utilisant MITAPP INVENTOR.

```
Extension: Dart build.gradle X AndroidManifest.xml
C:\Users\oussama> Documents\arduino_voice-master> android> app> build.gradle
1 def localProperties = new Properties()
2 def localPropertiesFile = rootProject.file('local.properties')
3 if (localPropertiesFile.exists()) {
4     localPropertiesFile.withReader('UTF-8') { reader ->
5         localProperties.load(reader)
6     }
7 }
8
9 def flutterRoot = localProperties.getProperty('flutter.sdk')
10 if (flutterRoot == null) {
11     throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties f
12 }
13
14 def flutterVersionCode = localProperties.getProperty('flutter.versionCode')
15 if (flutterVersionCode == null) {
16     flutterVersionCode = '1'
17 }
18
19 def flutterVersionName = localProperties.getProperty('flutter.versionName')
20 if (flutterVersionName == null) {
21     flutterVersionName = '1.0'
22 }
23
24 apply plugin: 'com.android.application'
25 apply plugin: 'kotlin-android'
26 apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
27
28 android {
29     compileSdkVersion 29
30
31     sourceSets {
32         main.java.srcDirs += 'src/main/kotlin'
33     }
34
35     lintOptions {
36         disable 'InvalidPackage'
37     }
38
39     defaultConfig {
40         // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-
41         applicationId "com.example.arduino_voice"
42         minSdkVersion 21
43         targetSdkVersion 29
44         versionCode flutterVersionCode.toInteger()
45         versionName flutterVersionName
46     }
47
48     buildTypes {
49         release {
50             // TODO: Add your own signing config for the release build.
51             // Signing with the debug keys for now, so `flutter run --release` works.
52             signingConfig signingConfigs.debug
53         }
54     }
55 }
56
57 flutter {
58     source '../..'
59 }
60
61 dependencies {
62     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
63 }
64
```

Figure 3.15. Code source utilisant Flutter.

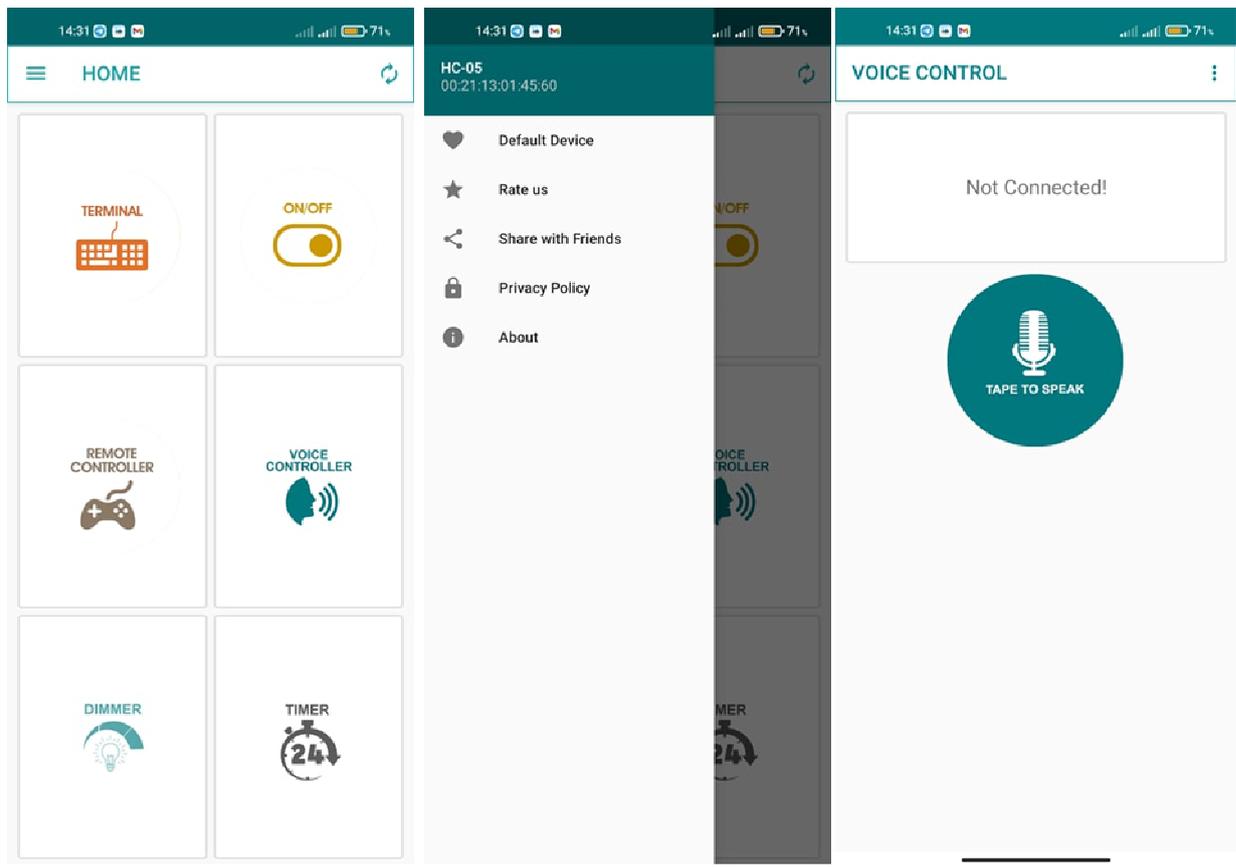


Figure 3.16.L'interface utilisateur de l'application.

3.7 Simulation du robot

3.7.1 ISIS PROTEUS

Le logiciel ISIS de PROTEUS est surtout connu pour simuler des schémas électriques et cela afin de pouvoir détecter certaines erreurs dès la phase de conception. aspect graphique des circuits. Indirectement, les circuits électriques conçus avec ce logiciel peuvent être utilisés dans la conception car il permet d'avoir l'essentiel de l'aspect graphique des circuits.



Figure 3.17. Logo du logiciel ISIS PROTEUS.

Pour lancer la simulation, il faut d'abord ajouter au logiciel Proteus les bibliothèques nécessaires à la simulation comme la bibliothèque Arduino, la bibliothèque Ultrasons et la bibliothèque Bluetooth. Ensuite on charge le programme 'HEX' généré par le logiciel Arduino IDE dans l'Arduino.

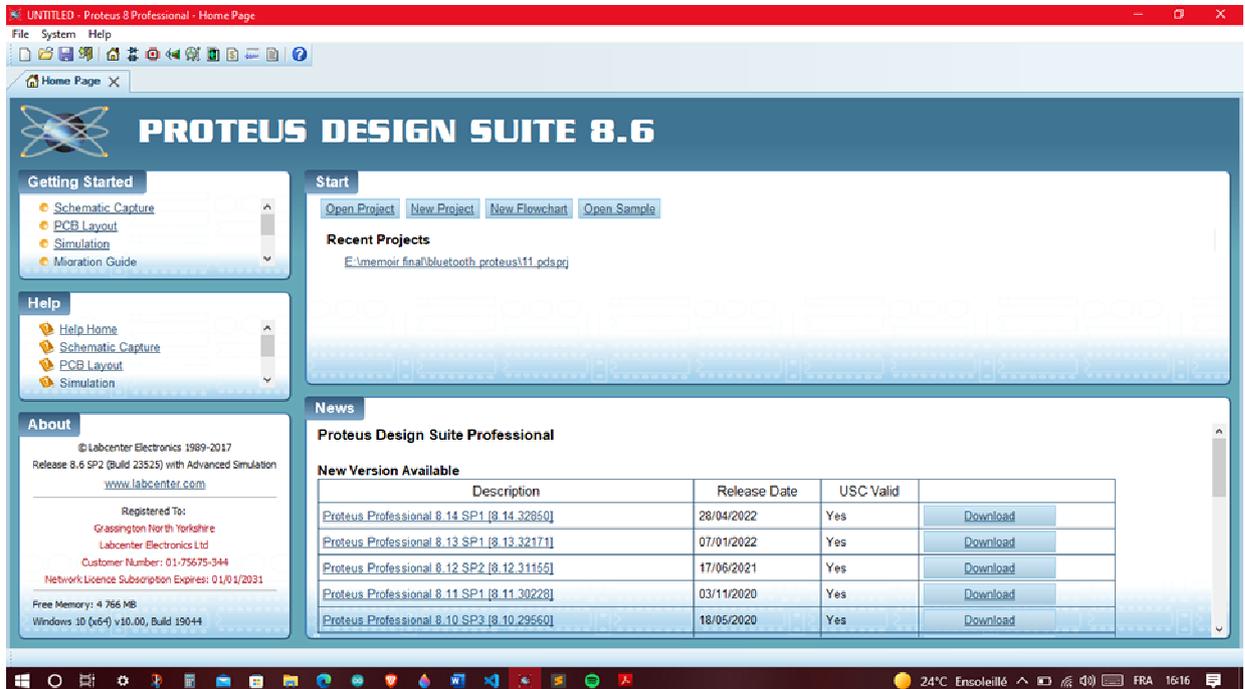


Figure 3.18.L'interface utilisateur PROTEUS.

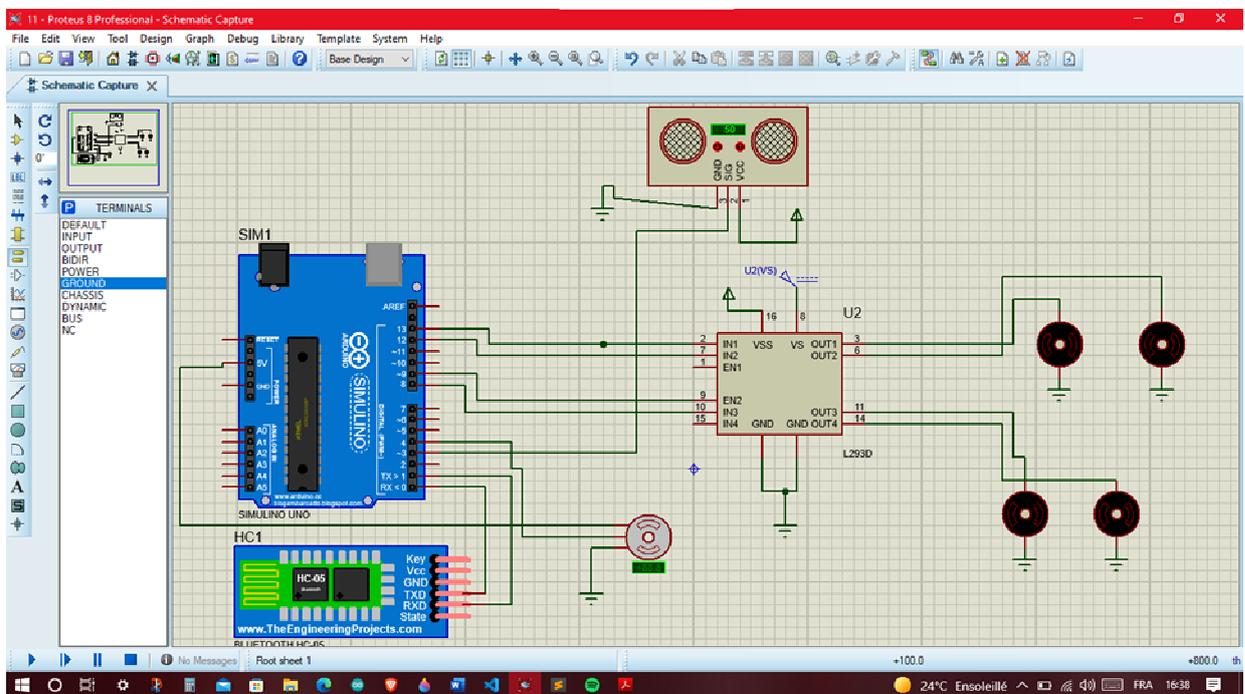


Figure 3.19.Schéma du circuit sous PROTEUS.

3.8 Réalisation du robot mobile

3.8.1 Assemblage électrique et connexion au Bluetooth

Le module Bluetooth HC-05 a quatre broches et il est connecté à la carte Arduino comme suit : Les broches Tx et Rx du Bluetooth aux broches numériques 0 et 1 respectivement de la carte Arduino UNO. Pour l'alimentation du module Bluetooth, nous avons utilisé un transformateur AC/DC 220v/5v.

Pour connecter le Smartphone au Bluetooth, il faut accomplir les étapes suivantes :

- Rechercher les adresses MAC des appareils disponibles depuis le menu Bluetooth du Smartphone.
- Sélectionner l'adresse MAC correspondante au module Bluetooth HC-05.
- Saisir le code PIN du module Bluetooth HC-05 sur le Smartphone.
- Avant de brancher le Vcc, il faut d'abord allumer la carte Arduino, la LED du module Bluetooth clignotera avec une cadence de 2 secondes.
- Une fois le Smartphone connecté au HC-05, la LED s'arrête de clignoter et reste allumée.

3.8.2 Assemblage global

Dans la construction du robot, nous avons utilisé quatre moteurs et un servomoteur à 180° (le servo tourne à 90° soit à droite soit à gauche selon la commande) inclus un capteur à ultrasons qu'il captera s'il y a un obstacle ou non. Le module Bluetooth HC-05 est le pont de communication entre l'Arduino et le Smartphone.

Pour faire avancer/reculer le robot, les quatre moteurs sont alimentés en même temps (la polarisation du moteur détermine le sens de rotation). Pour la rotation gauche/droite, seuls deux moteurs sont alimentés du même côté.

- Câblage avec l'Arduino
- Câblage du pilote de moteur L293D (il est placé directement au-dessus de l'Arduino)
- Câblage du module ultrasons(placer le module ultrasons avec son support sur le servomoteur).

Le servomoteur permet de tourner le module ultrasons à droite et à gauche pour balayer l'environnement et par conséquent déterminer approximativement la forme de l'obstacle.

3.8.3 Google assistant :

Google Assistant est l'assistant intelligent à commande vocale de Google. Il s'agissait à l'origine d'une mise à niveau d'une extension de Google Now ; élargissant les commandes vocales «OK Google» qui existait déjà dans Google.

« OK Google » ou « Hey Google » donne un accès à des commandes vocales, à la recherche vocale et au contrôle des appareils on utilise une commande vocale ; vous donne la possibilité de faire plusieurs actions comme envoyer des messages, vérifier des rendez-vous et ainsi de suite sur votre appareil Android, iPhone ou iPad, mais allant bien au-delà, avec une expérience d'IA centrée sur le bot, conçue pour vous fournir des interactions conversationnelles.

Car oui, l'étape suivante est la conversation avec la machine. Cela signifie que vous n'avez pas besoin de dire « Hey Google » pour les demandes suivies. L'assistant Google apprendra également à vous reconnaître parmi les autres personnes présentes dans la pièce. Les actions multiples sont également une nouvelle fonctionnalité qui vous permet de demander plusieurs choses en même temps. La maîtrise de telles requêtes comme celle-ci est probablement ce qui va permettre à l'Assistant Google de mieux avancer dans la concurrence contre ses rivaux.

De plus, un nouveau mode Pretty Please pour Google Assistant vous aidera à vous assurer que tout le monde dans votre foyer dit «s'il vous plaît» et «merci». Lorsqu'il est activé, il répond positivement lorsque vous prononcez l'une de ces phrases. Pretty Please peut être activé pour un membre spécifique de votre famille. Ainsi, vous pouvez encourager vos enfants à être polis.

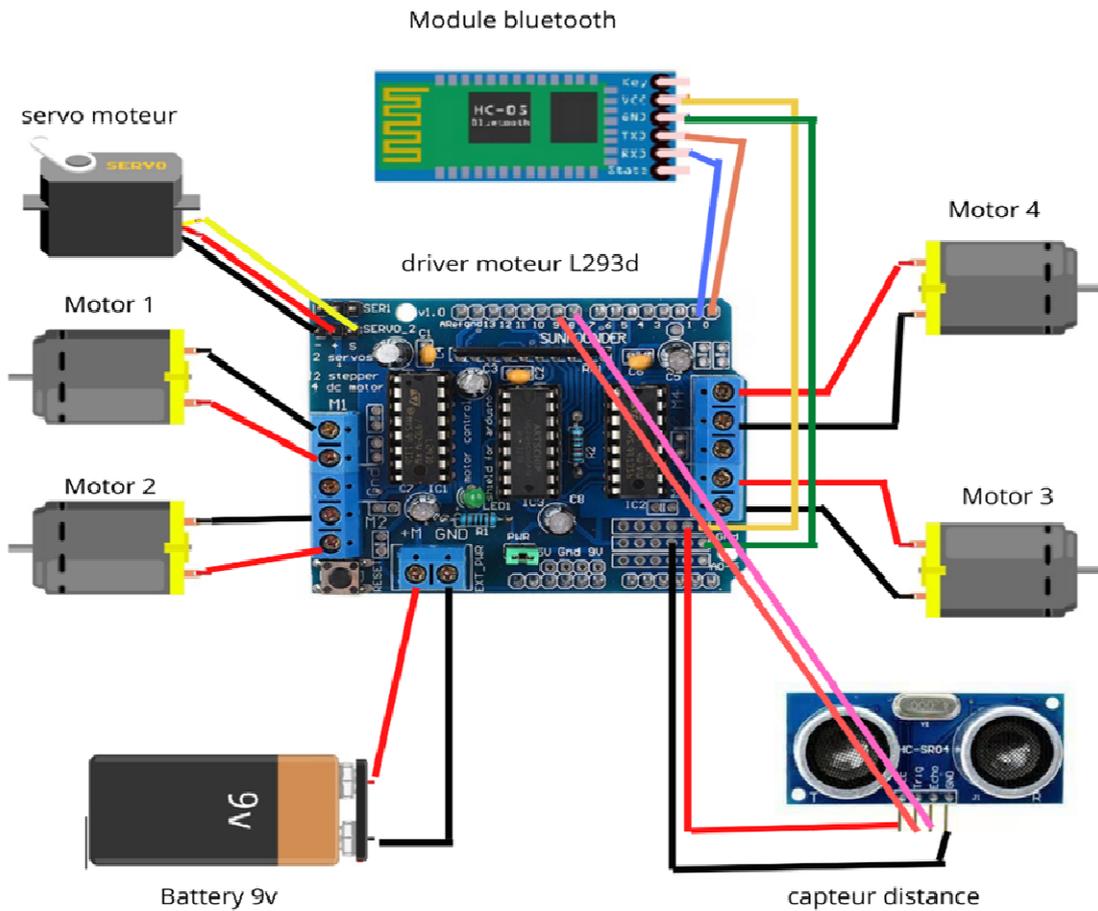


Figure 3.20. Schéma du robot mobile.

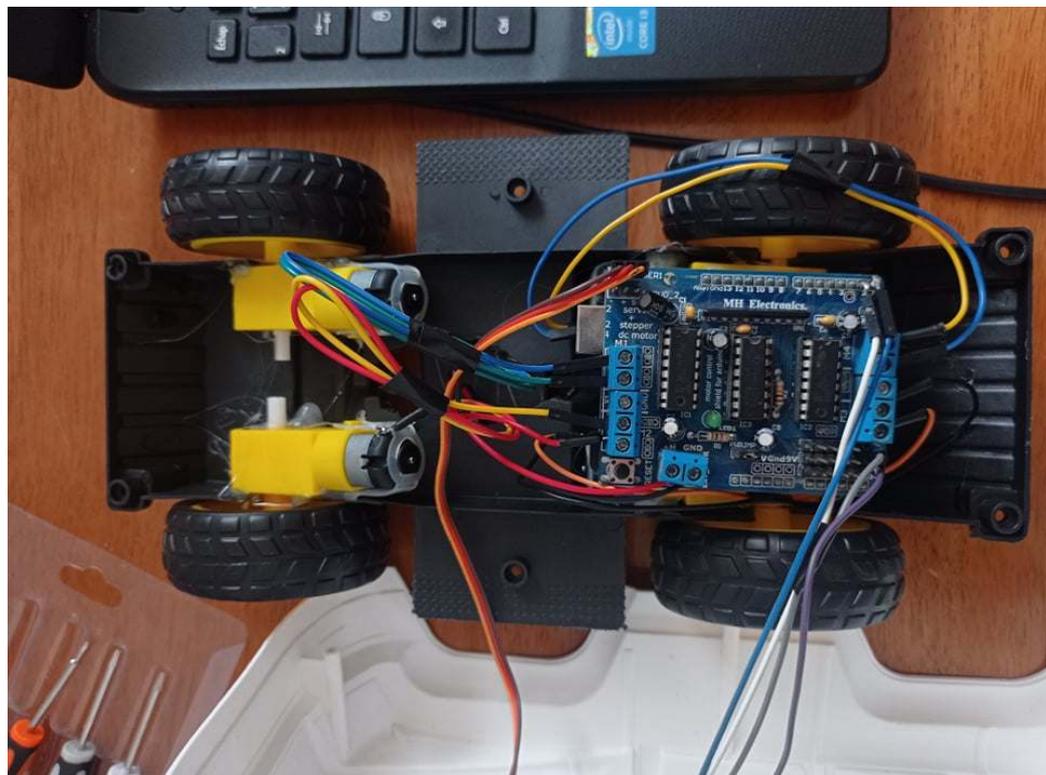


Figure 3.21. Montage électrique du robot mobile.



Figure 3.22. Photos finales du robot mobile.

3.9 Conclusion

Ce chapitre étudie la conception pratique d'un robot à commande vocale. Pour cela nous avons présenté les différents composants utilisés et les schémas réalisés sous ISIS et la simulation par PROTEUS ainsi qu'une réalisation d'un système de commande pour un robot mobile.

Nous avons également abordé divers domaines tels que : l'informatique, l'électronique, l'électronique programmée et les systèmes embarqués. Nous avons essayé certaines méthodes de commande comme : par PC local et par téléphone portable via Bluetooth (commande vocale).

Conclusion générale

L'objectif de ce projet de master est de développer un prototype de robot à commande vocale utilisant une carte Arduino UNO. La voix sera utilisée pour diriger le mouvement du robot qui répondra aux individus qui le contrôlent par des sons de voix humaine en réponse à chaque commande prononcée.

Le concept pourrait profiter aux personnes ayant des besoins spéciaux en leur permettant de se déplacer de façon autonome. Cela peut être bénéfique pour les personnes handicapées physiques qui peuvent dire à une voiture robotisée d'avancer, de reculer, de tourner à droite, de tourner à gauche ou de s'arrêter, et la le véhicule obéira à leurs ordres.

Le prototype est construit autour d'une carte Arduino UNO qui sera reliée à un Smartphone doté d'Android via Bluetooth pour écouter les instructions vocales. La commande donnée via la voix est traitée par une application Android et envoyée à la carte Arduino qui contrôle le mouvement du robot. Après avoir reçu les données, le robot répond à l'instruction en se déplaçant dans la bonne direction conformément à la commande vocale. La communication de ces deux parties est assurée par un module Bluetooth.

Des capteurs à ultrasons connectés à l'Arduino aident à l'arrêt automatisé du robot en cas de détection soudaine d'un obstacle.

Le matériel du prototype a été assemblé et la programmation a été effectuée à l'aide de l'IDE Arduino.

La reconnaissance vocale des mots parlés était assurée par Google Assistant.

Enfin, l'analyse des tests sur le prototype réalisé a révélé que les instructions (commandes) ont été exécutées avec succès.

Code Arduino

```
#include <AFMotor.h>

#include <Servo.h>

String command;

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

Servo myservo;

void setup() {
  Serial.begin(9600);
  myservo.attach(10);
  myservo.write(90);
}

void loop() {
  delay(10);
  while(Serial.available()) {
    command = "";
    command = Serial.readString();
    Serial.print(command);
  }
  if(command == "avancer"){
    left();
```

```

}else if(command == "arrière"){
    right();
}else if(command == "à gauche"){
    backward();
}else if(command == "à droite"){
    forward();
}else if(command == "stoppe") {
    Stop();
}
command = "";
}
void forward() {
    myservo.write(0);
    delay(500);
    myservo.write(90);
    delay(500);
    motor1.setSpeed(255);
    motor1.run(FORWARD);
    motor2.setSpeed(255);
    motor2.run(FORWARD);
    motor3.setSpeed(255);
    motor3.run(FORWARD);
    motor4.setSpeed(255);
    motor4.run(FORWARD);
    delay(700);
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
    motor4.run(RELEASE);
}

```

```

}
void backward() {
    myservo.write(180);
    delay(500);
    myservo.write(90);
    delay(500);
    motor1.setSpeed(255);
    motor1.run(BACKWARD);
    motor2.setSpeed(255);
    motor2.run(BACKWARD);
    motor3.setSpeed(255);
    motor3.run(BACKWARD);
    motor4.setSpeed(255);
    motor4.run(BACKWARD);
    delay(700);
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
    motor4.run(RELEASE);
}
void left() {
    motor1.run(BACKWARD);
    motor1.setSpeed(255);
    motor2.run(FORWARD);
    motor2.setSpeed(255);
    motor3.run(FORWARD);
    motor3.setSpeed(255);
    motor4.run(BACKWARD);
    motor4.setSpeed(255);

```

```
delay(1500);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
}
void right() {
    motor1.run(FORWARD);
motor1.setSpeed(250);
motor2.run(BACKWARD);
motor2.setSpeed(250);
motor3.run(BACKWARD);
motor3.setSpeed(250);
motor4.run(FORWARD);
motor4.setSpeed(250);
delay(1500);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
}
void Stop() {
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
    motor4.run(RELEASE);
}
```

Bibliographie

- [1] RICHA Brahim et BOUYEKHF Mohamed Redah, « Etude et réalisation d'un bras robot à 2DDL », mémoire master, Université KHEMIS MILIANA (2017).
- [2] Jugurtha LAOUEDJ et Abderrahmane DJERIANI, « Commande vocale d'un rover via Bluetooth à base d'Arduino », Mémoire Master Professionnelle , Université Mouloud Mammeri – tizi Ouzou.
- [3] B. BAYLE. « Robotique Mobile », Ecole Nationale supérieure de Strasbourg, France, 2008-2009.
- [4] R. Carona « CONTROL OF UNICYCLE TYPE ROBOTS » Université de Lisboa, Portugal 2008.
- [5] Barnes D.P., Summers, P., Shaw, A., An investigation into aerobot technologies for planetary exploration (étude sur la technique des aérojets pour l'exploration planétaire), in Proc. 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation, ASTRA 2000. ESTEC Noordwijk, NL, pp. 3.6-5, December 2000.
- [6] A. Colozza, G. A. Landis, and V. Lyons, Overview of Innovative Aircraft Power and Propulsion Systems and Their Applications for Planetary Exploration (vue d'ensemble sur les nouveaux systèmes de propulsion des aéronefs et leurs applications dans l'exploration planétaire), NASA TM-2003-212459 (July 2003)
- [7] « ROPITS : le robot de transport automatique par Hitachi » [archive], sur GNT - Génération Nouvelles Technologies
- [8] « People Transportation » [archive], sur B2B Service Robots - Robosoft
- [9] « RoboCourier Autonomous Mobile Robot » [archive], sur Swisslog - inspired solutions
- [10] Mais le docteur est d'or (Titre original : Gold) Pocket, (ISBN 978-2-266-06926-7) p. 235.
- [11] A. ALLOUI, A. HADJ BRAHIM. « Proposition d'une solution multi-agent pour la commande et la coopération multi -robot mobile ». Mémoire d'ingénieur d'état en automatique, Université Biskra, Juin 2007.
- [12] TAKHI Hocine (conception et réalisation d'un robot mobile a base d'arduino) Memoire De Master Université Amar Telidji- Laghouat

- [13] S. KERMICHE « Modélisation et commande d'un robot par méthodes intelligentes » Thèse doctorat d'état 2006, université Annaba.
- [14] BELAID Issam Eddine et AZZOUZ Youcef, « étude et réalisation d'un bras manipulateur de type r-p », Université TLEMCEM (2018).
- [15] A. PRUSKI. « Robotique générale » Edition Ellipse 1988.
- [16] Robotics (par AppuuKuttan) , AppuuKuttan, ISBN978-81-89866-38-9
- [17] [https:// arduino.technologiescollege.fr/IMG/PDF/cahier_ \(\) _initialisation](https://arduino.technologiescollege.fr/IMG/PDF/cahier_()_initialisation).
- [18] Le grand livre d'Arduino, chapitre I, page 3,4,5 et 6.
- [19] Astalaseven, Eskimon et olyte, "Arduino pour bien commencer en électronique et en programmation » ,
- [20] Jean-Noël « Livret Arduino en français », Centre de Ressources Art Sensitif, novembre 2006.
- [21] Erik Bartmann, « LE GRAND LIVRE D'ARDUINO »,
- [22] Jim Kardach, « Tech History: How Bluetooth got its name » [archive], EETimes, 3 April 2008,
- [23] Ericsson, « Milestones in the Bluetooth advance », page web archivée
- [24] Brent A. Miller, « Bluetooth : A Wireless Personal Area Network », dans Hossein Bidgol, éd. The Internet Encyclopedia, Volume 1, John Wiley & Sons, 2004, p. 84-85.
- [25] <https://knowledge.parcours-performance.com/arduino-bluetooth-hc-05-hc-06>
- [26] MEKKI Walid et BELHADJ Mohammed, « Conception et réalisation d'un panneau solaire à base d'Arduino », mémoire master, Université d'Ain-Temouchent (2018).
- [27] SAADI Naim, « Réalisation d'une Maquette Didactique de Robotique et Arduino », mémoire master, Université Bejaia (2017).
- [28] TAZEROUT Yacine, « Commande d'un bras robotisé avec une application androïde », mémoire master, Université LORRAINE (2018).