

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

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



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Génie Electrique

Spécialité : Informatique Industriel

THÈME

Commande et contrôle d'un robot mobile par Bluetooth à base d'un microcontrôleur

Préparé par : KEBIR HABIBA
BENABDELLAH NOUREDDINE

Devant le Jury :

Nom et prénoms	Grade	Qualité
OTMANI Redoune	MAA	Président
MOULAHCENE Fateh	MAA	Examinateur 1
BELHADJI Youcef	MAA	Examinateur 2
TIFFOUR Abdelkader	MAA	Encadreur

Année Universitaire : 2016/2017

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

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



MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Génie Electrique

Spécialité : Informatique Industriel

THÈME

**Commande et contrôle d'un robot
mobile par Bluetooth à base d'un
microcontrôleur**

Préparé par : KEBIR Habiba
BENABDELLAH Noureddine

Devant le Jury :

Nom et prénoms	Grade	Qualité
MOULAHCENE Fateh	MAA	Président
OTMANI Redoune	MAA	Examinateur 1
BELHADJI Youcef	MAA	Examinateur 2
TIFFOUR Abdelkader	MAA	Encadreur

Année Universitaire : 2016/2017

Résumé

Ce projet concerne la conception, la réalisation et la commande d'un robot mobile à trois roues à l'aide d'une carte électronique "Arduino" qui va être relié au robot après avoir développé le programme en logiciel Arduino, et une application androïde son rôle est de contrôler ce robot à distance en deux mode "manuel - vocal" par Bluetooth.

Mots clés : Robot Mobile - Arduino – Androïde – Bluetooth

ملخص

يتعلق هذا المشروع بتصميم وإنتاج مراقبة الروبوت المحمول ثلاث عجلات باستخدام لوحة إلكترونية "أردوينو" يتم وصلها إلى الروبوت بعد أن يتم تحميل برنامج، وكذلك تطبيق أندرويد الذي يديره روبوت عن بعد بوضعين "عادي- صوتي" عن طريق البلوتوث.

كلمات مفتاحية: روبوت المتحرك - أردوينو - أندويد - بلوتوث

Abstract

This project concerns the design, production and control of a three-wheeled mobile robot using an "Arduino" electronic board adapted to connect it to the robot after having developed the Arduino software program, and an android application which control the robot in two modes "manual - vocal" by bluetooth.

Keywords: Robot Mobile - Arduino - Android - Bluetooth

Remerciements

Nous tenons à remercier ALLAH le tout puissant

Nous remercions notre encadreur, M: Tiffour Abdelkader pour avoir accepté de diriger notre travail, et pour tous ses conseils et orientations qui nous ont été particulièrement bénéfiques.

Nous remercions, également, les membres de jury pour avoir accepté de juger notre travail.

Toute notre gratitude va, aussi, à tous ceux, qui de près ou de loin, nous ont prodigué leurs conseils et encouragements.

Tableau I.1 : développements historiques de la robotique	6
Tableau I.2 : Les avantages et des inconvénients des différents types de robots à roues	11
Tableau I.3: Applications des robots mobiles.....	14
Tableau II.1 : Différentes capteur passif	21
Tableau III.1 : Développements historiques des microprocesseurs/microcontrôleur.....	31
Tableau III.2 : Différentes cartes Arduino et leur Aspects techniques	38
Tableau III.3 : Tableau explicatif « interface ISIS & ARES »	50
Tableau III.4 : Tableau explicatif « interface MIT app Inventor2 ».....	52
Tableau III.5 : Table de vérité de l'étage de puissance	52
Tableau III.6 : Tableau de commande BT	Error! Bookmark not defined. 55
Tableau III.7 : Description des cas d'utilisations.....	63

Figure 1 :Organisation de mémoire.....	3
Figure I.1 : architecture générale d'un robot.....	7
Figure I.2 : Architecture d'un robot mobile	9
Figure I.3 : Robot de type uni-cycle.....	10
Figure I.4 : Robot de type tricycle.....	10
Figure I.5 : Robot de type voiture	11
Figure I.6 : Robot mobile omnidirectionnel	10
Figure I.7 : Robots mobiles à chenilles	12
Figure I.8 : Robots marcheurs	11
Figure I.9 : Robot volant.....	13
Figure II.1 : Vu générale d'un capteur	18
Figure II.2 : Capteur dans la chaîne de mesure.....	19
Figure II.3 : Schéma d'un capteur composite.....	21
Figure II.4 : La sensibilité d'un capteur	22
Figure II.5 : Capteur ultrasonique	23
Figure II.6 : Schéma synoptique d'un capteur ultrasonique	24
Figure II.7 : Montage module HC-06	25
Figure II.8 : Représentation d'un moteur à courant continu.....	28
Figure III.1 : Schéma des éléments principaux d'un microcontrôleur.....	32
Figure III.2 : L'architecture d'un microcontrôleur (Von Neuman)	33
Figure III.3 : L'architecture d'un microcontrôleur (Harvard)	40
Figure III.4 : Différent types des cartes arduino	42
Figure III.5 : Microcontrôleur ATmega328	43
Figure III.6 : Différent types des capteurs pour l'arduino	44
Figure III.7 : Différents actionneurs pour l'arduino.....	45
Figure III.8: Interface de logiciel arduino	49
Figure III.9 : Détail de barre des boutons	49
Figure IV.1 : Fenêtre Proteus professionnelle v8.1 (ISIS).....	51
Figure IV.2 : Fenêtre Proteus professionnelle v 8.1 (ARES).....	52
Figure IV.3 Interface design MIT app Inventor2	53
Figure IV.4Interface Blocks MIT app Inventor2	54
Figure IV.5 : Fenêtre Enterprise Architect.....	55
Figure IV.6 : Schéma bloc	57
Figure IV.7 : Schéma de brochage L293D	58

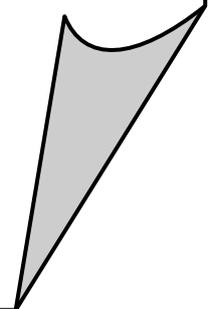
Liste des figures

Figure IV.8 : Driver L293D réalisée par le logiciel PROTEUSE	53
Figure IV.9 : Driver L293D en réalité.....	53
Figure IV.10 : Capteur bluetooth Module HC-06	54
Figure IV.11 : Robot mobile assemblé	57
Figure IV.12: Carte réalisée sous ISIS-PORTEUS.....	56
Figure IV.13 : Programme Arduino.....	65
.....	65
Figure IV.14 : Digramme de cas d'utilisation	64
Figure IV.15 : Digramme de cas séquence «Authentification ».....	65
Figure IV.16 : Digramme de cas séquence « Registraction ».....	65
Figure IV.17 : Digramme de cas séquence « Manuel control »	66
Figure IV.18 : Fenêtre d'accueil M2II	67
Figure IV.19 : Fenêtre de registration	69
Figure IV.20 : Fenêtre de menu.....	70
Figure IV.21 : Fenêtre de la commande manuel	70
Figure IV.22 : Fenêtre de la commande vocale.....	71

AREF :	Analog RE ference, référence analogique.
ARES:	Advanced R outing and E ding S oftware
CAO :	Conception A ssisté par O rdinateur
CISC:	I nstruction C omplicated S et C omputer
CPU:	Central C ross C nit
EA :	E ntreprise A rchitecte
EEPROM :	E lectrally E rasable P rogrammable R ead O nly M emory
FTDI :	F uture T echnology D evelopments I nternational
ICSP :	I n C ircuit S erial P rogramming,(programmation série sur circuit.)
IDE :	I ntegrated D evelopment E nvironment
ISIS :	I ntelligent S chematic I nterface S ystem
MCC :	M oteur à C ourant C ontinu
MIT :	Massachusetts Institute of T echnologie
OTPROM :	O ne- T ime P rogrammable R ead O nly M emory
PC:	P ersonal C omputer
PCB:	P rinted C ircuit B oard
PDA :	P ersonal D igital A ssistant
PWM :	P ulse W idth M odulation (largeur d'impulsion modulée)
RAM:	R andom A ccess M emory
RISC:	R educed I nstruction S et C omputer
ROM:	R ead O nly M emory
SCL :	S erial C lock
SDA:	S erial D ata A ccess
SDK :	S oftware D evelopment K it
SPI :	S erial P eripheral I nterface (Interface Série Périphérique)
TWI :	T wo W ire I nterface - Interface "2 fils"
UAL :	U nité A rithmétique et L ogique
UART :	U niversal A synchronous R eceiver T ransmitter
UC :	M icrocontrôleur
UML :	U nified M odeling L anguage
UML:	U nified M odeling L anguage
UP :	M icroprocesseur
USB :	U niversel S erial B us

ID :	Identifiant
USER :	Utilisateur
M2II :	Master 2 Informatique Industriel
V :	Volt
A :	Ampère
T :	Temps
D :	Distance
V_{son} :	Vitesse de déplacement des ultrasons dans l'air
E :	coefficient d'élasticité
P :	Densité de l'air
Ω :	Ohm
GND :	Ground - la masse - 0 Volt.
E/S :	Entrée / Sortie
I/P :	Input / Output
A/N :	Analogique / Numérique

***Introduction
Générale***



Contexte

Depuis fort longtemps, l'humain rêve de créer des machines intelligentes capables d'effectuer des tâches à sa place. Ainsi, les humains auraient plus de temps à consacrer pour leurs loisirs, ou prendraient moins de risques pour effectuer des tâches dangereuses. Or créer une machine pouvant réaliser des tâches que seuls les humains sont normalement capables de faire n'est pas aussi simple qu'on pourrait le penser. En effet, sans toujours y penser, les tâches les plus élémentaires de la vie quotidienne d'un humain peuvent devenir extrêmement complexes lorsqu'on les analyse de plus près.

Malgré son aspect, la robotique tire ses origines des civilisations les plus antiques. Il est peut-être même vrai que toutes les autres sciences ont existé juste pour permettre à la robotique de se développer afin d'aider l'Homme à créer ce « esclave » qui appliquera ses ordres au doigt et à l'œil et qui le libérera à jamais du travail.

Problématique

Un robot est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a en prend notre cas et on pose quelques problématiques :

- Est-ce que le robot mobile est compliqué par rapport à autre type des robots? Effectuez toutes ses fonctions sans complexité? Et dans tout environnement?
- Vous pouvez programmer par d'autres moyens?
- La carte arduino « Uno » Est-ce suffisant pour le programme?
- Avez-vous robot mobile pouvez-vous déplacer dans toutes les directions et éviter les obstacles?
- Pouvez-vous contrôler ce robot à distance ?

Objectifs de notre travail

L'objectif principal de notre travail est de réaliser un robot mobile commandé par le Bluetooth basé sur une carte de capteurs émetteur/récepteur « Bluetooth – Ultrason » et une carte de commande à base d'une carte arduino, ainsi qu'un driver L293D pour commander les deux moteurs à courant continu, aussi la création d'une application androïde à l'aide de MIT app Inventor qui assure la commande de notre robot à distance, La tâche principale du robot fait de se déplacer dans quatre directions « Avant – Arrière – droite - gauche », même il peut éviter les obstacles, comme il peut être contrôlé par le robot par la voix

Organisation de mémoire

Ce travail est alors organisé en quatre chapitres de la manière suivante :

- Le premier chapitre présente certaines généralités concernant la robotique et les différents types des robots et robots mobiles.
- Le second chapitre introduit les capteurs, les actionneurs.
- Le troisième chapitre est consacré au microcontrôleur, Arduino langages de programmation et en particulier l'IDE de l'Arduino.
- La conception et la réalisation pratique de notre carte, fait l'objectif du quatrième chapitre Enfin, notre mémoire est clôturé par une conclusion générale & quelques perspectives.

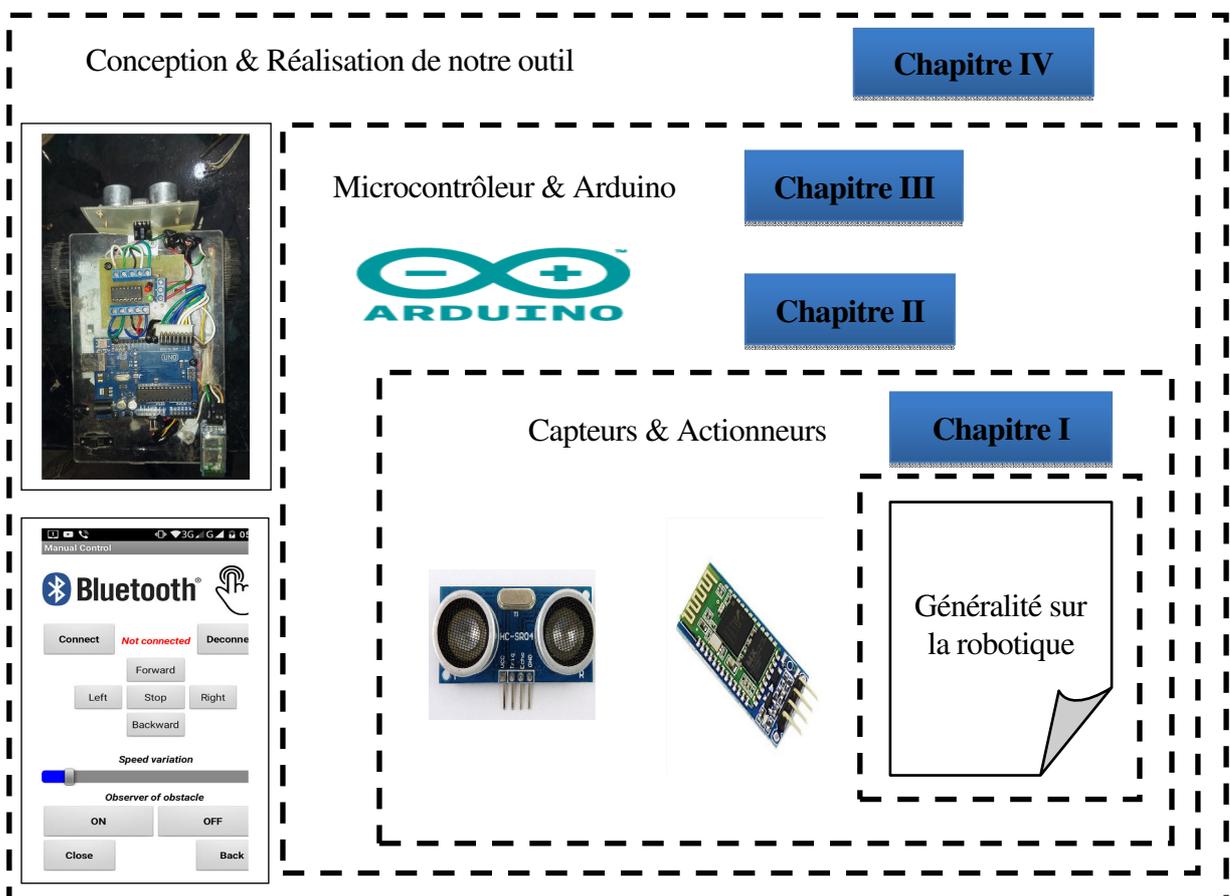
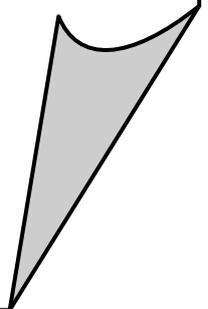


Figure 1 : Organisation de mémoire

Généralité
sur
la robotique



I.1 Introduction

Le rêve de créer des machines qui sont qualifiés et intelligentes a fait partie de l'humanité depuis le début du temps. Ce rêve est en train de devenir une partie de la réalité de notre monde.

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

Les robots industriels sont généralement fixes, ils sont utilisés dans des nombreuses applications industrielles: l'assemblage mécanique, la soudure, la peinture... Les robots mobiles ne sont pas fixes, ils sont classifiés selon la locomotion en robots marcheurs, à roues, à chenilles... comme ils peuvent être classifié selon le domaine d'application en robots militaires, de laboratoire, industriels et de services.

Les robots mobiles présentent un cas particulier en robotique. Leur intérêt réside dans leur "mobilité", destinés à remplir des taches pénibles (exemple : transport de charges lourdes) et ils travaillent même en ambiance hostile (nucléaire, marine, spatiale, lutte contre l'incendie, surveillance...).

I.2 Bref historique sur la robotique

La robotique est passée par plusieurs générations comme suit [1]

Année		
1600 av.j-c	L'horlogerie	Les clepsydres (horloges à eau) furent probablement inventées vers 1600 av. J-C. en Egypte.
1738	Les automates	Un des automates les plus surprenants est certainement le canard digérateur de Jacques de Vaucanson (1738).
1915	La cybernétique	Les automates autorégulés forment la base de la cybernétique,Le chien électrique de Hammond et Miessner (1915) se guide à la lumière
1928		le chien de Henri Piraux (1928)
1953		le renard de Ducrocq (1953)
1950	L'intelligence artificielle	Alan Turing à définir en 1950 une méthode afin de tester la présence d'une conscience ou d'une intelligence au sein d'une machine.
1960	Les rovers	Véhicules destinés à explorer la surface de planètes de façon quasiment autonome, effectuer des mesures et prélever des échantillons, les rovers (ou astromobiles)

		furent développés par les russes dès les années 1960.
1997-2012		La Nasa développe aussi des rovers à partir de 1970 dans le cadre du programme Apollo. Sojourner explora la planète Mars en 1997 et en 2012
1985	Les animats	Animaux artificiels, ils sont l'objet d'une branche particulière de la cybernétique dont l'objectif est de copier les mécanismes du vivant. Si le terme animat ne fut défini qu'en 1985 par S.W. Wilson
1973	Les robots humanoïdes	La maîtrise de la bipédie est certainement l'accomplissement le plus technique de l'histoire des robots humanoïdes. WABOT-1 de l'université de Waseda (1973) est l'un des premiers robots humains capables de se déplacer (de façon rudimentaire)
1986-1997		les premiers robots de Honda E0 à E6 (1986-1993) et P1 à P3 capable de marcher comme un humain et de monter des escaliers (1992-1997).
200-2011		Le onzième robot développé par Honda, Asimo, est capable de courir (2000). La dernière version de 2011 peut sautiller sur un pied, courir à 9km/h, manipuler des objets délicatement comme remplir un gobelet

Tableau I.1 : développements historiques de la robotique

I.3 C'est quoi un robot ?

I.3.1 Étymologie

Robot a été utilisé pour la première fois en 1921 par Karel Capek dans sa pièce R.U.R. (Rossums Universal Robots). Il provient du tchèque "robota" qui signifie corvée, travail obligatoire (forcé). Le terme robotique a été employé pour la première fois par Asimov en 1941.[2]

I.3.2 Définition

- Un robot est un système mécanique mû par des actionneurs et commandé par un ordinateur qui est destiné à effectuer une grande variété de tâches[3]
- Un robot est la combinaison de composants matériels (mécanique, hardware) et immatériels (logiciels)
- C'est une machine pouvant manipuler des objets en réalisant des mouvements variés dictés par un programme aisément modifiable

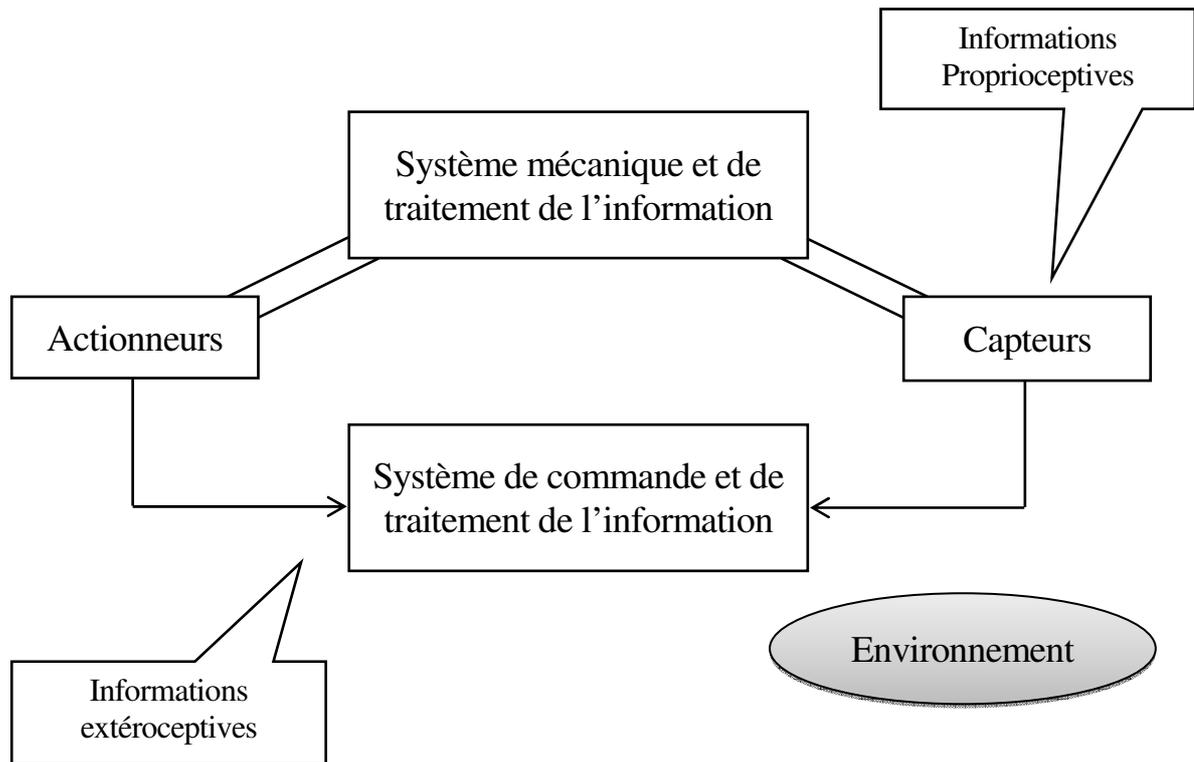


Figure I.1 : architecture générale d'un robot

I.3.3 Trois lois de la robotique

N'importe quel robot est soumis à trois lois comme suit :

1. Un robot ne peut blesser un humain ni, par son inaction, permettre qu'un humain soit blessé.
2. Un robot doit obéir aux ordres donnés par les humains, sauf si de tels ordres se trouvent en contradiction avec la première loi.
3. Un robot doit protéger sa propre existence aussi longtemps qu'une telle protection n'est pas en contradiction ni avec la première et/ou ni avec la deuxième loi [5]

I.4 Type des robots

Au cours de l'histoire, 2 types différents de robots sont apparus attestant chacun de nouvelles évolutions : Robots manipulateurs et Robots mobiles.

I.4.1 Robots manipulateurs

Un robot manipulateur est en forme d'un bras et se compose d'un certain nombre de segments qui est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont des dispositifs qui permettent aux humains d'interagir avec des objets dans

un environnement en toute sécurité. Les robots manipulateurs sont utilisés dans des applications industrielles pour s'effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage.[4]

I.4.1.1 Types des robots manipulateurs

Les robots manipulateurs viennent sous plusieurs formes. Les formes se répartissent en cinq grandes catégories :

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

I.4.2 Robots mobiles

Les robots mobiles ont une place particulière en robotique. Leur intérêt réside dans leur mobilité qui ouvre des applications dans de nombreux domaines. Comme les robots manipulateurs, ils sont destinés à assister l'homme dans les tâches pénibles (transport de charges lourdes), monotones ou en ambiance hostile (nucléaire, marine, lutte contre l'incendie, surveillance....ect).

L'aspect particulier de la mobilité impose une complexité technologique et méthodologique qui s'ajoute en général aux problèmes rencontrés par les robots manipulateurs. La résolution de ces problèmes passe par l'emploi de toutes les ressources disponibles tant au niveau technologique (capteur, motricité, énergie) qu'à celui du traitement des informations par utilisation des techniques de l'intelligence artificielle ou de processeurs particuliers (vectoriels, cellulaires).[6]

I.4.2.1 L'architecture des robots mobiles

L'architecture des robots mobiles se structure en quatre éléments :

- La structure mécanique et la motricité
- Les organes de sécurité
- Le système de traitement des informations et gestion des tâches.
- Le système de localisation.

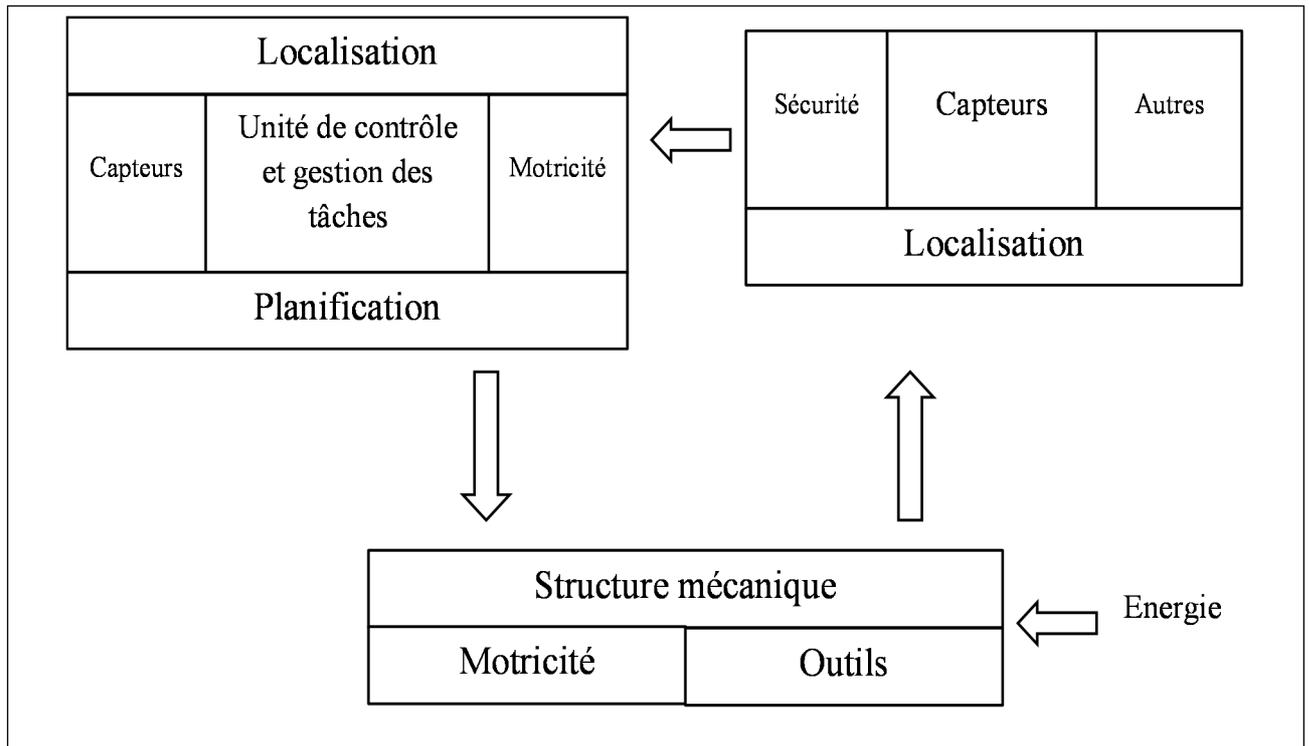


Figure I.2 : Architecture d'un robot mobile

I.4.2.2 Classification selon le type de locomotion

Selon le système de locomotion, on peut distinguer quatre types des robots:

A. Les robots mobiles à roues

La mobilité par roues est la structure mécanique 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.

Nous citerons ici les quatre classes principales de robots à roues. [6]

- 1. Robot uni-cycle :** est actionné par deux roues indépendantes, il possédant éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites.



Figure I.3 : Robot de type uni-cycle

2. **Robot tricycle** : 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. 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.4 : Robot de type tricycle

3. **Robot 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.



Figure I.5 : Robot de type voiture

4. **Robot omnidirectionnel** : est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral.

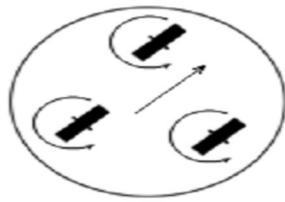


Figure I.6 : Robot mobile omnidirectionnel

Type du robot	Avantage	Inconvénient
Unicycle	<ul style="list-style-type: none"> • Stable • Rotation sur soi-même • Complexité mécanique faible 	<ul style="list-style-type: none"> • Non-holonyme
Tricycle	<ul style="list-style-type: none"> • Complexité mécanique modérée 	<ul style="list-style-type: none"> • Non-holonyme • Peu stable • Pas de rotation sur soi-même
Voiture	<ul style="list-style-type: none"> • Stable • Complexité mécanique 	<ul style="list-style-type: none"> • Non holonyme • Pas de rotation sur soi-même
Omnidirectionnel	<ul style="list-style-type: none"> • Holonyme • Stable • Rotation sur soi-même 	<ul style="list-style-type: none"> • Complexité mécanique importante

Tableau I.2 : Les avantages et des inconvénients des différents types de robots à roues

B. Les robots mobiles à chenilles

L'utilisation des chenilles présente l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence (présence de boue, herbe,...)



Figure I.7 : Robots mobiles à chenilles

C. 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.8 : Robots marcheurs

D. Les robots mobiles rampants

La reptation est une solution de locomotion pour un environnement de type «tunnel» qui conduit à réaliser des structures filiformes. Le système est composé d'un ensemble de module ayant chacun plusieurs mobilités. Ici aussi les techniques utilisées découlent des méthodes de locomotion des animaux et des insectes.

E. Autres moyens de locomotion

Les applications de ce type de robots sont très spécialisées et les architectures des robots sont en général spécifiques à l'application visée. [5]



Figure I.9 : Robot volant

I.4.3 Domaine d'application des robots mobiles

Parmi les domaines d'applications possibles de la robotique mobiles, citons :[5]

Domaine	Application
Industrie nucléaire	Surveillance de sites, manipulation de matériaux radioactifs, démantèlement de centrales
Sécurité civile	Neutralisation d'activité terroriste, déminage, pose d'explosif, surveillance de munitions
Militaire	Surveillance, patrouille, pose d'explosifs, manipulation de munitions
Chimique	Surveillance de site, manipulation de matériaux toxiques
Médecine	Assistance d'urgence, aide aux handicapés physiques, aux aveugles
Lutte contre l'incendie	Localisation d'une source d'incendie, détection de fumée, suppression de flammes
Sous-marine	Pose de câbles, recherche de nodules, recherche de navires immergés, inspection des fonds marins
Agricole	Cueillette de fruits, traite, moisson, traitement des vignes
Construction BTP	Projection mortier, lissage du béton
Nettoyage	Coque de navire, nettoyage industriel
Espace	Exploration
Industriel	Convoyage, surveillance

Tableau I.3 : Applications des robots mobiles

Un système robotique consiste non seulement des robots mais aussi d'autres dispositifs et systèmes qui sont utilisés avec le robot pour effectuer la tâche nécessaire. Les avantages des robots sont: [7]

- Robotique et automatisation peut dans de nombreuses situations d'accroître la productivité, la sécurité, l'efficacité, la qualité et la cohérence des produits.
- Les robots peuvent travailler dans un environnement dangereux, sans le besoin de soutien de la vie, ou les préoccupations concernant la sécurité.
- Robots travailler continuellement, sans ressentir une fatigue ou l'ennui, et ne nécessitent pas une assurance médicale ou de vacances.

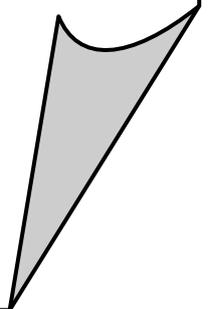
L'inconvénient des robots est qu'ils manquent de capacité de réagir en cas d'urgence, à moins que les situations comprises et les réponses sont inclut dans le système. Les mesures de sécurité nécessaires pour s'assurer qu'ils ne lèsent pas les opérateurs et n'endommagent les machines qui travaillent avec eux. Inconvénients des robots comprennent:[7]

- Le manque de pouvoirs prendre une décision , Réponse inadéquate ou mal
- Consommation de l'énergie.

I.5 Conclusion

A travers ce chapitre on peut dire que dans le cadre de la robotique, la robotique mobile joue un rôle à part. Contrairement aux robots industriels manipulateurs qui travaillent de façon autonome dans un grand nombre d'usines automatisées, les robots mobiles sont très peu répandus. Cette situation n'est pas due au manque d'applications possibles, mais dès qu'on dispose de la mobilité, on peut imaginer des robots facteurs, nettoyeurs, gardiens, démineurs, explorateurs, jardiniers et beaucoup d'autres. La faible diffusion est surtout due au fait que ces tâches ont une complexité bien supérieure à celles effectuées par des robots manipulateurs industriels. Le monde dans lequel un robot mobile doit se déplacer est souvent très vaste, partiellement ou totalement inconnu, difficilement caractérisable géométriquement et ayant une dynamique propre.

Capteurs
&
Actionneurs



II.1 Introduction

Ce chapitre discute sur les composants du robot. Ce n'est pas tous les composants, mais plus précisément, le chapitre traite les capteurs, actionneurs, et d'autres concepts sur ces deux termes.

Le système de commande est un système dynamique qui contient un dispositif de commande en tant que partie intégrante. Le but du contrôleur est de générer des signaux de commande, qui conduira le processus pour être contrôlé. Les actionneurs sont nécessaires pour effectuer les actions de contrôle, les capteurs et transducteurs aussi sont nécessaires pour mesurer les signaux de sortie (réponses de processus) et de mesurer les signaux d'entrée pour la commande anticipatrice, et pour diverses autres fins). Parmi les différents types et les niveaux de signaux sont présents dans un système de contrôle, la modification du signal (y compris le conditionnement du signal et la conversion du signal) est en effet joue un rôle crucial à tout système de contrôle. En particulier, la modification du signal est une considération importante dans l'interfaçage des composants. Il est clair que le sujet du système de contrôle doit contenir des capteurs, des transducteurs, actionneurs, modification du signal, et l'interconnexion des composants. En particulier, le sujet de l'identification des composants du système de commande doit respecter les fonctions, le fonctionnement et l'interaction, la bonne sélection et l'interfaçage de ces composants pour diverses applications de contrôle. La sélection des paramètres (y compris les réglages du système) aussi est une étape importante. La conception est une partie nécessaire de l'instrumentation du système de contrôle, car c'est elle qui nous permet de construire un système de contrôle qui répond aux exigences de performance "on commence, peut-être, avec des composants de base tels que les capteurs, les actionneurs, les contrôleurs, les compensateurs, et appareils de modification de signal."

II.2 Capteurs

Avant de donner la définition d'un capteur, il est nécessaire de connaître quelques définitions de métrologie.

Le mesurande : c'est l'objet de la mesure ou plus simplement la grandeur à mesurer. Le mesurage : c'est l'ensemble des opérations pour déterminer la valeur du mesurande. La mesure c'est le résultat d'u mesurage. Autrement dit c'est la valeur du mesurande.

II.2.1 C'est quoi un capteur ?

Un capteur transforme une grandeur physique en une grandeur normée, généralement électrique, qui peut être interprétée par un dispositif de contrôle commande.[8]

Un capteur est un dispositif convertissant une grandeur physique analogique (pression, température, déplacement, débit,...) en un signal analogique rendu transmissible et exploitable par un système de conditionnement (courant électrique, radiation lumineuse, radiofréquence). Le capteur est la partie d'une chaîne de mesure qui se trouve au contact direct du mesurande. [9]

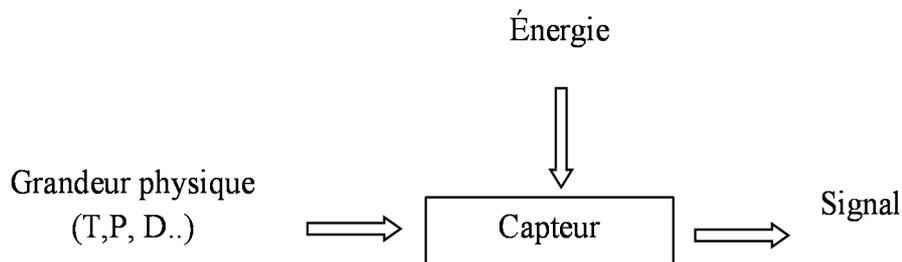


Figure II.1 : Vue générale d'un capteur

II.2.2 Chaîne de mesure

Généralement, le signal de sortie de capteur n'est pas directement utilisable. On appelle chaîne de mesure l'ensemble des circuits ou appareils qui amplifient, adaptent, convertissent, digitalisent le signal avant sa lecture sur le support de sortie.

Pour obtenir une image d'une grandeur physique, la chaîne de mesure peut faire intervenir plusieurs phénomènes différents. Par exemple, la mesure d'un débit peut se faire en plusieurs étapes :

- Transformation du débit en une pression différentielle.
- Transformation de la pression différentielle en la déformation mécanique d'une membrane.
- Transformation de la déformation mécanique en une grandeur électrique (à l'aide d'un piézoélectrique) via un circuit électrique associé.

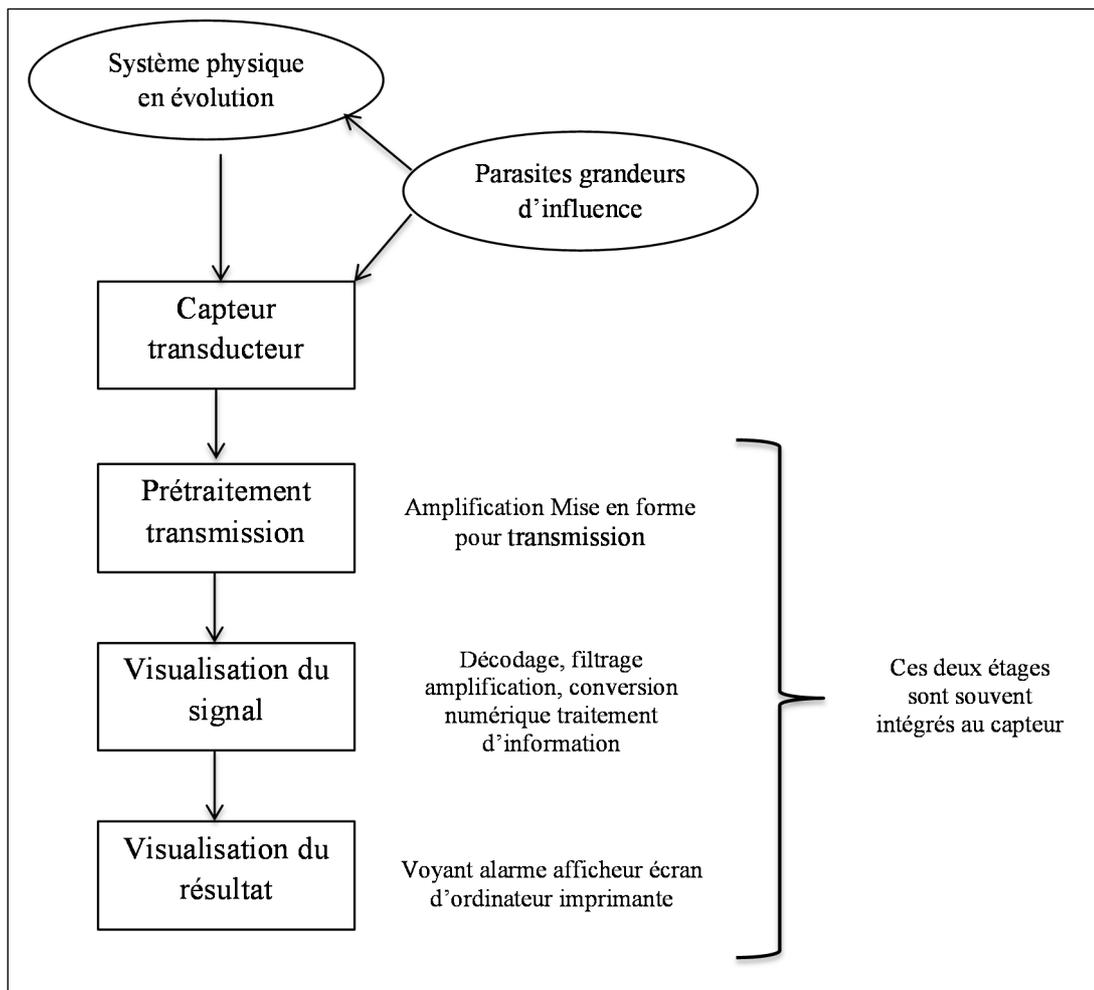


Figure II.2 : Capteur dans la chaîne de mesure

II.2.3 Classification des capteurs

On peut classer les capteurs de plusieurs manières :

- Par le mesurande qu'il traduit (capteur de position, de température, de pression, etc.)
- Par son rôle dans le processus industriel (contrôle de produit finis, de sécurité, etc.)
- Par le signal qu'il fournit en sortie qui peut être numérique, analogique, logique ou digital.
- Par leur principe de traduction du mesurande (capteur résistif, piézoélectrique, etc.)
- Par leur principe de fonctionnement : capteur Actif ou Passif.

Toutes ces classifications permettent d'avoir une vue d'ensemble des capteurs et bien sur aucune des méthodes de classification n'est meilleure que l'autre car toutes présentent des avantages et des inconvénients.

II.2.3.1 Capteur actif

Fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie propre au mesurande : énergie thermique, mécanique ou de rayonnement. [10]

- **Effet thermoélectrique (ou effet Seebeck) :** Un circuit formé de deux conducteurs de nature chimique différente, dont les jonctions sont à des températures T_1 et T_2 , est le siège d'une force électromotrice d'origine thermique $e(T_1, T_2)$.
- **Effet piézo-électrique :** L'application d'une contrainte mécanique à certains matériaux dits piézo-électriques (le quartz par exemple) entraîne l'apparition d'une déformation et d'une même charge électrique de signe différent sur les faces opposées.
- **Effet d'induction électromagnétique :** La variation du flux d'induction magnétique dans un circuit électrique induit une tension électrique (détection de passage d'un objet métallique).
- **Effet photo-électrique :** La libération de charges électriques dans la matière sous l'influence d'un rayonnement lumineux ou plus généralement d'une onde électromagnétique
- **Effet Hall :** Un champ magnétique B (aimant, angle q /surface du matériau) et un courant électrique I créent dans le matériau une différence de potentiel $U_H = K_H B I \sin q$
- **Effet pyroélectrique :** certains matériaux ont une polarisation spontanée en l'absence de champ électrique extérieur. Une variation de température induit une variation de cette polarisation et donc l'apparition de charges électriques à la surface du matériau

II.2.3.2 Capteur passif

Il s'agit généralement d'impédances (résistance, inductance, capacité) dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- d'une variation de dimension du capteur (capteurs de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile)
- d'une déformation résultant d'une force ou d'une grandeur s'y ramenant (pression accélération). Exemples : armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée à une structure déformable [10]

Grandeur mesurée	Caractéristique sensible	Types de matériaux utilisés
Température	Résistivité électrique	Platine, nickel, cuivre...
Rayonnement optique	Résistivité électrique	Semi-conducteur
Déformation	Résistivité électrique Résistivité magnétique	Alliage de Ni, Si dopé Alliage ferromagnétique
Position	Résistivité électrique	Matériaux magnétorésistants (bismuth, antimoine d'indium)
Humidité	Résistivité électrique	Chlorure de lithium

Tableau III.1 : Différentes capteur passif

II.2.3.3 Capteur composite

Un capteur composite est un capteur constitué d'un corps d'épreuve et d'un capteur actif ou passif. Le corps d'épreuve quant à lui est un capteur qui soumis au mesurande donne une grandeur physique non électrique appelée mesurande secondaire qui elle va être traduit en une grandeur électrique par un capteur. Le schéma d'un capteur composite est le suivant :

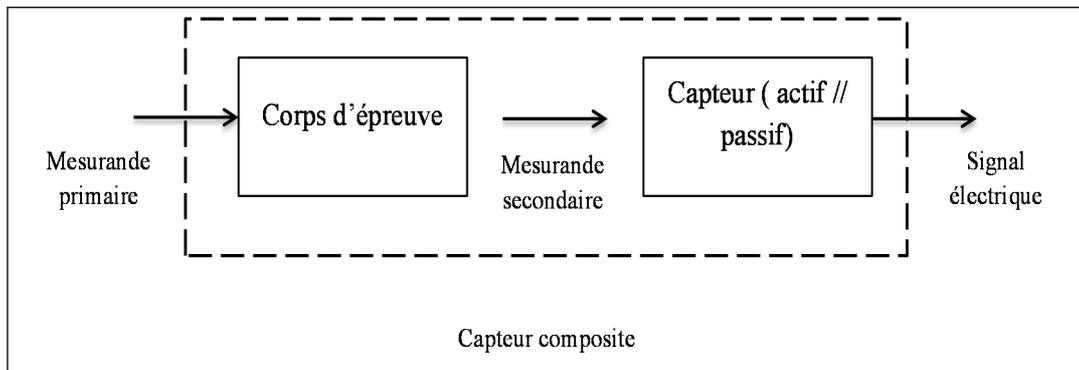


Figure II.3 : Schéma d'un capteur composite

II.2.3.4 Capteur intégré

Un capteur intégré est un capteur qui utilise la microélectronique. Ce capteur est constitué d'une plaque en silicium dans lequel on a fixé le capteur, le corps d'épreuve si besoin et d'autres composants électroniques qui peuvent servir à linéariser, amplifier, convertir le courant en tension, etc. Ce type de capteur est très utile vu qu'il fournit un signal linéaire avec une grande sensibilité, une miniaturisation et un coût faible.

II.2.4 Caractéristiques d'un capteur

Il existe plusieurs propriétés associées à un capteur qui sont critiques pour les performances du capteur. Les plus importantes sont :

1. **Etendue de mesure** : Valeurs extrêmes pouvant être mesurée par le capteur.
2. **Résolution** : Plus petite variation de grandeur mesurable par le capteur.

3. **Sensibilité** : Variation du signal de sortie par rapport à la variation du signal d'entrée.

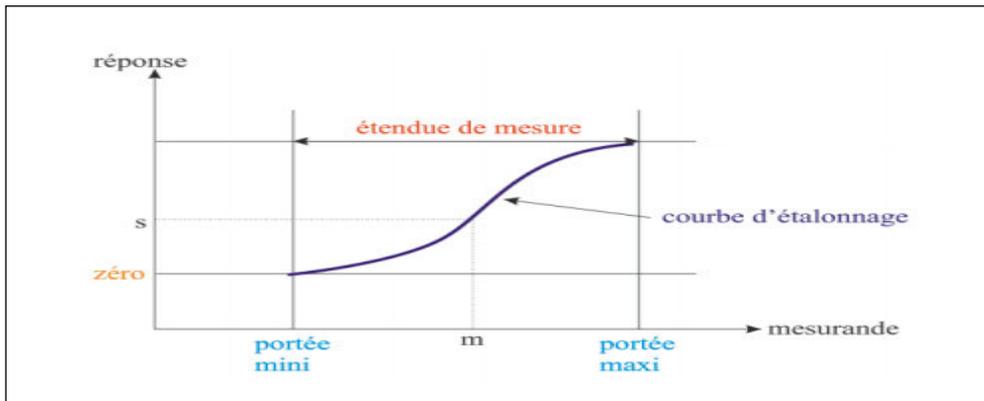


Figure II.4 : La sensibilité d'un capteur

- 4. **Précision** : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- 5. **Rapidité** : Temps de réaction du capteur. La rapidité est liée à la bande passante.
- 6. **Linéarité** : représente l'écart de sensibilité sur l'étendue de mesure

II.2.5 Capteur utilisé dans la réalisation

II.2.5.1 Capteur ultrasonique

a. C'est quoi un capteur ultrasonique ?

Les capteurs ultrasons fonctionnent en mesurant le temps de retour d'une onde sonore inaudible par l'homme émise par le capteur. La vitesse du son étant à peu près stable, on en déduit la distance à l'obstacle

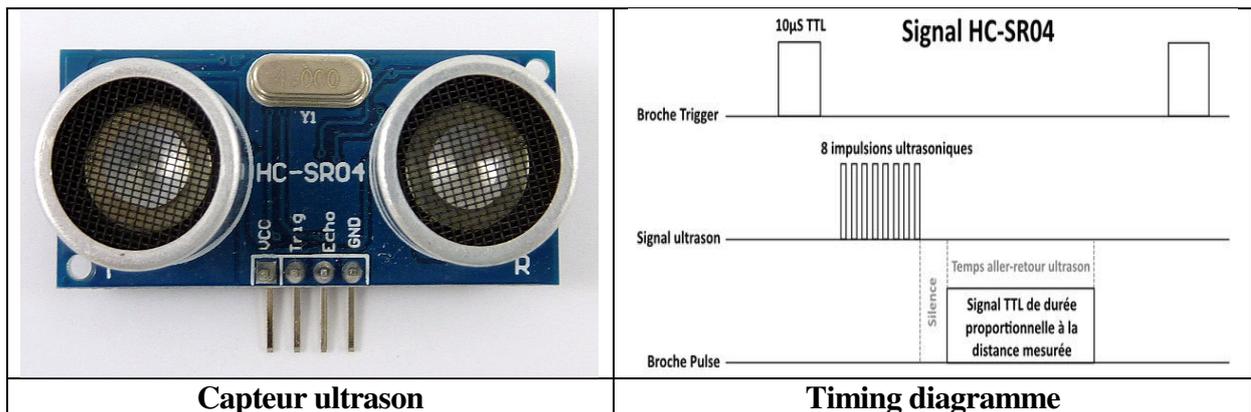


Figure II.5 : Capteur ultrasonique

Comme il est décrit précédemment, on utilise ces capteurs pour localiser le robot par rapport à des obstacles statiques ou dynamiques, ce capteur repose sur l'idée de calculer le temps de vol d'un signal ultrasonique depuis son point de départ. Alors si la distance entre le transducteur et

l'obstacle est « D » donc la distance parcourue par l'onde ultrasonique est « 2D », le temps du parcours est donné par :

$$T = 2D/V_{son}$$

Ou :

T : le temps entre l'émission et la réception.

D : distance entre le transducteur et l'obstacle.

V_{son} : la vitesse de déplacement des ultrasons dans l'air.

Généralement, l'air est la porteuse de l'onde ultrasonique, cette dernière se propage, alors, à la même vitesse que le son (célérité de l'onde sonore). Physiquement et mathématiquement la vitesse du son ne dépend pas de la fréquence du signal sonore, mais du type du milieu, de la pression et de la température suivant la relation mathématique suivante :

$$V_{son} = \sqrt{E/p} = \sqrt{Y.P/p}$$

V_{son} : Célérité du son.

E=Y.P : Coefficient d'élasticité du gaz.

p : Densité de l'air (1.293 Kg/m³).

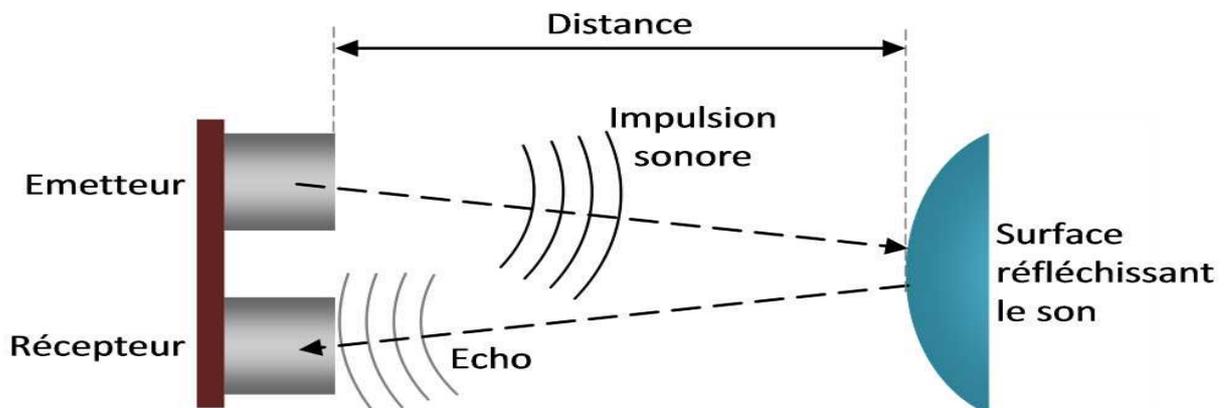


Figure II.6 : Schéma synoptique d'un capteur ultrasonique

II.2.5.2 Capteur Bluetooth

a. C'est quoi un capteur Bluetooth ?

Bluetooth est un standard de communication permettant l'échange bidirectionnel de données à très courte distance en utilisant des ondes radio UHF sur une bande de fréquence de 2,4 GHz. Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires. Elle peut remplacer par exemple les câbles entre ordinateurs, tablettes, téléphones mobiles entre eux ou avec des imprimantes, scanners, claviers, souris, manettes de jeu vidéo, téléphones portables, assistants personnels, systèmes et kits mains libres micro ou écouteurs, autoradios, appareils photo numériques, lecteurs de code-barres et bornes publicitaires interactives.

b. Modules bluetooth HC-05 et HC-06

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

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

Les deux modules peuvent être utilisés en mode COMMANDE, pour les programmer avec des « commandes AT », ou en mode DATA, pour échanger des données.

c. Montage pour HC-06

Noter que TXD et RXD du module BlueTooth HC-06 doivent être « inversés » : TXD arrive au port RX de l'arduino et RXD au port TX de l'arduino

Noter aussi le diviseur de tension pour la liaison RX à TX: la tension d'entrée des données doit être de l'ordre de 3.3V et non 5V.[11]

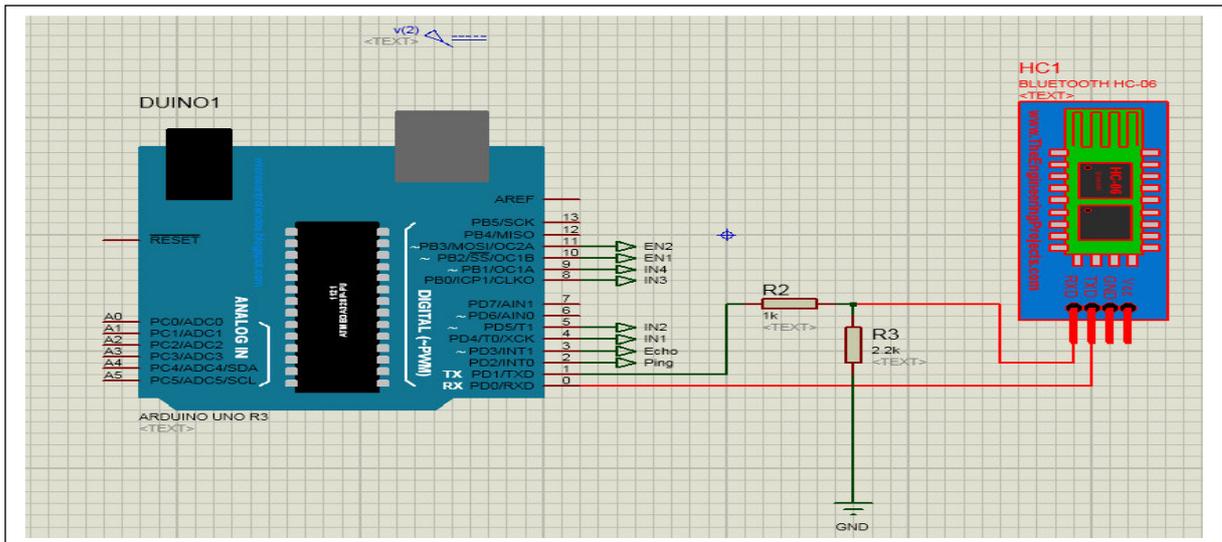


Figure II.7: Montage module HC-06

II.3 Actionneurs

II.3.1 C'est quoi un actionneur ?

Un actionneur est un dispositif qui transforme l'énergie délivrée par l'interface de puissance, en énergie utilisable par les effecteurs de processus. Le schéma fonctionnel d'un actionneur ainsi que les schémas des principaux actionneurs sont représentés à la figure ci-dessous. Les moteurs électriques, les vérins pneumatiques ainsi que les éléments chauffants sont des exemples typiques d'actionneurs utilisés en automatisation industrielle.

Les effecteurs : un effecteur est un dispositif qui transforme l'énergie délivrée par un actionneur, en valeur ajoutée. Selon le cas, un effecteur peut être séparé ou non de l'actionneur. Quelques exemples d'effecteurs rencontrés en milieu industriel sont : les ventilateurs, les broyeurs, les pinces à outils de robots articulés, dispositifs de transfert de chaleur [12]

II.3.2 Familles d'actionneurs

On distingue 3 grandes familles d'actionneur :

II.3.2.1 Actionneurs électriques

Il existe plusieurs actionneurs électriques on site :

- Moteur asynchrone
- Moteur pas à pas
- Moteur à courant continu
- Résistance chauffante
- ...etc.

II.3.2.2 Actionneurs pneumatiques

Les actionneurs pneumatiques sont utilisés principalement pour des mouvements séquentiels simples ils utilisent de l'air comprimé à 6 bar et permettent de réaliser des vérins dont la force peut atteindre 50000N. L'air est fourni par un compresseur, complété de filtres, d'un séparateur d'eau et d'un déshuileur. Il est souvent produit pour tous les ateliers, et distribué à toutes les machines. Les actionneurs sont généralement des vérins linéaires, mais aussi des moteurs rotatifs. On utilise également des aspirateurs suceurs à vide pour saisir des objets. Ils sont commandés en tout ou rien par des distributeurs, actionnés mécaniquement ou électriquement, on prend comme exemple :

- Vérins pneumatiques
- Techniques du vide (Tventouses)
- Muscle pneumatique

II.3.2.3 Actionneurs hydrauliques

Un actionneur hydraulique est composé de deux parties : une partie vérin et une partie valve. Dans un système à commande proportionnelle, on place dans l'étage de pilotage (valve) une bobine (solénoïde) qui régule la pression du fluide proportionnellement à la consigne électrique (courant tension).[13], on prend comme exemple :

- Vérins hydrauliques
- Pompes

II.3.3 Actionneur utilisé dans la réalisation

II.3.3.1 Moteur à courant continu

a .C'est quoi un Moteur à courant continu ?

Mcc utilisé actuellement pour la traction de véhicules : chariots élévateurs, chariots filoguidés et pour obtenir les déplacements des organes de machines à commande numérique.[14]

b. Constitution

- **Le stator** : ou inducteur C'est un électro-aimant ou un aimant permanent. Il possède deux pôles fixes N et S (Nord et Sud). Son circuit est appelé : circuit d'excitation.
- **Le rotor** : ou induit Il est constitué d'un nombre important de bobinages indépendants. A l'extrémité du rotor, chacune des extrémités de chacun des

bobinages est reliée à une lame du collecteur. Les lames du collecteur sont isolées entre elles.

- **Les balais** : Appelés couramment « charbons » Ce sont des morceaux de carbone (prismatiques) qui assurent la liaison électrique entre les fils d'alimentation de l'induit et les bobinages par l'intermédiaire du collecteur.

c. Représentation d'un moteur à courant continu

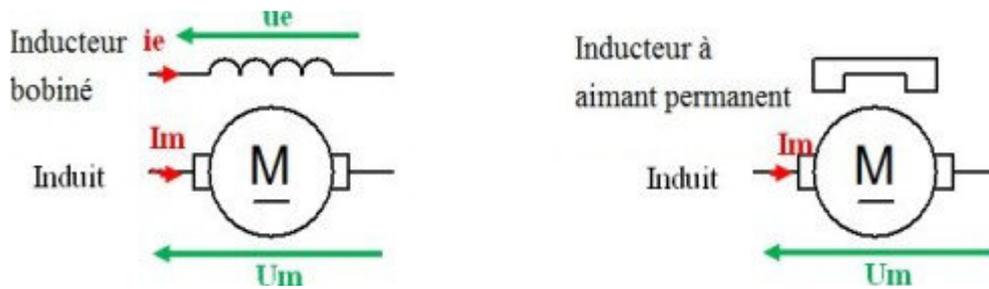


Figure II.8 : Représentation d'un moteur à courant continu

d. sens de rotation

Il est défini par les polarités de l'inducteur et de l'induit. Pour changer le sens de rotation de ce moteur, il suffit d'inverser la polarité de l'inducteur ou de l'induit.

e. types de moteur à courant continu

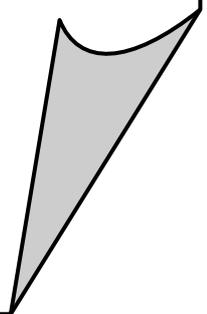
De construction, les moteurs à courant continu peuvent être :

- Excitation indépendante : inducteur et induit sont séparés
- Moteurs « série » : inducteur et induit sont en série
- Moteurs « dérivation » : inducteur et induit sont en parallèle

II.4 Conclusion

On peut dire à la fin de ce chapitre que toute système automatisé ou installation industrielle est équipée d'un ensemble de capteurs et actionneurs dont l'utilité consiste à contrôler des différents paramètres physiques telle que: la pression, le débit, la température,..., etc. Donc la compréhension des principes de fonctionnement des principaux capteurs et actionneurs utilisés est une phase très importante lors de l'étude de n'importe qu'el système automatisé.

***Microcontrôleur
&
Arduino***



III.1 Introduction

Le terme micro-ordinateur est utilisé pour décrire un système qui comprend au moins un microprocesseur, une mémoire de programme(ROM), la mémoire de données (RAM), et une entrée-sortie (I / O) périphérique. Certains systèmes de micro-ordinateurs comprennent des composants supplémentaires tels que des Timers, des compteurs et des convertisseurs analogique-numérique. Ainsi, un système de micro-ordinateur peut être existé sous plusieurs sortes à partir d'un gros ordinateur ayant disques durs, disquettes, imprimantes et à contrôleur intégré à puce unique. {Citation}

Dans ce chapitre, nous allons considérer seulement le type de micro-ordinateurs qui se composent d'une seule puce de silicium. Ces systèmes de micro-ordinateurs sont aussi appelés microcontrôleurs, et ils sont utilisés dans de nombreux produits ménagers tels que les fours à micro-ondes, unités de contrôle à distance d'un TV, cuisinières, équipements HIFI, lecteurs CD, ordinateurs personnels, et les réfrigérateurs. De nombreux microcontrôleurs sont disponibles au marché.

III.2 microcontrôleur

III.2.1Bref historique sur les Microcontrôleurs

Depuis la création de microprocesseurs, 4, 8, 16, 32 bits. Les microprocesseurs / microcontrôleurs ont développé et sont apparu sur le marché.

Afin de clarifier la plus simple Les développements historiques dans les produits de microprocesseurs sont listés dans le tableau suivant :

Année	Microprocesseur / Microcontrôleur	Remarque
1971	Intel 4004	Microprocesseur à 4 bit
1972	Intel 4040	Microprocesseur à 4 bit
1974	Intel 8080	Microprocesseur à 8 bit
	TMS 1000	Microcontrôleur à 8 bit
1975	Motorola 6800	Microprocesseur à 8 bit
1976	MCS-48	Microcontrôleur à 8 bit
	Intel 8085	Microprocesseur à 8 bit
1978	8686, Motorola 68000, Zilog Z-8000	Microprocesseur à 16 bit
1980	8051	Microcontrôleur à 8 bit
1982	68010,6805	Microcontrôleur
	80186, 80188,80286	Microprocesseur à 16 bit
	8096(MCS-96)	Microcontrôleur à 16 bit
1984	86020	Microprocesseur à 32 bit
1985	80386	Microprocesseur à 32 bit

	PIC microcontrollers by Microchip	Microcontrôleur à 8 bit
1987	Z280	Microprocesseur à 16 bit
1989	80386sx, 80486	Microprocesseur à 32 bit
1993	Pentium	Microprocesseur à 32 bit
1997	Atmel 8-bit AVR family	Microcontrôleur RISC à 8 bit

Tableau III.1 : Développements historiques des microprocesseurs/microcontrôleur

En Juin 1997, les microcontrôleurs ATMEL 8 bits AVR ont été introduites qui ont également le jeu d'instructions réduit. Aujourd'hui, nous voyons un certain nombre de familles de microcontrôleurs, de sources multiples et dans de nombreuses versions.[21]

III.2.2 C'est quoi un microcontrôleur ?

Le **microcontrôleur** correspond au **cerveau du robot**. C'est lui qui va traiter les informations provenant des capteurs et qui va donner la réponse voulue aux servo-moteurs.

Le microcontrôleur est composé de **quatre parties** :

III.2.2.1 Microprocesseur

Qui va prendre en charge la partie traitement des informations et envoyer des ordres. Il est lui-même composé d'une unité arithmétique et logique(UAL) et d'un bus de données. C'est donc lui qui va exécuter le programme embarqué dans le microcontrôleur.

III.2.2.2 Mémoire de données

(RAM ou EEPROM) dans laquelle seront entreposées les données temporaires nécessaires aux calculs. C'est en fait la mémoire de travail qui est donc volatile.

III.2.2.3 Mémoire programmable

(ROM), qui va contenir les instructions du programme pilotant l'application à laquelle le microcontrôleur est dédié. Il s'agit ici d'une mémoire non volatile puisque le programme à exécuter est à priori toujours le même. Il existe différents types de mémoires programmables que l'on utilisera selon l'application. Notamment :

- a. OTPROM : programmable une seule fois mais ne coute pas très cher.
- b. UVPROM : on peut la préfacée plusieurs fois grâce aux ultraviolets.
- c. EEPROM : on peut la préfacée plusieurs fois de façon électrique comme les mémoires flash.

- **dernière partie** correspond aux ressources auxiliaires. Celles-ci sont généralement :
 - Ports d'entrées / sorties parallèle et série
 - Des timers pour générer ou mesurer des signaux avec une grande précision temporelle.
 - Des convertisseurs A/N pour traiter les signaux analogiques[22]

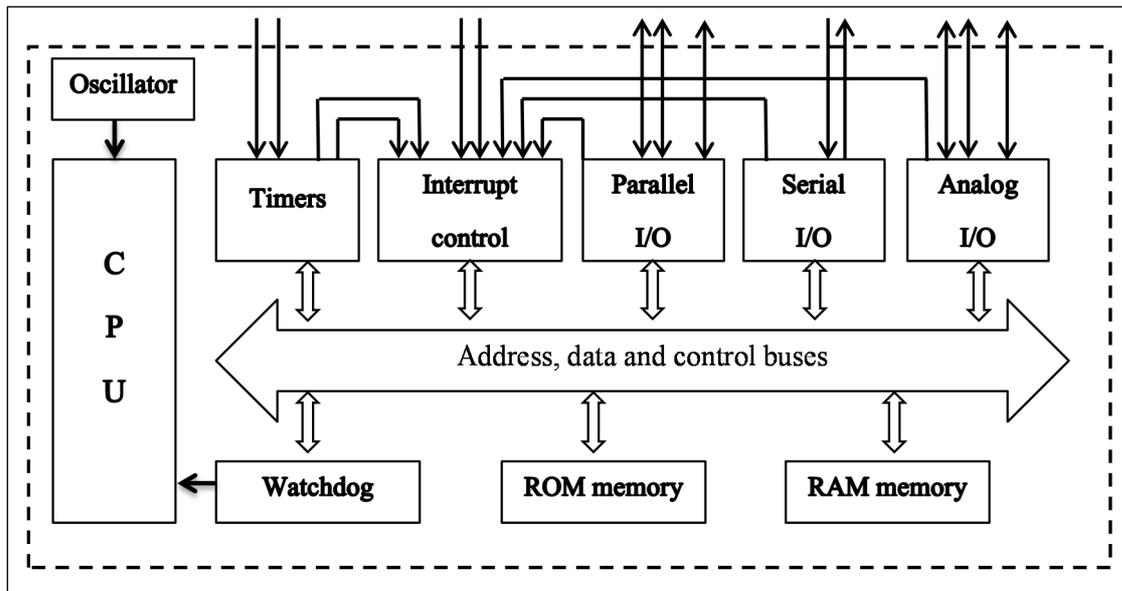


Figure III.1: Schéma des éléments principaux d'un microcontrôleur

III.2.3L'architecture des microcontrôleurs

Architecture de Von Neumann-(Princeton): quand un UC a une mémoire commune utilisé pour le programme et les données, on dit qu'il a l'architecture Princeton de mémoire dans le processeur.

Architecture de Harvard: quand la mémoire de programme se distingue de la mémoire de données, le UC a l'architecture de Harvard dans son processeur. Alors l'accès à chacune des deux mémoires se fait via un chemin distinct. Cette organisation permet de transférer une instruction et des données simultanément, ce qui améliore les performances.

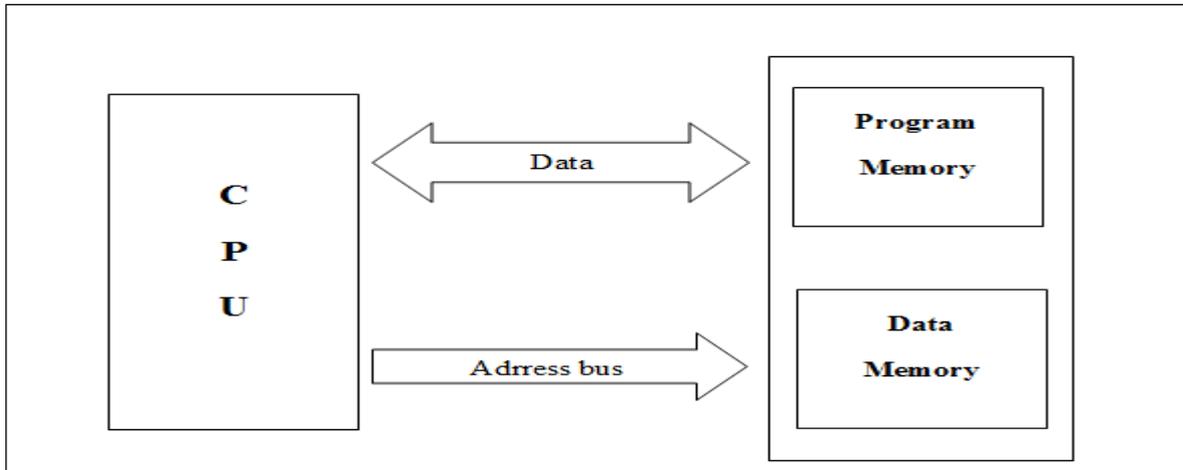


Figure III.2 : L'architecture d'un microcontrôleur -Von Neuman

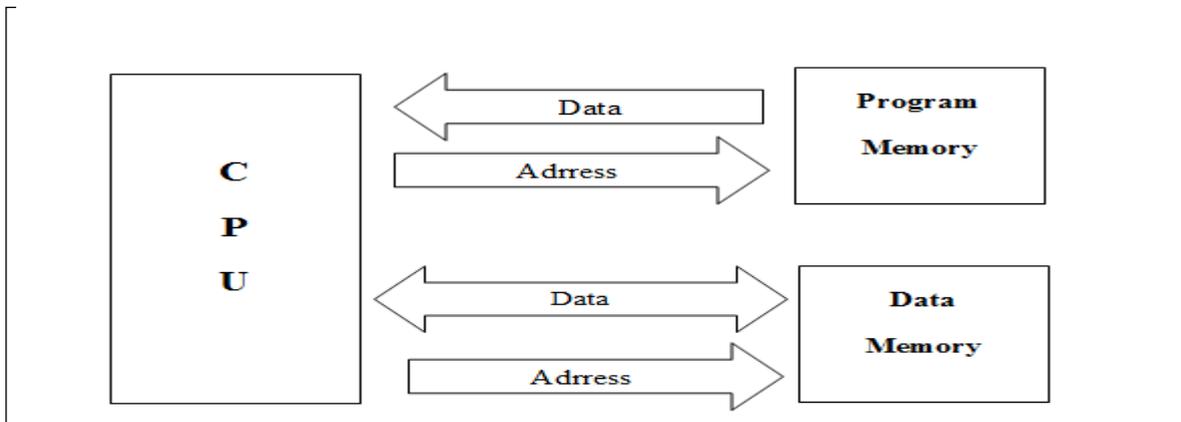


Figure III.3 : L'architecture d'un microcontrôleur -Harvard

III.2.3.1 RISC et CISC l'architecture des microcontrôleurs

l'architecture RISC (Reduced Instruction Set Computer) des microcontrôleurs: quand un microcontrôleur a un jeu d'instructions qui prend moins de modes d'adressage pour les instructions arithmétique et logique et pour les instructions de transfert de données, le microcontrôleur est dit être de l'architecture RISC. [23]

l'architecture CISC (instruction Complicated Set Computer) des microcontrôleurs: quand un MCU a un jeu d'instructions qui prend de nombreux modes d'adressage pour les instructions arithmétique et logiques et pour l'accès à la mémoire et des instructions de transfert de données de le UC est dit être de l'architecture CISC. [24]

III.2.4 Langage de programmation d'un microcontrôleur

III.2.4.1 C : le C est un langage de programmation impératif conçu pour la programmation système. Inventé au début des années 1970 avec UNIX, C est devenu un des langages les plus utilisés. De nombreux langages modernes comme C++, Java et PHP reprennent des aspects de C.[25]

Toutefois, les professionnels placent le langage C en tête de liste pour plusieurs raisons :[26]

- Il est souple et puissant
- Le langage C est modulaire. Son code peut être écrit sous forme de sous-programme appelés fonctions
- Lorsqu'il nouvelle architecture (nouveau processeur, nouveau système d'exploitation...) apparaît, le premier langage disponible est généralement le C car contrairement à d'autres, il est facile porter. De plus, un compilateur est souvent disponible sur les ordinateurs

III.2.4.2 C++ : dans les années 80 B. Stroustrup propose d'appeler C++ un nouveau langage, conçu non pas comme un remplaçant mais comme une amélioration du langage C.[27]

Comme le C, C++ adopte une vision très proche de la machine. Il à été destiné en premier lieu à l'écriture de systèmes d'exploitation mais ses caractéristiques lui ont ouvert d'autres perspectives.

III.2.4.3 Java : Java est un langage de programmation informatique de haut niveau. Langages de haut niveau, comme Java, permettent aux programmeurs d'écrire des instructions en utilisant des commandes en anglais. Chaque instruction dans un langage de haut niveau correspond à de nombreuses instructions dans la langue de la machine.[28]

III.2.4.4 L'assembleur : Le langage assembleur est très proche du langage machine (c'est-à-dire le langage qu'utilise l'ordinateur : des informations en binaire, soit des 0 et des 1). Il dépend donc fortement du type de processeur. Ainsi il n'existe pas un langage assembleur, mais un langage assembleur par type de processeur.

III.2.5 Domaine d'applications des microcontrôleurs

- Electronique de consommation: Appareil hifi, TV, vidéo, télécommande
- Moyens de transport: Autos, avions, vélo électrique
- Système de communication: Les microcontrôleurs 8 bits sont souvent utilisés pour les téléphones portables simples et la téléphonie fixe alors que les microcontrôleurs 32 bits se retrouvent plutôt dans les Smartphones et les PDA.
- Technique médicale: Les instruments de mesure (par exemple mesure de la glycémie), les organes artificiels
- Les technologies de la sécurité: Les systèmes pour gérer la sécurité dans les moyens de transport (par exemple : les passages à niveau), dans les bâtiments (par exemple: alarme incendie, effractions) etc.

III.3 Arduino

III.3.1 Bref historique de l'Arduino

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

En 2003, Hernando Barragan, pour sa thèse de fin d'études, avait entrepris le développement d'une carte électronique dénommée Wiring, accompagnée d'un environnement de programmation libre et ouvert. Pour ce travail, Hernando Barragan réutilisait les sources du projet Processing. Basée sur un langage de programmation facile d'accès et adaptée aux développements de projets de designers, la carte Wiring a donc inspiré le projet arduino

En 2005, était d'arriver à un dispositif simple à utiliser, dont les coûts seraient peu élevés, les codes et les plans « libres » & « multi-plates-formes » [29]

Le nom Arduino trouve son origine d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort ».[30]

III.3.2 C'est quoi Arduino ?

Arduino est un circuit imprimé en matériel libre (dont les plans de la carte elle-même sont publiés en licence libre mais dont certains composants sur la carte, comme le microcontrôleur par exemple, ne sont pas en licence libre) sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques.[31]

III.3.2.1 Application

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

- contrôler les appareils domestiques
- fabriquer votre propre robot
- faire un jeu de lumières
- communiquer avec l'ordinateur
- télécommander un appareil mobile (modélisme)
- etc.
- Dans notre projet, nous allons faire un robot mobile évitera les obstacles commander par Bluetooth.

III.3.2.2 Types de cartes

Il y a trois types de cartes :[32]

- Les dites « officielles » qui sont fabriquées en Italie par le fabricant officiel : *Smart Projects*.
- Les dites « compatibles » qui ne sont pas fabriquées par *Smart Projects*, mais qui sont totalement compatibles avec les Arduino officielles.
- Les « autres » fabriquées par diverse entreprise et commercialisées sous un nom différent (Freeduino, Seeduino, Femtoduino, ...).

III.3.3 Différentes cartes Arduino

Des cartes Arduino il en existe beaucoup, Comme on peut le voir sur le tableau ci-dessous, Aspects techniques illustrant les différences techniques :[33]

Microcontrôleur & Arduino

Carte Arduino/Caractéristiques	UNO R3(classique& CMS)	UNO R3 Ethernet(classique & POE)	Leonardo	Mega2560	MegaADK	DUE	Esplora	Mini	Nano	Yun(classique& POE)	Zero PRO
Microcontrôleur	ATmega328p	ATmega328p	ATmega32u4	ATmega2560	ATmega2560	AT91SAM3X8E	ATmega32u4	ATmega328p	ATmega328p	ATmega32u4	ATSAMD21G18
Cadencement horloge	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	84Mhz	16Mhz	16Mhz	16Mhz	16Mhz	48Mhz
Tension d'entrée	7-12v	7-12v	7-12v	7-12v	7-12v	7-12v	7-12v	7-9v	7-9v	5v	5v
Tension de fonctionnement	5v	5v	5v	5v	5v	3.3v	5v	5v	5v	5v	3.3v
Entrée/Sortie Numérique	14/6	14/4	20/7	54/15	54/15	54/12	X	14/6	14/6	20/7	14/12
Entrée/Sortie (PWM) Analogique	6/0	6/0	12/0	16/0	16/0	12/2 (DAC)		8/0	8/0	12/0	6/1 (DAC)
Mémoire vive (Flash)	32Ko	32Ko	32Ko	256Ko	256Ko	512Ko	32ko	32Ko	32Ko	32Ko	256Ko
Mémoire vive (SRAM)	2Ko	2Ko	2.5Ko	8Ko	8ko	96Ko	2.5Ko	2Ko	2Ko	2.5Ko	32Ko
Mémoire morte (EEPROM)	1Ko	1Ko	1Ko	4Ko	4ko		1Ko	1ko	1Ko	1Ko	16Ko
Interface USB	USB-Bmâle	USB-B mâle	Micro-USB	USB-Bmâle	USB-Bmâle& USB-A pour arduino	2 ports micro-USB(Native et programming)	Micro-USB	X	Mini-USB	Micro-USB	2 ports micro-USB (Native et programming)
Port UART	1	1	1	4	4	4	X	X	1	1	2
Carte SD	X	V	X	X	X	X	X	X	X	V	X
Ethernet	X	V	X	X	X	X	X	X	X	V	X
Wi-fi	X	X	X	X	X	X	X	X	X	V	X
Dimensions	68x53mm	68x53mm	68x53mm	101x53mm	101x53mm	101x53mm	165x60mm	30x18mm	45x18mm	68x53mm	68x53mm

Tableau III.2: Différentes cartes Arduino et leur Aspects techniques

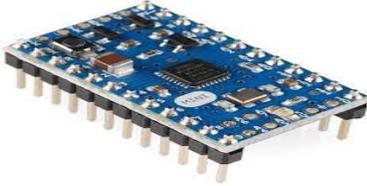
		
Arduino Uno	Arduino Mega	Arduino Yun
		
Arduino Mini	Arduino Nano	Arduino Zero

Figure III.4: Différent types des cartes arduino

Remarque : presque toutes cartes Arduino fonctionnent en 5V sauf l'Arduino DUE et l'Arduino Zero PRO qui fonctionnent en 3.3V.

III.3.4 Les raisons de choisir Arduino

Il existe pourtant dans le commerce, une multitude de plateformes qui permettent de faire la même chose. Notamment les microcontrôleurs « PIC » du fabricant Microchip. Nous allons voir pourquoi choisir l'Arduino

III.3.4.1 prix : En vue des performances qu'elles offrent, les cartes Arduino sont relativement peu coûteuses, ce qui est un critère majeur pour le débutant. ce qui est un bon rapport qualité/prix.

III.3.4.2 liberté : C'est un bien grand mot, mais elle définit de façon assez concise l'esprit de l'Arduino. Elle constitue en elle-même deux choses :

- a. **Logiciel :** gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris, Il peut être téléchargé directement à partir du lien suivant : [34]
- b. **Matériel :** cartes électroniques dont les schémas sont en libre circulation sur internet.

III.3.4.3 Compatibilité : Le logiciel, tout comme la carte, est compatible sous les plateformes les plus courantes (Windows, Linux et Mac), contrairement aux autres outils de programmation du commerce qui ne sont, en général, compatibles qu'avec Windows.

III.3.4.4 Communauté : La communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet. De plus, on trouve les références du langage Arduino ainsi qu'une page complète de tutoriels sur le site : <https://www.arduino.cc>

Remarque: Finalement, nous retiendrons ce projet pour toutes ses qualités.

III.3.5 Les outils Arduino

III.3.5.1 Matériel

Il existe plusieurs variétés de cartes Arduino. La figure ci-dessous montre par exemple, la dernière version de la carte Arduino : la « Uno », sortie en 2010.[35], ce qui est utilisée dans notre projet.

a. Constitution de la carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEGA, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

b. Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique.

Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée

en langage C. la figure I.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino[36]



Figure III.5: Microcontrôleur ATmega328

c. Entrées / sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite()` (broche, HIGH).

En plus, certaines broches ont des fonctions spécialisées :

- **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. - Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la bibliothèque pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (Ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la bibliothèque `Wire/I2C` (ou TWI - Two-Wire interface - interface "2 fils").
- **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité).

Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.[37]

d.ports de communications

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

Côté entrées, des capteurs qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence ou un accéléromètre, le contact via un bouton-poussoir, etc...

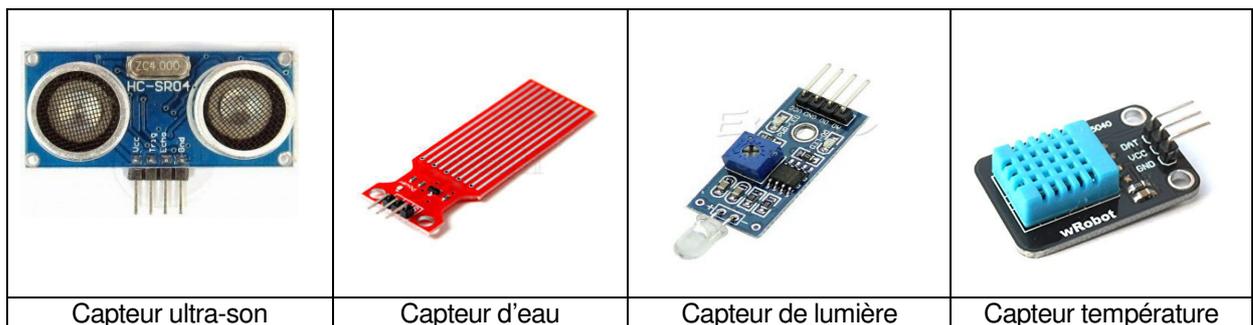


Figure III.6: Différent types des capteurs pour l'arduino

Côté sorties, des actionneurs qui agissent sur le monde physique telle une petite lampe, un moteur, haut-parleur.

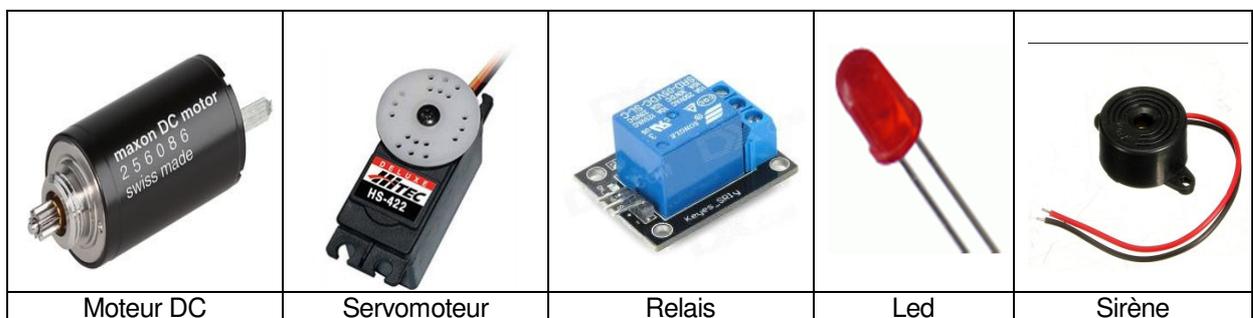


Figure III.7: Différents actionneurs pour l'arduino

III.3.5.2 Logiciel

C'est un logiciel de programmation par code, code qui contient une cinquantaine de commandes différentes. A l'ouverture, l'interface visuelle du logiciel ressemble à ceci:

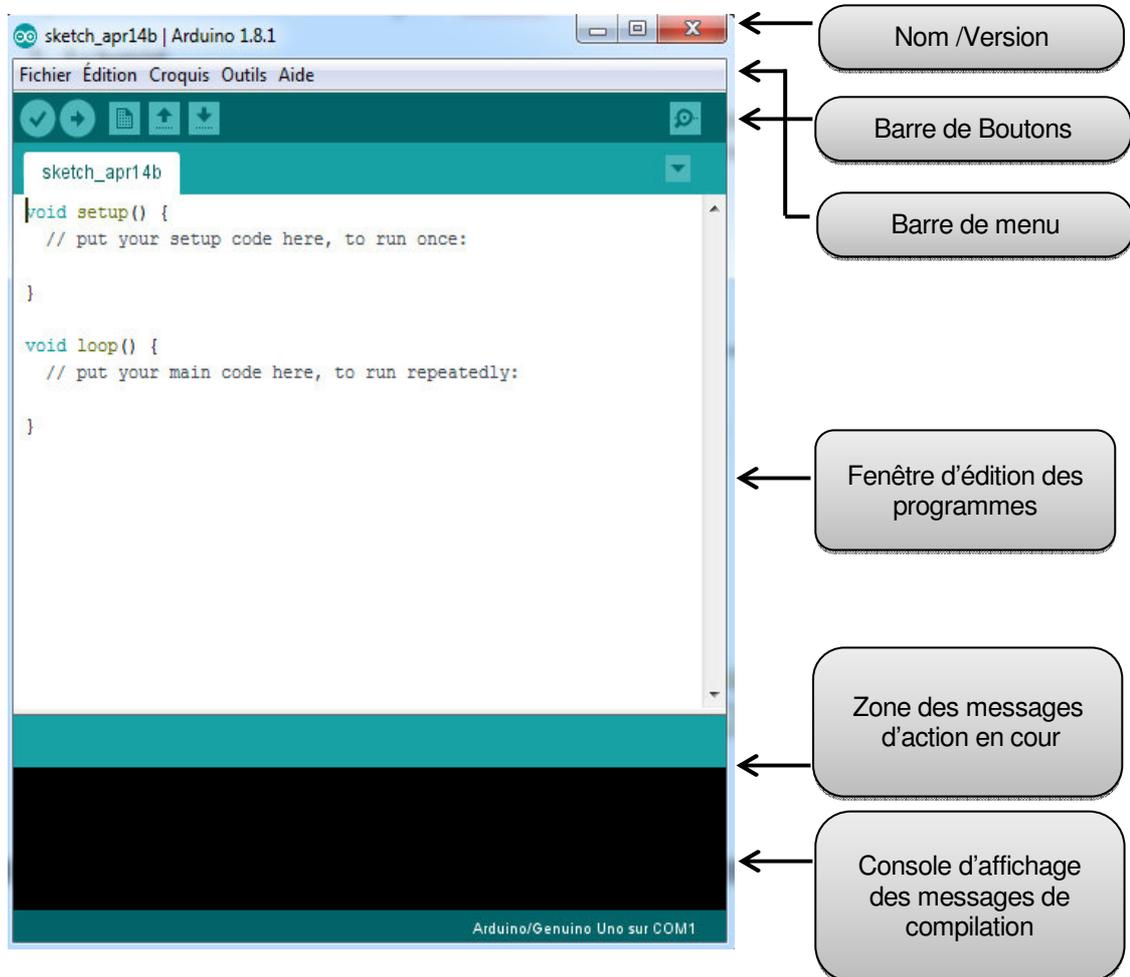


Figure III.8 : Interface de logiciel arduino

Cette fenêtre vide sera remplie de mots et de chiffres et d'autres textes. Cette fenêtre est comme tout autre logiciel que vous avez utilisé. Elle contient des menus, des boutons, des alertes spéciales, et toutes sortes de contrôles: Ouvrir, Enregistrer, et le bouton du moniteur de série à l'extrême droite

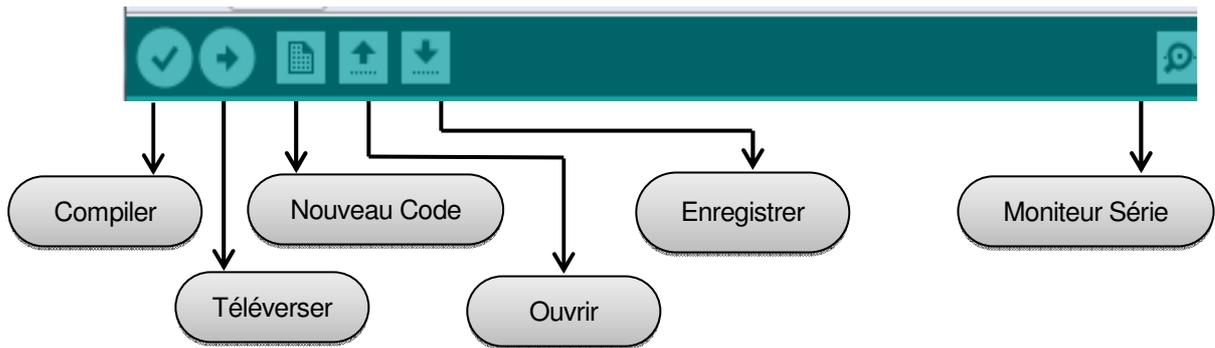


Figure III.9 : Détail de barre des boutons

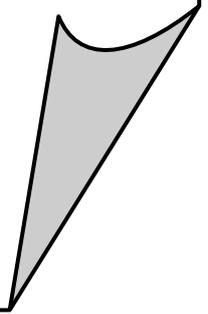
III.3.6 Exemples d'usage

- Prototypage rapide de projets innovants utilisant l'électronique, Arduino facilitant l'expérimentation en amont de la phase d'industrialisation
- Production artisanale d'objets numériques et de machines-outils à faible coût dans la perspective d'une culture d'appropriation technologique favorisant le bricolage et la débrouille
- Projets pédagogiques à destination d'étudiants, de professionnels ou du grand public selon les porteurs de ces initiatives : écoles supérieures, centres de formation spécialisée ou des Media Labs

III.4 Conclusion

À travers ce chapitre on peut dire que lors de la conception d'un circuit électronique, si celui-ci nécessite une unité de calcul, l'implantation de celle-ci est soit un assemblage de portes logiques (programmation matérielle), soit une carte arduino comme le cas de notre travail (programmation logicielle). Les premiers ont un très faible coût de fabrication s'ils sont produits en très grande quantité. L'avantage des seconds est qu'ils sont de toute manière fabriqués en masse afin de les rendre le plus accessible possible en réduisant au maximum leur prix, et il suffit d'y embarquer un logiciel pour qu'ils puissent accomplir une tâche spécifique, l'un de ses logiciels est l'arduino, qui représente un logiciel de programmation par code basé sur des cartes électroniques a microcontrôleur open source et qui peut être utilisé pour construire des objets interactifs indépendants (prototypage rapide), ou bien peut être connecté à un ordinateur pour communiquer et superviser en utilisant des logiciels de programmation (flash, labview, ...etc).

*Conception
&
Réalisation
de
notre outil*



IV.1 Introduction

Ce chapitre nous permet de montrer les résultats de notre projet dans sa phase de test et d'essai, ce qui nous permettra d'envisager les améliorations possibles.

Le but essentiel de notre travail est d'utiliser l'arduino pour commander le robot mobile. On doit construire un prototype de ce robot ,de tel sort qu'on va obtenir à la fin un boîtier qui circule à l'aide des trois roues, et pour pouvoir commander ce boîtier on doit le connecter à des circuits d'alimentation, de commande et des capteurs puis a une carte arduino qui contient le programme qui nous permet de gérer toute les applications de ce robot et le contrôler à distance par une application installée sur n'importe quel Smartphone ,PC ou Tablette ,à condition que le Bluetooth soit activé.

Notre réalisation pratique a été faite en deux parties:

- La première partie est la conception de tout le système électronique.
- La deuxième partie est la création d'une application androïde basée sur le langage de programmation par bloc.

IV.2 Les outils utilisés pour le développement

Nous présentons dans cette section les outils utilisés pour le développement de notre robot & application (Proteus, Arduino, MIT app Inventor et Enterprise Architect).

IV.2.1 Proteus Professionnel

Il s'agit d'un des ensembles d'outils numériques les plus complets du marché car la version 8.5 (la dernière) permet de créer sur le PC toute sorte de PCB ou circuits imprimés parmi presque 800 microcontrôleurs différents, et simuler leur fonctionnement réel directement sur la vue schématique du circuit. Et bien évidemment, il intègre les outils nécessaires pour le dessin et la simulation de l'environnement Arduino, une des plaques les plus populaires de nos jours. Peut être téléchargé directement à partir du lien suivant :[38] <https://www.labcenter.com>

IV.2.1.1 Composants principaux de Proteus

Ce logiciel dispose de deux composants principaux autour desquels tourne tout le fonctionnement du même :

a.ISIS

Ce sont les sigles d'Intelligent Schematic Input System et c'est le programme qui permet de dessiner sur le plan du circuit électrique, avec toute sorte de composants comme des résistances, des bobines, des condensateurs, des sources d'alimentation ou même des microprocesseurs.

b.ARES

Ce sont les sigles d'*Advanced Routing and Editing Software* et c'est l'outil consacré au dessin de circuits imprimés ou PCB, avec des options de localisation et d'édition de composants électriques.

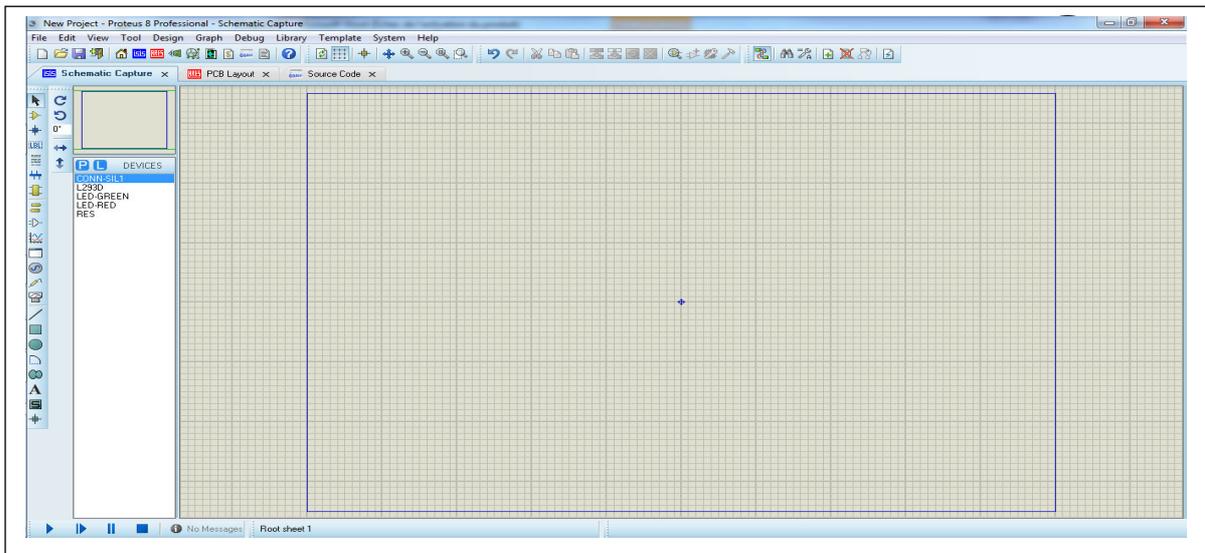


Figure IV.1 : Fenêtre Proteus professionnelle v8.1 (ISIS)

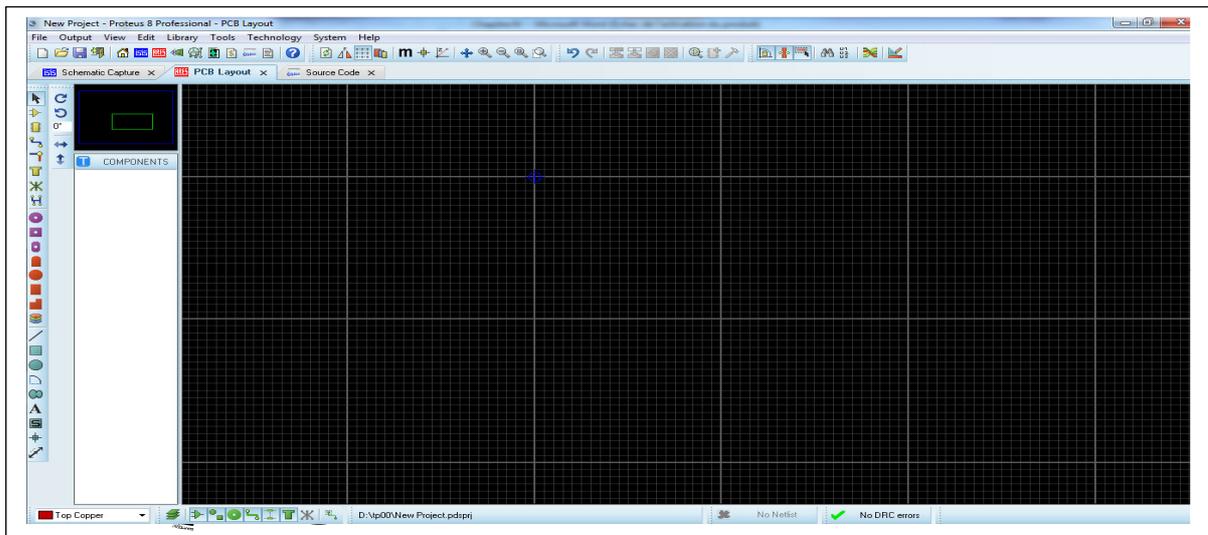


Figure IV.2 : Fenêtre Proteus professionnelle v 8.1 (ARES)

Conception & Réalisation de notre outil

N°	Nom	Description	Remarque
1 & 5	Barre d'outils	Elle est composée d'un ensemble d'icônes dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc.	Isis et Ares
2	Espace de travail	La surface la plus grande de l'écran s'appelle "Fenêtre d'édition" et se comporte comme une fenêtre de dessin. C'est là que vous placez et câblez les composants	Isis et Ares
3	Vue d'ensemble	Fenêtre ou Vous pouvez déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.	Isis et Ares
4	Liste des composants	Contient tous les composant utiliser	Isis et Ares
6	Boutons de simulation	4 boutons « Run, Stop, Pause, advance the simulation by one animation frame	Isis
	Couche sélectionné	Liste contenant types de couches des différents circuits	Ares

Tableau IV.1 : Tableau explicatif « interface ISIS & ARES »

IV.2.2 Arduino

Nous avons utilisé le langage de programmation arduino pour programmer la carte arduino du type « Uno » que nous avons traité au troisième chapitre.

IV.2.3 MIT app Inventor

App Inventor pour Androïde est une application développée par Google a base de java. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT).

MIT App Inventor est une initiation novatrice à la création de programmation et d'applications qui transforme le langage complexe du codage basé sur le texte en blocs de construction visuels, glisser-déposer. L'interface graphique simple accorde même à un novice inexpérimenté la possibilité de créer une application, il suffit de connecter avec un compte Google pour accéder à cette application.

L'environnement App Inventor contient trois fenêtres qui sont proposées pendant le développement :

- 1^{ère} fenêtre : pour la création de l'interface homme machine, permet de créer l'allure de l'application (App Inventor Designer) .

- 2^{ème} fenêtre pour la programmation par elle-même : elle permet, par l'assemblage des blocs de créer le comportement de l'application (App Inventor Block Editor).
- 3^{ème} fenêtre pour l'émulateur : qui permet de remplacer un terminal réel pour vérifier le bon fonctionnement du programme.

La connexion d'un terminal réel sous Androïde permettra ensuite de télécharger le programme pour un test réel. Ce terminal pourra aussi bien être un smart phone, le comportement du programme sera identique. [39]

Remarque : Une application exécutable sous Androïde (interprétable par une interface en JAVA) est un fichier avec l'extension « APK ».

IV.2.3.1 Commencement avec App Inventor

Google fournit gratuitement un kit de développement (SDK) prévu pour s'intégrer (sous la forme d'un Plug-in) à l'environnement de développement Eclipse (libre). Il permet de développer des applications codées en langage Java pour les différentes versions Androïde

1. Se connecter à Internet.
2. Ouvrir notre navigateur et se connecter au compte Google.
3. Se connecter au site Internet d'App Inventor du MIT : [40]

a.Interface graphique

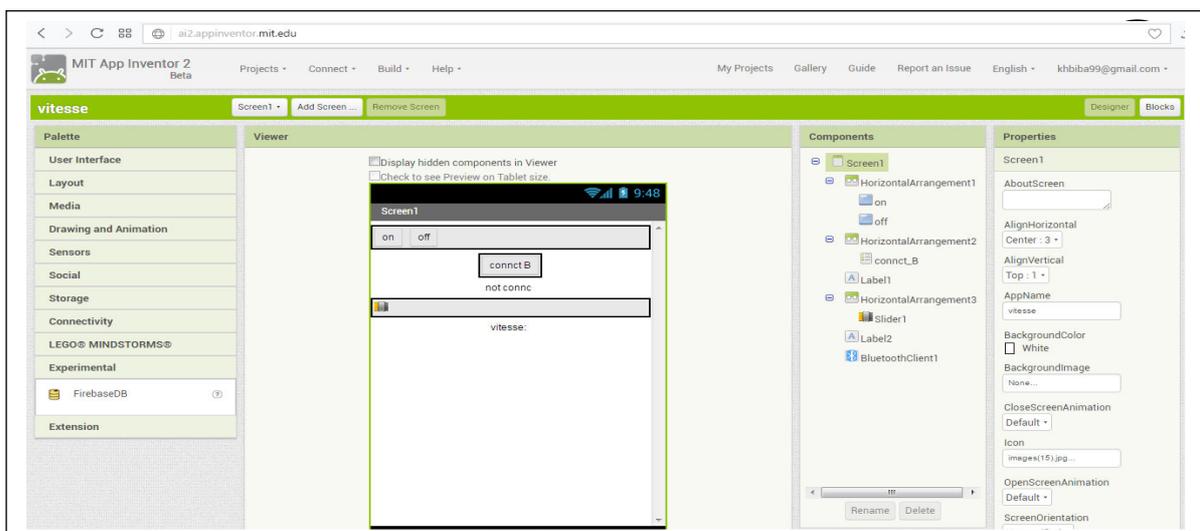


Figure IV.3 : Interface design MIT app Inventor2

N°	Nom	Description
1	Adresse Gmail	Pour créer l'application sous App Inventor il faut se connecter avec un compte Google
2	Barre menu	Où on peut créer, supprimer ou sauvegarder ... un projet
3	palette	Une palette sous App Inventor contenant tous les éléments qui peuvent être positionnés sur l'écran du smart phone
4	Zone du travail	C'est la surface du smart phone ajusté automatiquement par app inventor ou manuellement par nous-mêmes en utilisant le composant « Screen arrangement »
5	Liste des composants	La liste des éléments et des medias utilisés sur l'écran, Cette partie contient des composants (visibles ou invisibles)
6	Propriétés des Composants	Les propriétés des différents éléments utilisés par exemple la couleur ,la taille, image,etc du bouton ou texture ou autre

Tableau IV.2 : Tableau explicatif « interface MIT app Inventor2 »

b. Interface blocks (*Editeur de blocs*)

Une fois les composants de l'écran de smart phone mis en place et désigné, nous passons à la deuxième phase de développement d'une application via App Inventor : l'interface Scratch, pour cela, il faut cliquer sur «Open the Blocks Editor» en haut à droite de la page.

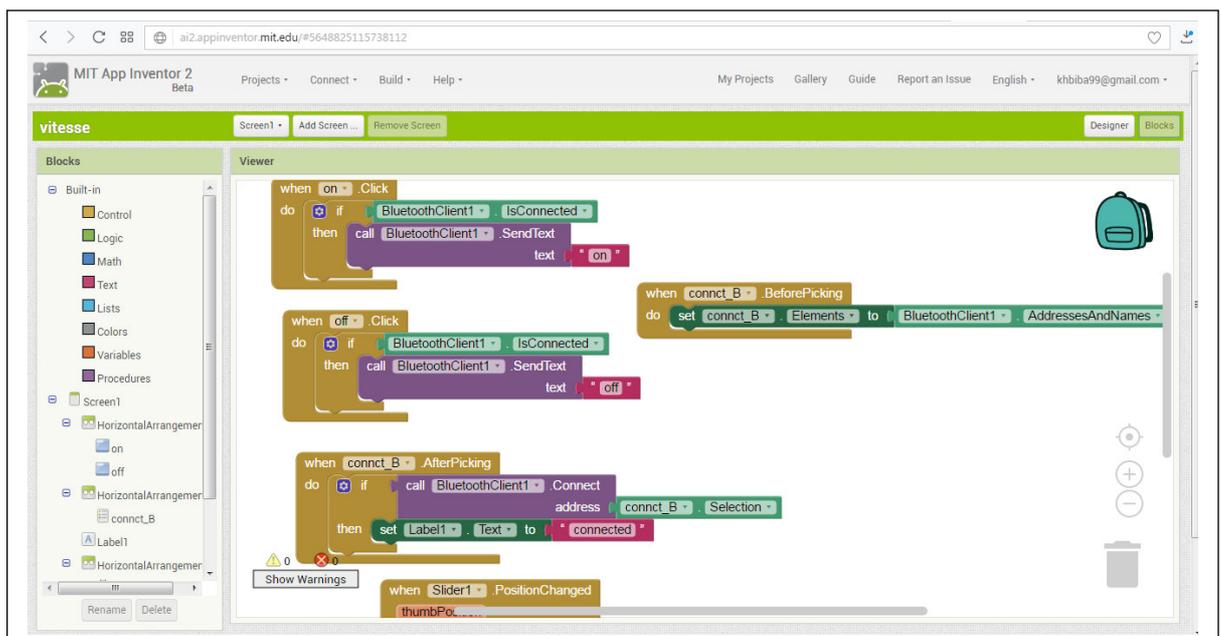


Figure IV.4: Interface Blocks MIT app Inventor2

IV.2.4 Enterprise Architect

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systèmes. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus, Il peut simplement être téléchargé à partir du lien suivant : [41]

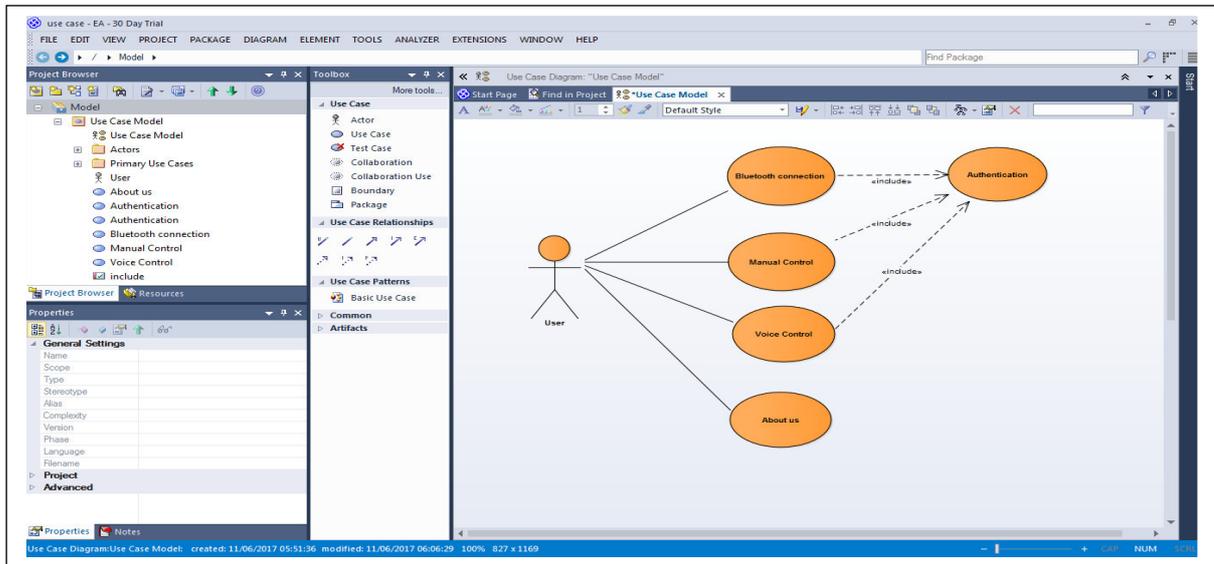


Figure IV.5 : Fenêtre Enterprise Architect

IV.3 Partie électronique

Dans la 1^{ère} partie « réalisation pratique », on passe par les deux étapes suivantes :

- Présenter les différents composants de la réalisation pratique de notre robot.
- On assemble ensuite les composants suivants notre montage sur la plaque d'essai.
- programmation de notre carte arduino

IV.3.1 Schéma bloc

La figure suivante représente les étapes qui constituent notre robot mobile

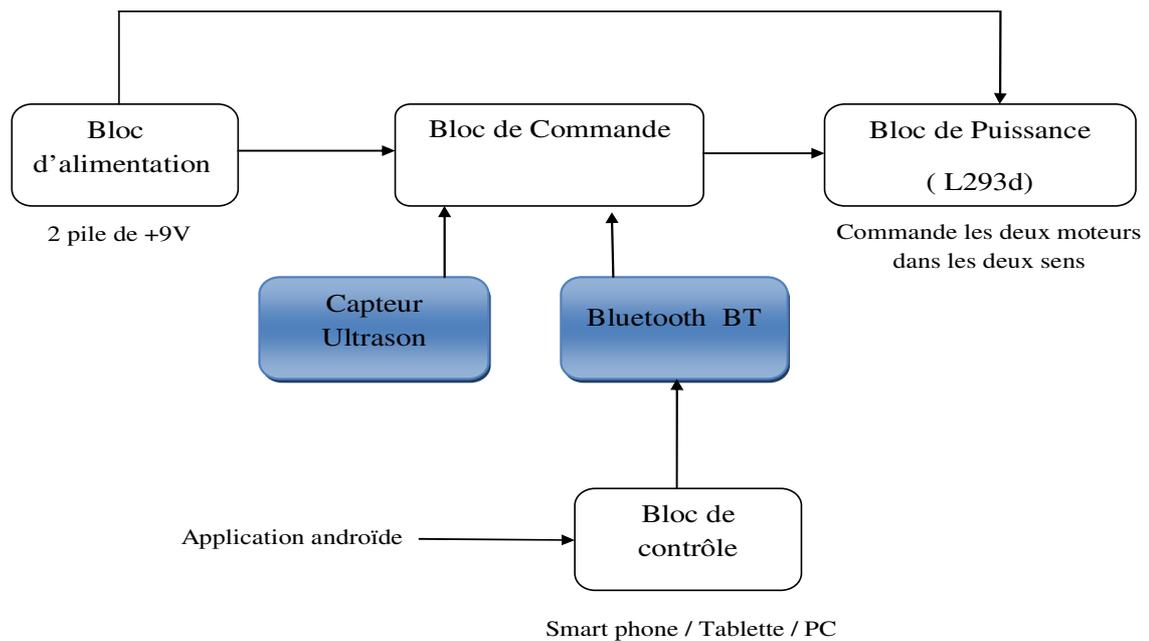


Figure IV.6 : Schéma bloc

IV.3.1.1 Bloc de traitement et de contrôle

Bloc concerné est le Smart phone, nous allons réaliser dans la section qui suit une application à multi Screen gérer la commande de deux moteurs (autoguidage) en deux modes « manuel - vocal » à travers le Bluetooth.

IV.3.1.2 Bloc de commande

Notre bloque de commande, on le résume tout simplement par l'utilisation du module Arduino UNO qui est détaillé précédemment au chapitre III.

IV.3.1.3 Bloc de puissance

Le moteur utilisé est alimenté avec +5V. La carte arduino délivre une tension de 5V « tension USB », ce qui suffit pour mettre le moteur en marche « moteur à faible puissance », l'inversion du sens de rotation du moteur nécessite un circuit qui inverse les pôles du moteur.

Nous avons pensé à utiliser un hacheur, ce qui engendre un encombrement du montage, ceci nous a conduits à choisir spécialement le L293D qui permet de commander un moteur à courant continu (CC) en marche « Avant - Arrière », grâce à 2 broches numériques de la carte Arduino.

Conception & Réalisation de notre outil

Donc, notre bloc de puissance se base sur un circuit intégré qui est le driver « L293D ». Ce bloc contient deux PINs qui s'allument et s'éteignent à savoir le signal émis par la carte Arduino.

Dans cet étage, nous avons utilisé un modèle de pont-H vraiment robuste, il existe le L293D, qui peut être tout à fait convenable.

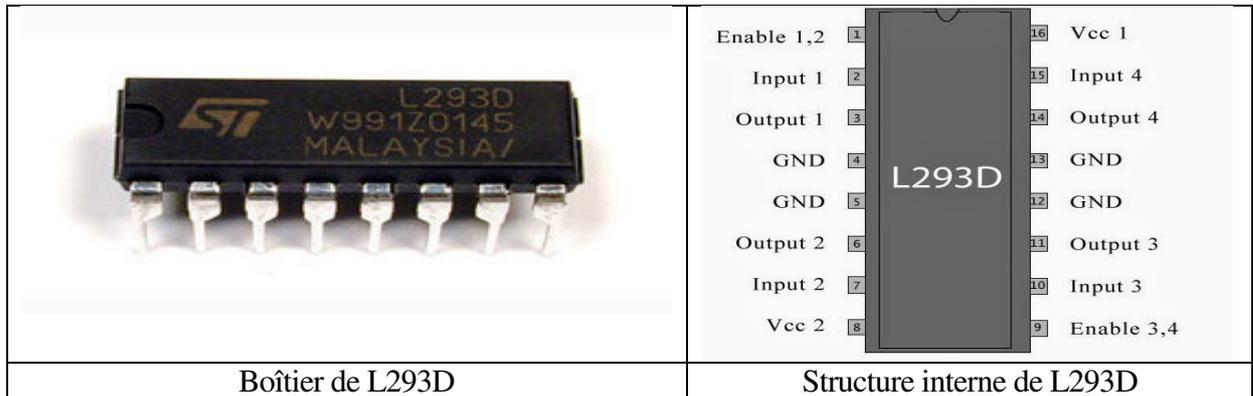


Figure IV.7 : Schéma de brochage L293D

Ce circuit est appelé « double pont en H », il permet d'inverser la polarité aux bornes des moteurs en fonction de l'état des broches numériques d'entrées et il est composé de deux étages identiques appelés ici A et B. Voici le brochage du L293D monté sur un DIL 16, figureIV.7

Les principes de montages du L293D sont rigoureusement identiques à ceux du L293, ce qui facilite grandement son utilisation seul le brochage diffère, parmi les avantages qui nous ont laissé choisir ce composant :

- L293 supporte une tension de puissance jusqu'à 36 volts maximum
- Un courant de service de 2A. supporte les pics occasionnels à 3A et pics répétitifs a 2.5A
- Compatible TTL (peut donc être commandé directement avec Arduino)
- Dispose d'un dispositif de mesure du courant (sensor/sense pins). A raccorder à la masse si on ne l'utilise pas
- Chute de tension **V_{ce_sat}** totale de 1.8 volts (typique), 3.2v à IL=1A, 4.9v à IL=2A. Si cela semble beaucoup, c'est aussi un avantage car cela permet d'utiliser un accu de 7.2v directement avec un moteur +5V. La chute de tension **V_{ce_Sat}** du Pont-H sera suffisante pour adapter la tension au moteur (généralement +/- 5V).

Conception & Réalisation de notre outil

Ce composant dispos de 16 broches, on peut éclaircir son brochage sous forme de tirets comme suit :

- **VSS** - Alimentation de la logique de commande (+5V). A raccorder à la borne +5V d'Arduino (donc sur le régulateur d'Arduino) ;
- **VS** - Alimentation de puissance des moteurs ;
- **GND** - Doit être raccordée à la masse (GND) de la source d'alimentation de puissance VS (donc le négatif de l'accumulateur) et à la masse de la source d'alimentation de VSS (donc GND Arduino) ;
- **OUTPUT1, OUTPUT2** - Broches à raccorder à la charge (le moteur). Via ces broches que le L293D commande le sens de rotation du moteur.
- **INPUT1, INPUT2** - Broches de commande du Pont-H. Se raccorde a Arduino.
- **ENABLE A** - (Chip Enable) permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 & OUTPUT2. **ENABLE. A** commande l'activation du premier Pont-H. Si **ENABLE A** = GND, le pont-H est déconnecté et le moteur ne fonctionne pas. Si **ENABLE A** = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquées sur I1 & I2.
- **CURRENT SENSING A & CURRENT SENSING B** - permet de faire une mesure du courant dans le circuit de puissance. A placer impérativement sur GND si cette fonctionnalité n'est pas utilisée.

Pour maintenir l'étage actif en permanence avec le carte Arduino et assurer sa synchronisation et son bon fonctionnement, sa broche Enable A (V_{en}) sera connectée directement au +5V venant du circuit de validation.

Le moteur est connecté sur les deux broches output «analogiques», avec des diodes de protection. La table de vérité du circuit est représentée par le tableau suivant:

Entrée		Fonction
$V_{en}=H$	C=H;D=L	Tournant à droite
	C=H;D=H	Tournant à gauche
	C=D	Arrêt du moteur
$V_{en}=L$	C=X;D=X	Arrêt du moteur

Tableau IV.3 : Table de vérité de l'étage de puissance

a.Réalisation de driver L293D

Dans la figure IV.8 on peut voir les schémas de la carte de puissance à base de l293d coté cuivre et coté composants réalisée par le logiciel de simulation PROTEUSE

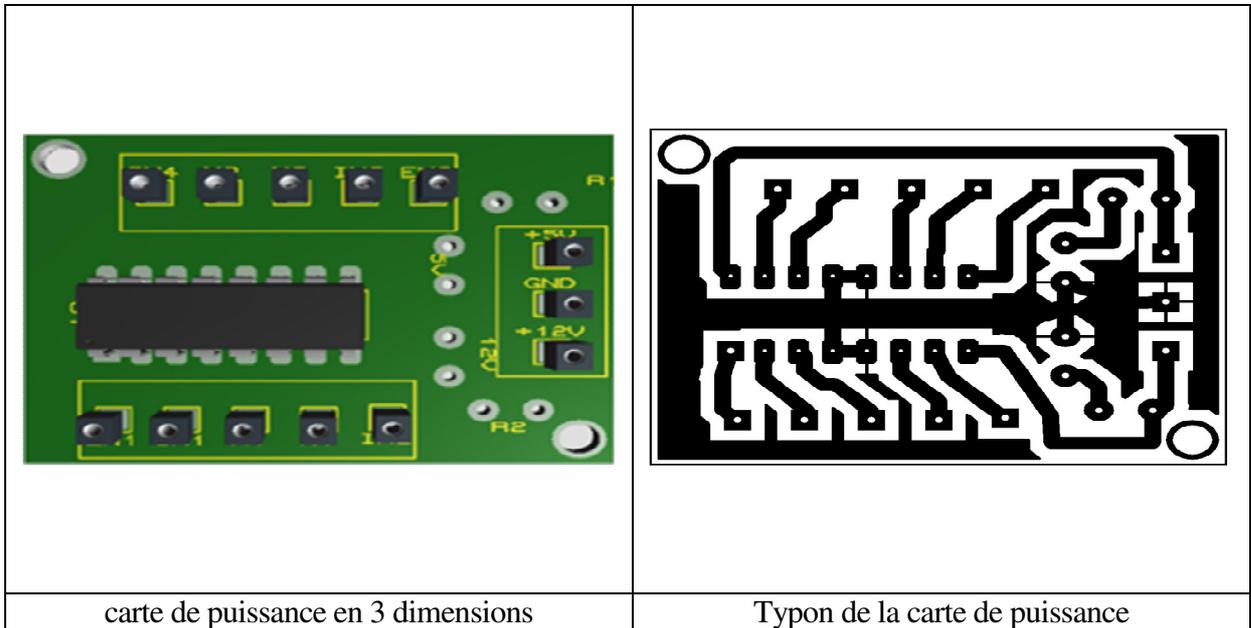


Figure IV.8 : Driver L293D réalisée par le logiciel proteus

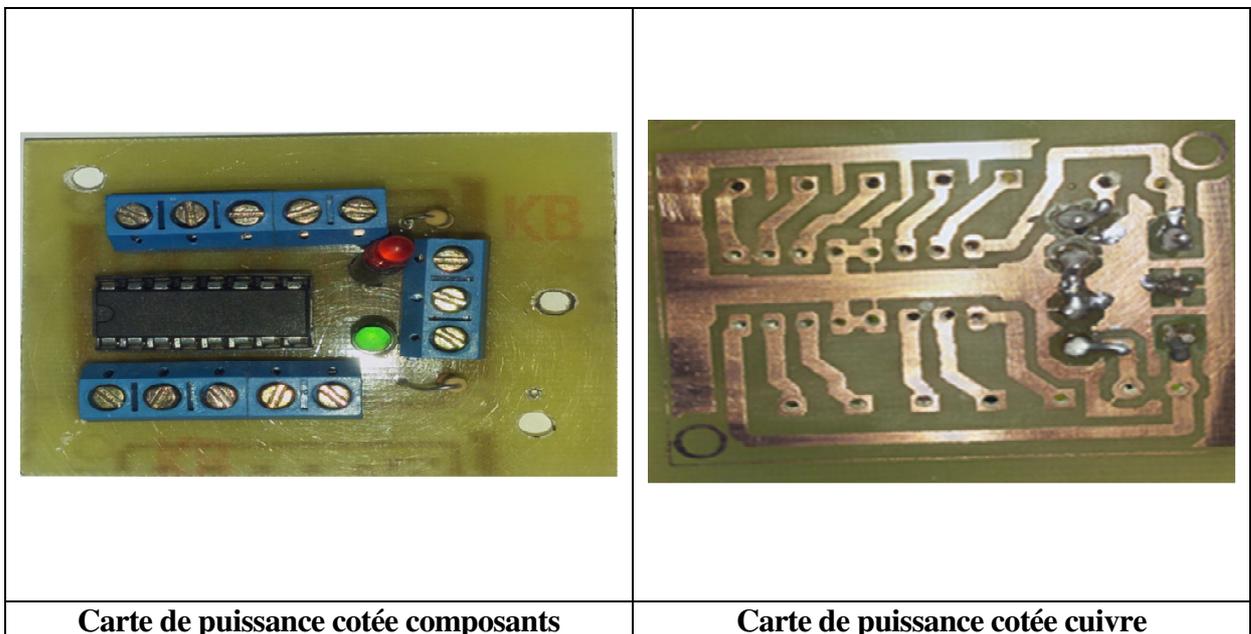


Figure IV.9 : Driver L293D en réalité

IV.3.1.4 Bloc d'alimentation

On a utilisé deux piles de 9V une pour le la commande du moteur et l'autre pour alimenter la carte arduino UNO.

IV.3.1.5 Capteur Bluetooth

Dans cette partie, on peut classer l'accessoire Bluetooth de l'Arduino comme une suite de blocs de commande puisque il prend la relève de la validation des signaux émis par le smart phone vers l'arduino ; donc, il a besoin d'une configuration lors de la programmation. Nous avons utilisé un modèle nommé **HC-06**.

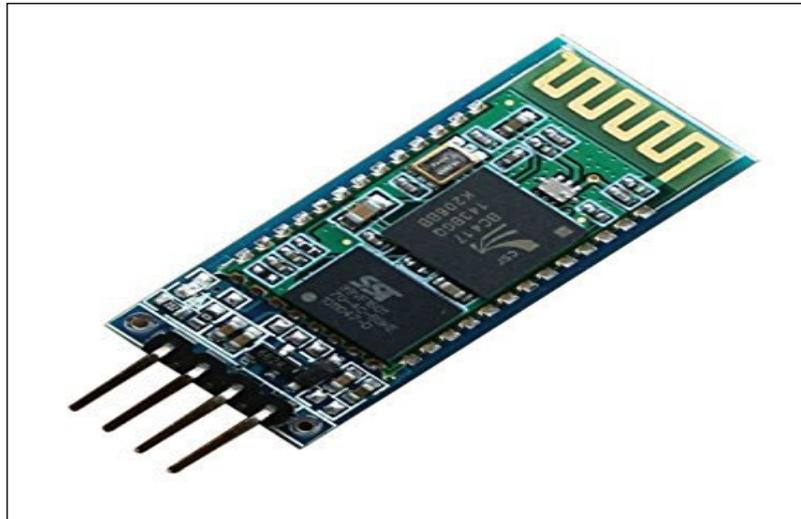


Figure IV.10: Capteur Bluetooth Module HC-06

Avant de passer à la configuration du Bluetooth, il faut réaliser quelques étapes et ce comme suit :

- Enlever toutes les communications
- Placer la position de fonctionnement **DAT** de commutateur
- Relier le port d'USB à l'ordinateur
- Télécharger le programme ou ouvrir un terminale série
- Employer ce mode pour vérifier ou changer la configuration de Bluetooth : le mot de passe PIN, la vitesse baud, le mode de maître / esclave

La configuration d'un Bluetooth module de HC-06, peut être réalisé par l'intermédiaire d'un terminale série aux commandes. (On peut utiliser IDE Arduino moniteur de série, sous menu "Tool / Serial Monitor", ou juste, cliquer sur CTRL+M du clavier du PC).

Dans le tableau qui suit, nous allons citer les plus importantes commandes que nous avons utilisées pour configurer le module de Bluetooth HC-06.

Parameters par default: Baud rate: 9600N81, ID: linvor, Password: 1234

Commande	Fonction
AT	Ne fait rien mais la réponse "OK" signifie bon essai pour connaître le mode de CMD.
AT+VERSION	donner la version Bluetooth HC-06.
AT+NAMEMyName	Changer le nom avec 'MyName'
AT+PINxxxx	Change Le mot de passe du Bluetooth à 4 chiffres xxxx
AT+BAUD1 AT+BAUD2	Placer la vitesse de baud, 1-----1200 2-----2400 3-----4800 4-----9600 (Default) 5-----19200 6-----38400 7-----57600 8-----115200 9-----230400 A-----460800 B-----921600 C-----1382400 PC can't support the baud rate lager than 115200.

Tableau IV.4 : Commande BT

IV.3.2 Simulation de notre carte électrique

On passe maintenant à la simulation de notre carte électrique et la simuler virtuellement comme le montre la figure suivante :

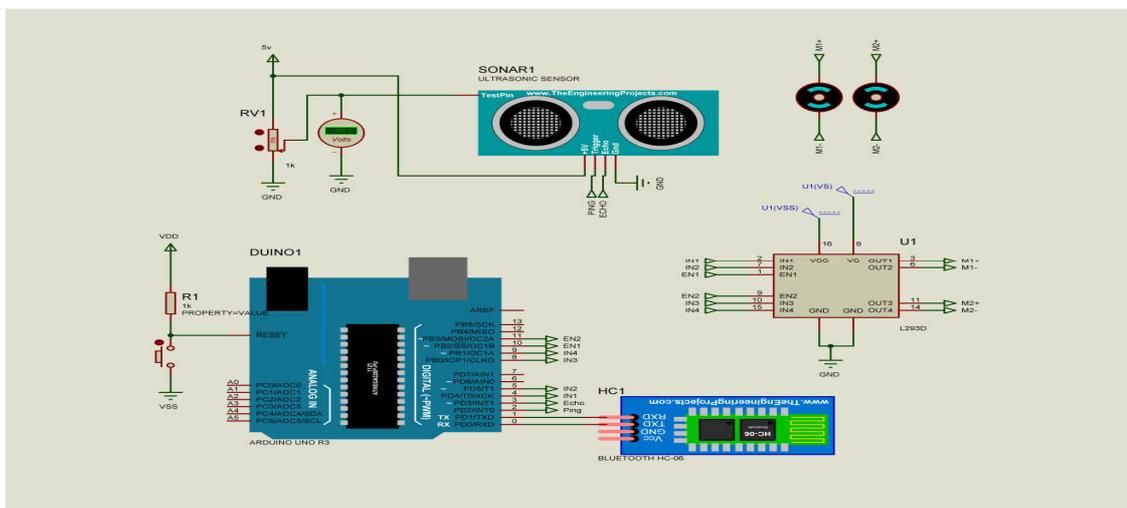


Figure IV.11: Carte réalisée sous ISIS-PORTUEUS

Ce CAO a la possibilité d'emporter même des codes hexadécimaux pour les réalisations qui contiennent des composants programmables ou des cartes programmables « Arduino » comme dans notre réalisation.

Le code hexadécimal de notre programme est représenté dans l'annexe A

La simulation de notre dispositif sous ISIS prouve la réussite, ce CAO possède même un virtuel USB qu'on peut utiliser comme un Bluetooth (un terminal USB).

IV.3.3 Réalisation pratique

IV.3.3.1 Composants utilisés

Pour une telle réalisation, nous avons assemblé les différents composants:

- Une Carte Arduino UNO
- Un Bluetooth module de HC-06
- un capteur ultrasons HC-SR04
- 2 LEDs
- 4 résistances (470 ohms, 1Kohms, 2.2 KOhms)
- 2 pile 9V
- 2 moteur a courant continue
- Fiche jack.
- plaque d'époxy de x* y cm
- 3 roues

IV.3.3.2 Robot mobile assemblé

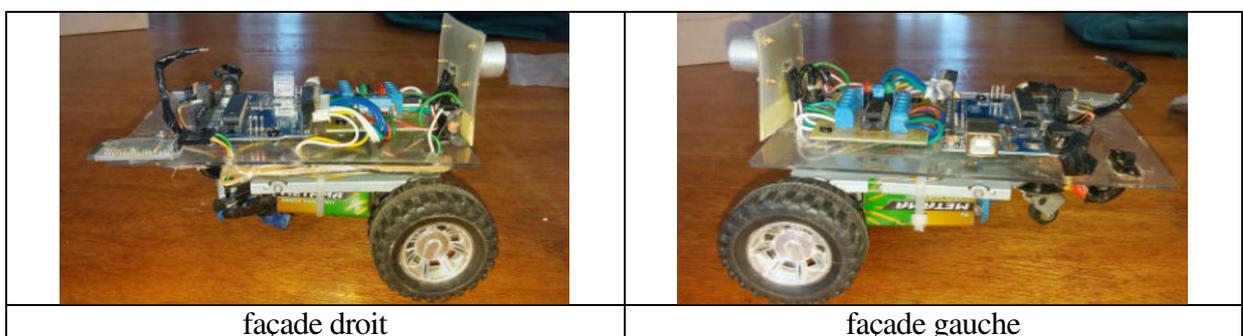


Figure IV.12 : Robot mobile assemblé

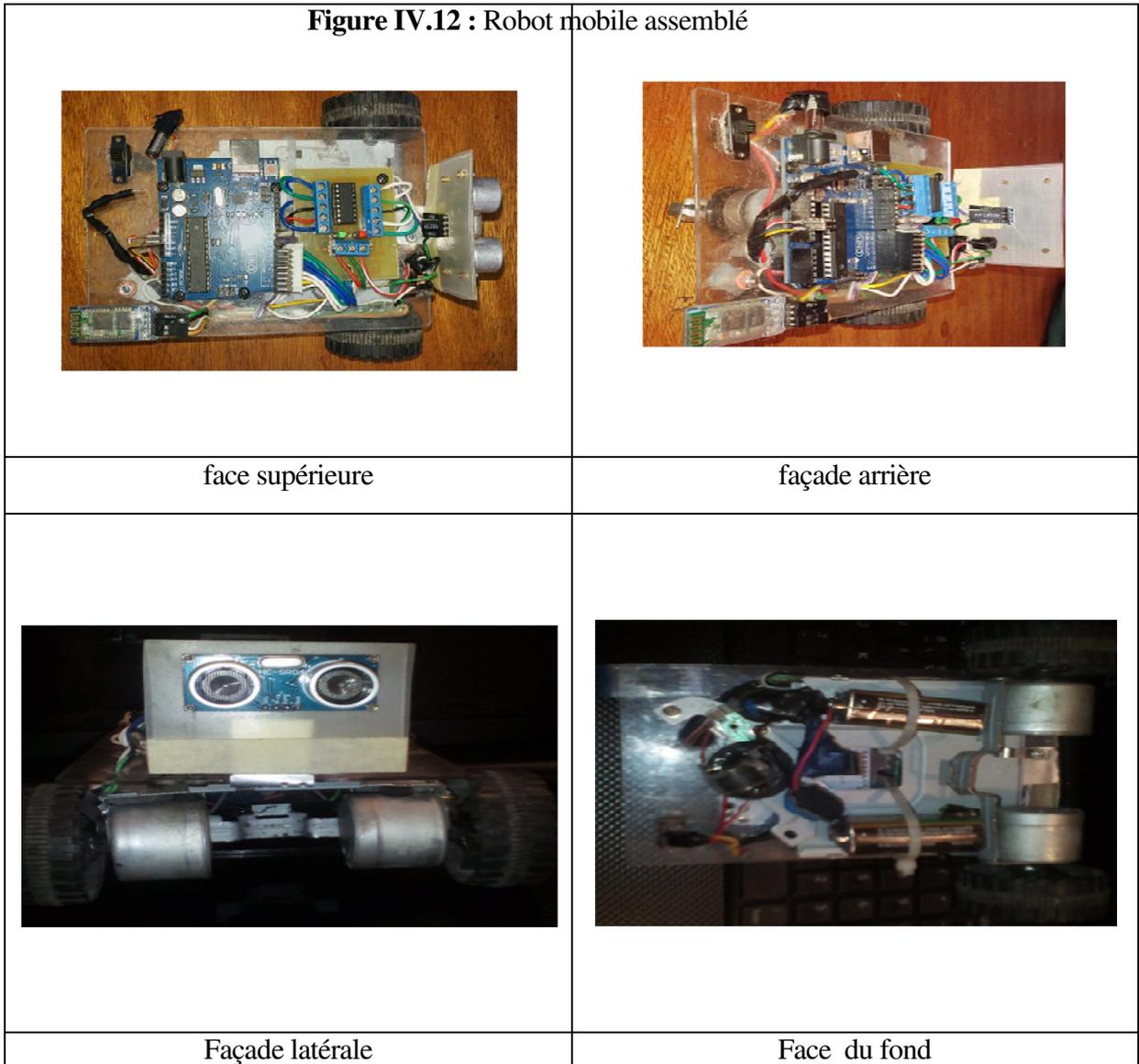


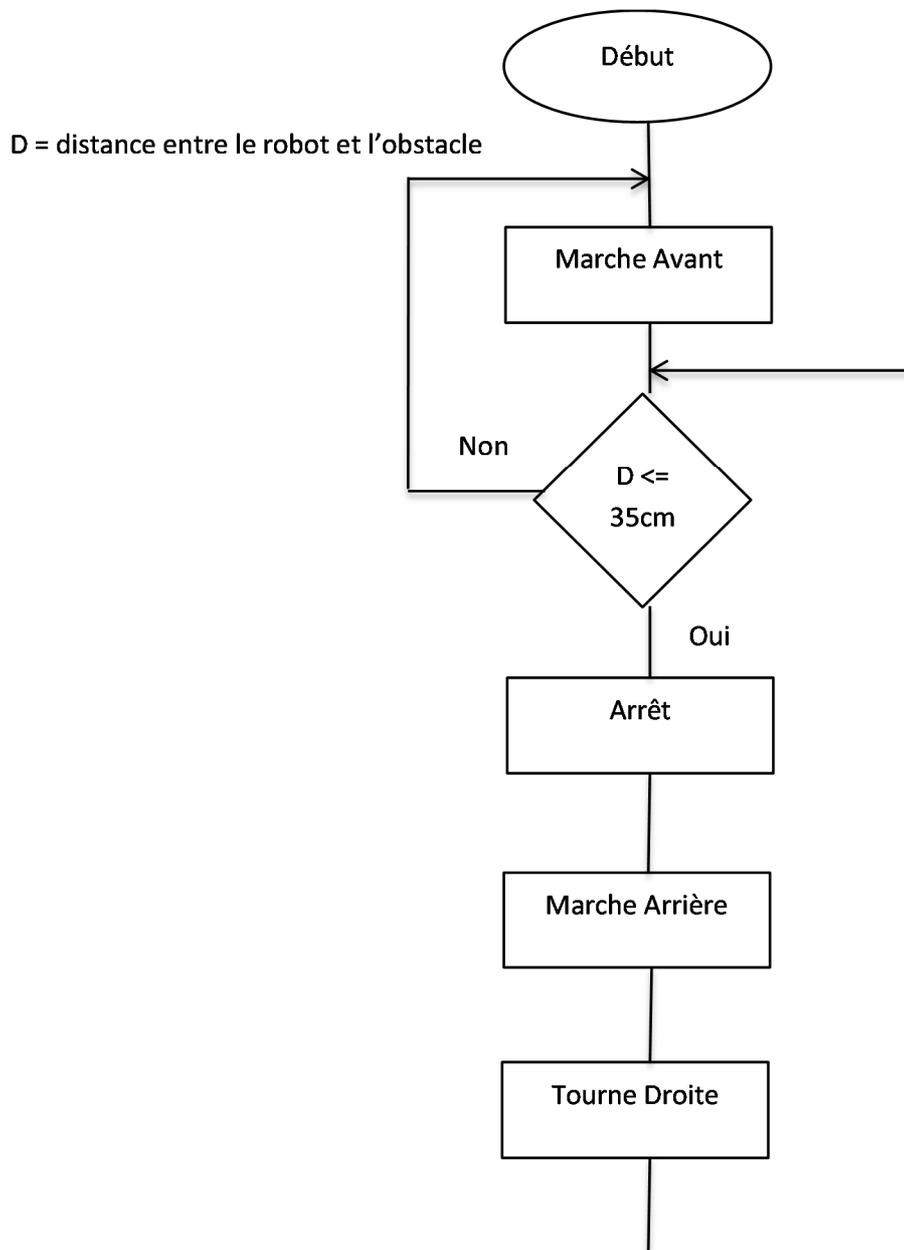
Figure IV.12 : Robot mobile assemblé

IV.3.4 Présentation du programme IDE

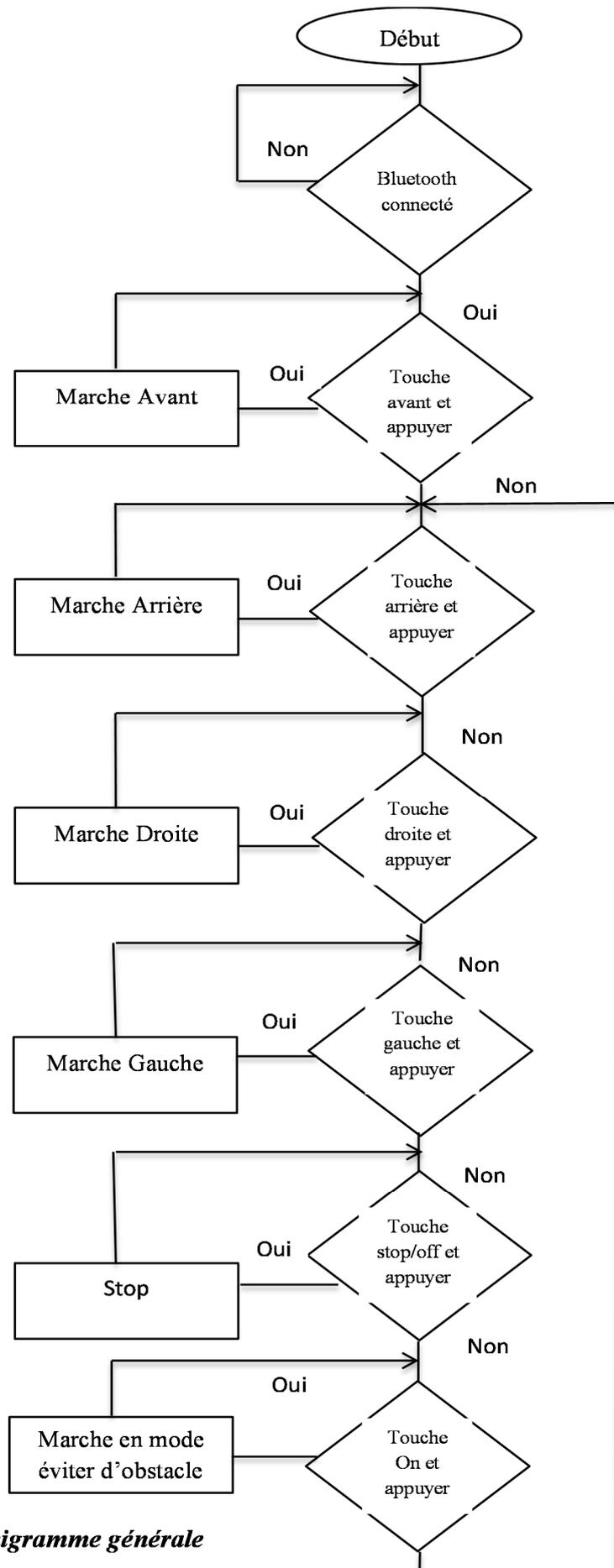
Nous préférons de photographier l'interface du IDE de l'environnement Arduino dans lequel nous avons simulé notre programme.

Le code hexadécimal de notre programme est représenté dans l'annexe A

Voilà notre organigramme ce qui simplifie la programme arduino



Organigramme de l'éviteur d'obstacle



Organigramme générale

```

bleutooth | Arduino 1.8.1
Fichier Édition Croquis Outils Aide
bleutooth

int s;
int spd;
int x;
int IN1 = 2;
int IN2 = 3;
int EN1 = 5;
int IN3 = 4;
int IN4 = 7;
int EN2 = 6;
int echo = 9;
int ping = 8;

void setup() {
  // put your setup code here, to run once:F

  pinMode (2, OUTPUT);
  pinMode (3, OUTPUT);
  pinMode (4, OUTPUT);
  pinMode (5, OUTPUT);
  pinMode (6, OUTPUT);
  pinMode (7, OUTPUT);
  pinMode (8, OUTPUT);

  Serial.begin(9600);
}

void loop() {
  //while ( Serial.available() ){

  if (Serial.available() > 0)
  {
    x = Serial.read();
  }

  switch (x) {

    case 'f' :
    {
      forward();
      delay(250);
      //arret();
    }
    break ;

    case 'b' :
    {
      backward();
      delay(250);
      //arret();
    }
    break;

    case 's' :
    {
      arret();
      delay(250);
    }

    break ;

    case 'l' :
    {
      left();
      delay(250);
      //arret();
    }
    break ;

    case 'r' :
    {
      right();
      delay(250);
      //arret();
    }
    break ;

    case 'e' :
    {
      long duration, cm;
      pinMode(ping, OUTPUT);
      digitalWrite(ping, LOW);
      delayMicroseconds(4);
      digitalWrite(ping, HIGH);
      delayMicroseconds(20);
      digitalWrite(ping, LOW);
      pinMode(echo, INPUT);
      duration = pulseIn(echo, HIGH);

      cm = microsecondsToCentimeters(duration);
      delay(100);

      if (cm <= 35)
      {
        arret();
        delay(500);
        backward();
        delay(1000);
        right();
        delay(1000);
      }

      else
      {
        forward();
      }
    }
    break ;

  default:
  //x = x + int (Serial.read());
  delay(50);
  spd = map(x, 0, 100, 2, 255);
  delay(50);
  Serial.println(spd);

  analogWrite (EN1, spd);
  analogWrite (EN2, spd);
  break ;
}
}

```

```
void backward()
{
  Serial.println("Back");
  digitalWrite(IN2, LOW);
  digitalWrite(IN1, HIGH);

  digitalWrite(IN4, LOW);
  digitalWrite(IN3, HIGH);
}
void arret()
{
  Serial.println("stop");
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);

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

  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
void right()
{
  Serial.println("right");
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);

  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
}
void forward()
{
  Serial.println("forward");
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);

  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}
long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}
```

Figure IV.13 : programme Arduino

IV.4 Partie informatique

Nous proposons dans cette partie l'élaboration d'un outil d'aide à l'utilisateur « M2II » permettant de commander un robot mobile. Nous débutons par l'expression des besoins fonctionnels exprimées par l'administrateur sous forme d'un cahier des charges, puis nous procédons à l'analyse de ce dernier dans le but de concevoir un outil englobant les exigences formalisées. Nous terminons par une description de la partie réalisation qui consiste à mettre en oeuvre l'application en utilisant des outils de développement.

IV.4.1 Cahier des charges et analyse des besoins fonctionnels

La cohérence et la complétude des besoins recueillis auprès des utilisateurs (User), permettent la rédaction d'un cahier des charges qui servira d'élément fondamental à la validation du système développé, et de s'assurer que ce dernier répond aux attentes et exigences de tous les utilisateurs (User). Avant de d'entamer l'étape d'analyse, nous présentons le cahier des charges de l'outil « M2II » qui englobe l'expression des besoins fonctionnels.

IV.4.1.1 Description des choix techniques

La réalisation d'un système quelconque passe obligatoirement par une étude approfondie concernant ce que doit retourner ce dernier. Durant cette phase d'étude, plusieurs outils et langages de modélisation sont exploités afin de permettre et de faciliter une conception et une modélisation de qualité. Dans le cadre de notre travail, notre choix à porté sur l'utilisation du langage de modélisation UML Basé sur des représentations graphiques (Diagrammes), il permet d'exprimer visuellement des solutions objets, ce qui facilite la comparaison et l'évaluation de ces dernières; et d'offrir également plusieurs niveaux d'abstraction, qui permettent de mieux contrôler la complexité dans l'expression des dénouements.

a.C'est quoi UML ?

UML, c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un **langage visuel** constitué d'un ensemble de schémas, appelés des **diagrammes**, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour **représenter** le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. [42]

IV.4.1.2 Cahier des charges

Nous nous intéressons dans cette section à l'identification des acteurs pouvant interagir avec l'outil, ainsi que le recueil des besoins fonctionnels exprimés par l'administrateur.

a.Identification des Acteurs

Le seul acteur de notre application est l'utilisateur qui commander le robot mobile.

IV.4.1.3 Analyse des besoins fonctionnels

L'analyse des besoins fonctionnels est une méthode permettant, à partir du cahier des charges défini précédemment, de décrire l'aspect fonctionnel de l'outil développé. Cette description sera réalisée à l'aide des diagrammes :

- Diagramme de Cas d'utilisation (identification des acteurs et fonctionnalités).
- Diagramme de Séquences (interactions entre les acteurs et les fonctionnalités de l'outil).

a. C'est quoi diagramme de cas d'utilisation ?

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. [43]

b. C'est quoi diagramme de séquences ?

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Ainsi, contrairement au diagramme de communication, le temps y est représenté explicitement par une dimension (la dimension verticale) et s'écoule de haut en bas [44]

c. Identification des cas d'utilisation de l'outil

Le tableau ci-dessous permet d'énumérer les principaux cas d'utilisation et de les décrire :

Use case	Description
Authentification	Assure la connexion de l'utilisateur à interface menu en précisant deux paramètres : nom d'utilisateur et mot de passe.
Commande manuel	Une fois la connexion établie, l'utilisateur peut commander le robot d'une manière manuel
Commande vocal	Une fois la connexion établie, l'utilisateur peut commander le robot d'une manière vocal

Tableau IV.5 : Description des cas d'utilisations

d. Description du diagramme de cas d'utilisation

A travers la figure suivante, nous représentons le diagramme de cas d'utilisation montrant les fonctionnalités globales de notre outil. Nous pouvons remarquer que l'utilisateur peut procéder soit à la commande manuel, soit à son vocal, et ceci après une authentification permettant la vérification du nom d'utilisateur et du mot de passe.

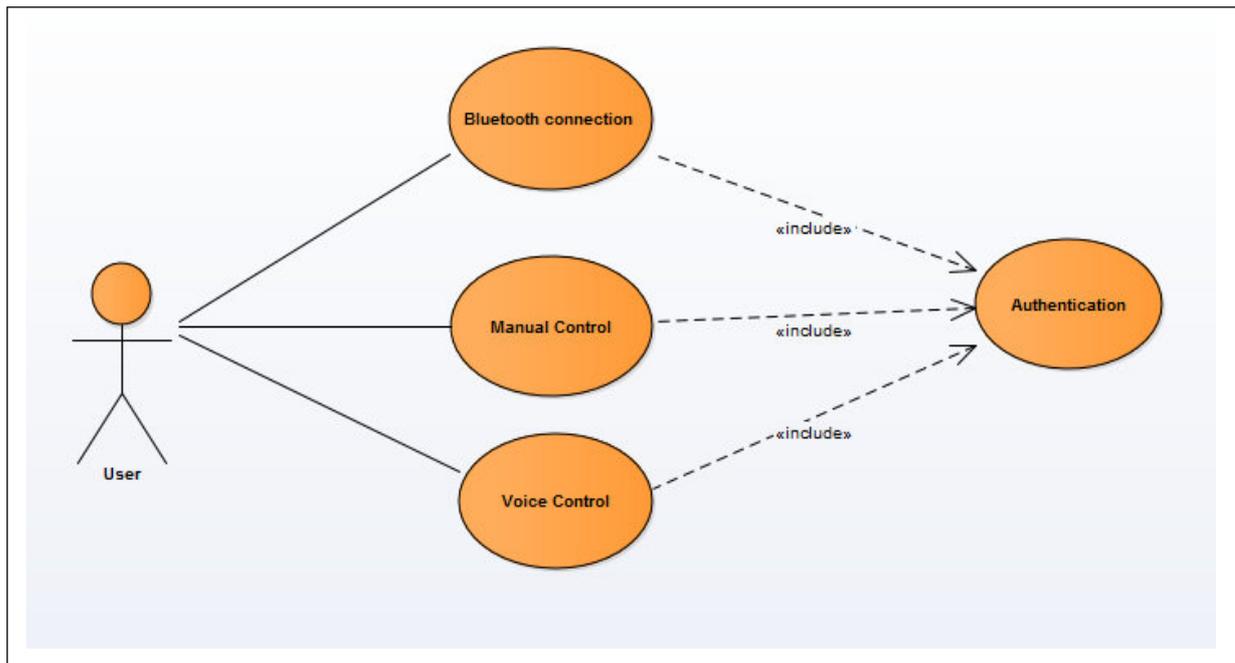


Figure IV.14 : Diagramme de cas d'utilisation

e. Diagramme de Séquence (Description dynamique des cas d'utilisation)

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Nous montrons à travers ces diagrammes les principales interactions dans le cadre d'un scénario du diagramme de cas d'utilisation défini précédemment. Le but étant de décrire comment se déroulent les actions entre les acteurs et objets.

1. Scénario nominal «Authentification »

Afin d'ouvrir l'application interface d'accueil est sera charger, utilisateur doit s'authentifier au départ en insérant ses paramètres de connexion (User Name et Password). Le système vérifie ces paramètres, puis charge l'interface Menu. Ce scénario est illustré par la figure IV.15

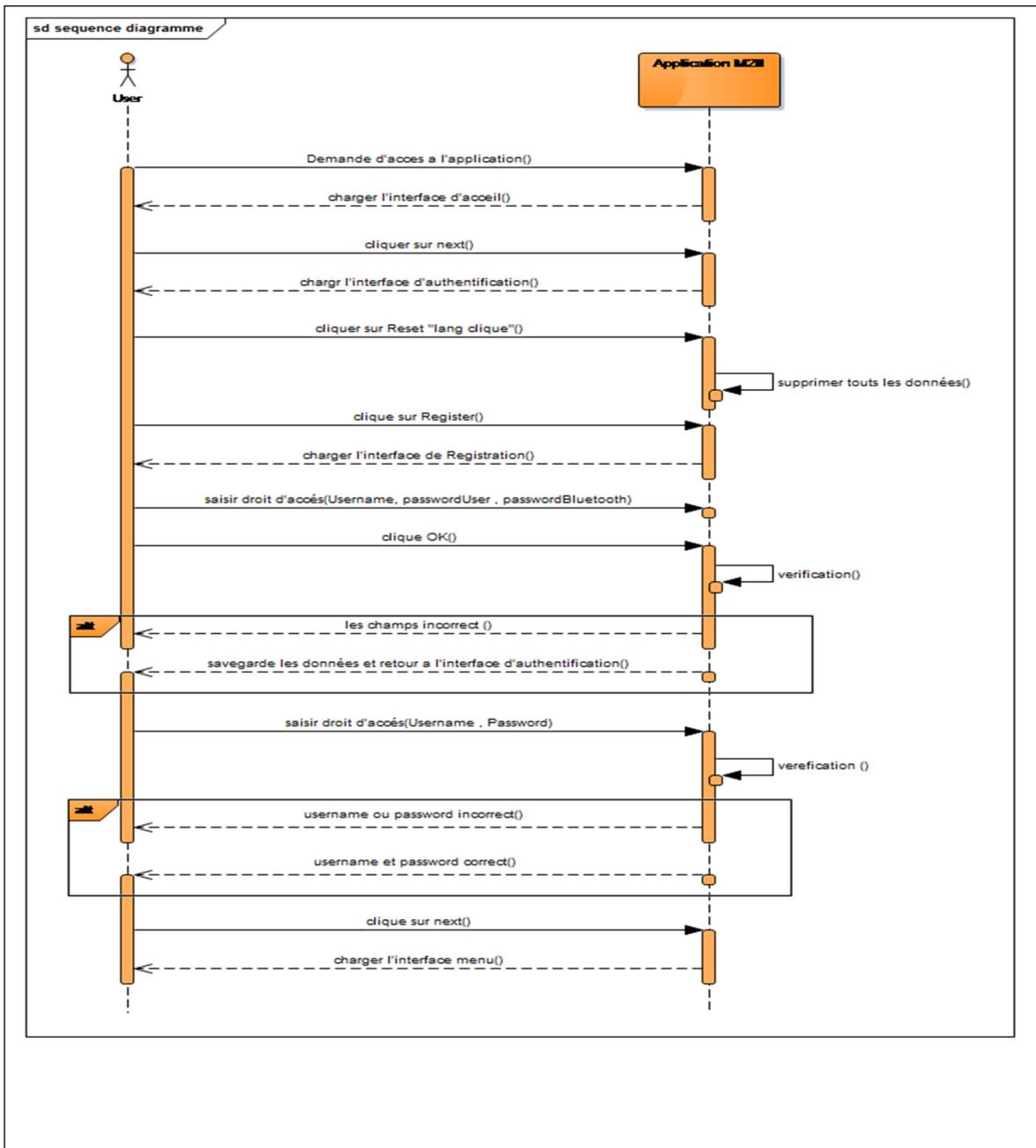


Figure IV.15 : Digramme de séquence «Authentification »

2. Scenario nominal2« Manuel control »

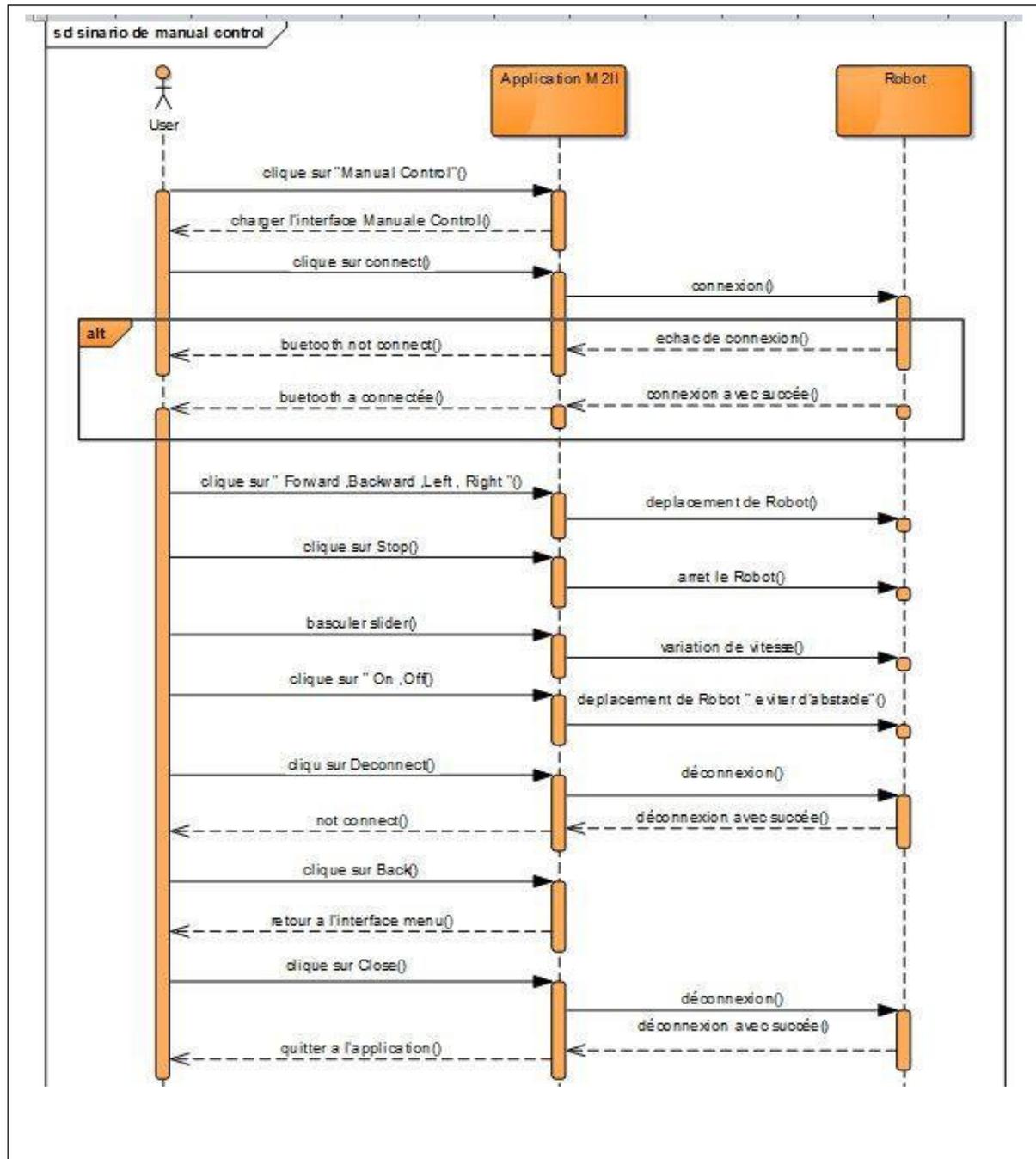


Figure IV.16 : Digramme de cas séquence« Manuel control »

IV.4.2 Implémentation de l’outil M2II

Après avoir exprimé les différents besoins fonctionnels et avoir présenté la succession des étapes de modélisation de l’outil que nous avons proposé, afin de procéder à la réalisation de cette application qui présente un support d’aide à l’utilisateur à commander le robot mobile à distance en communiquant par Bluetooth.

Nous abordons dans cette partie la mise en œuvre de notre outil « M2II».

IV.4.2.1 Présentation de l'application

Nous présentons dans cette section les principales fonctionnalités de l'outil « M2II» à travers des captures écran illustrant les interfaces de ce dernier. Pour bien décrire ces interfaces, nous considérons un scénario d'exécution et de manipulation de l'outil, à partir de l'authentification puis la connexion Bluetooth, en passant par les deux modes de commande « «manuel & vocal » » .

a.Interface « d'accueil M2II »

Lors du lancement de l'application, la fenêtre d'accueil est chargée (Figure IV.17).

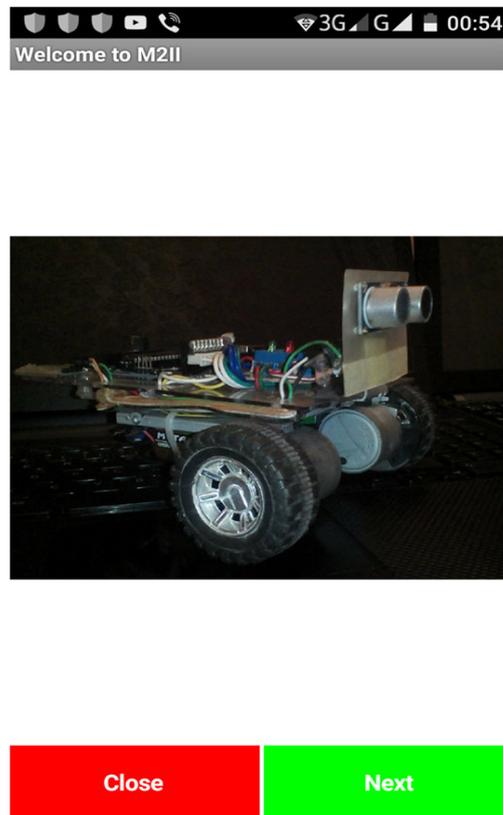


Figure IV.17 : Fenêtre d'accueil M2II

Dans cette application il y'a vingt boutons, trois parmi elles qui nous trouvons dans la plupart des interfaces qui sont:

N°	Titre	Type	fonctionnalité
1	Close	Bouton	Quitté application
2	Next	Bouton	Passer à l'interface suivant
3	Back	Bouton	Retour à l'interface précédent
4	Login	Bouton	Vérifier ID
5	Reset	Bouton	Supprimer tous les utilisateurs
6	Register	Bouton	Passer à l'interface de registration
7	Ok	Bouton	Valider les données et vérifier login
8	Manuel control	Bouton	Passer à l'interface de la commande manuel
9	Voice control	Bouton	Passer à l'interface de la commande vocal
10	Connect	Lispicker	Connecter au Bluetooth
11	Deconnect	Bouton	Déconnecter au Bluetooth
12	On	Bouton	Démarrer le robot en mode évité d'obstacle
13	Off	Bouton	Stop
14	Forward	Bouton	Déplacer le robot en avant
15	Backward	Bouton	Déplacer le robot en arrière
16	Right	Bouton	Déplacer le robot à droite
17	Left	Bouton	Déplacer le robot à gauche
18	Stop	Bouton	Stop
19	Micro	Bouton	Control vocal
20	Slider	slider	Varié la vitesse de 0 à 100%

Tableau IV.6 : Boutons d'application

b.Interface « d'authentification »

L'utilisateur saisit leur nom d'utilisateur et leur mot de passe et appuyez sur la touche **LOGIN**, Ce qui confirme l'authentification.

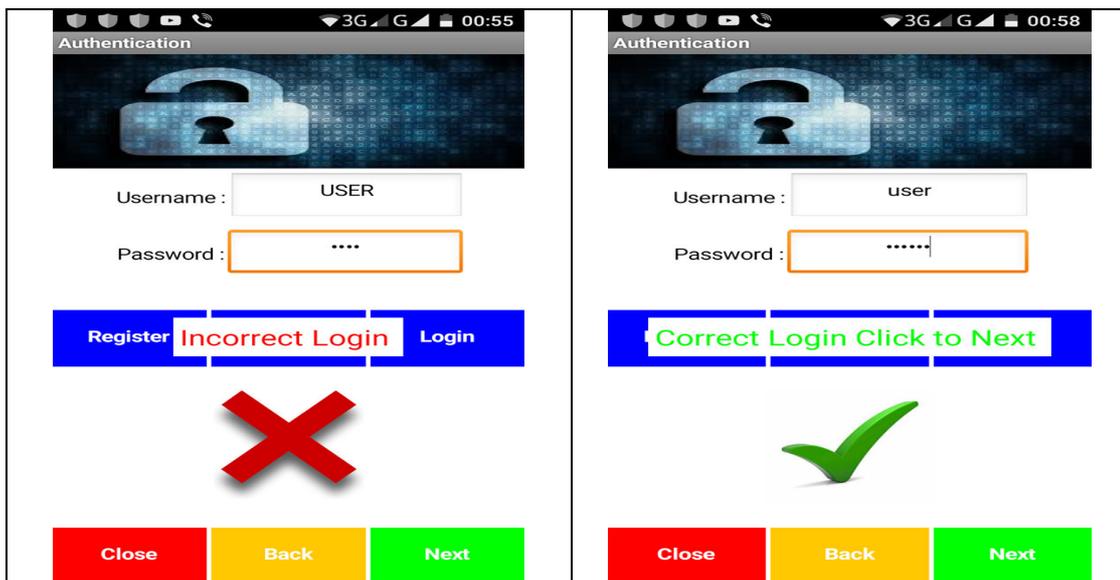


Figure IV.18 : Fenêtre d'authentification

c. Interface « registration »

En tant que premier utilisateur utilisant l'application, vous devez vous inscrire comme suit

1. Entrer le nom d'utilisateur
2. Confirmer le nom d'utilisateur
3. Entrer le mot de passe
4. Confirmer le mot de passe
5. Entrer le mot de passe du module Bluetooth
6. Cliquer sur ok

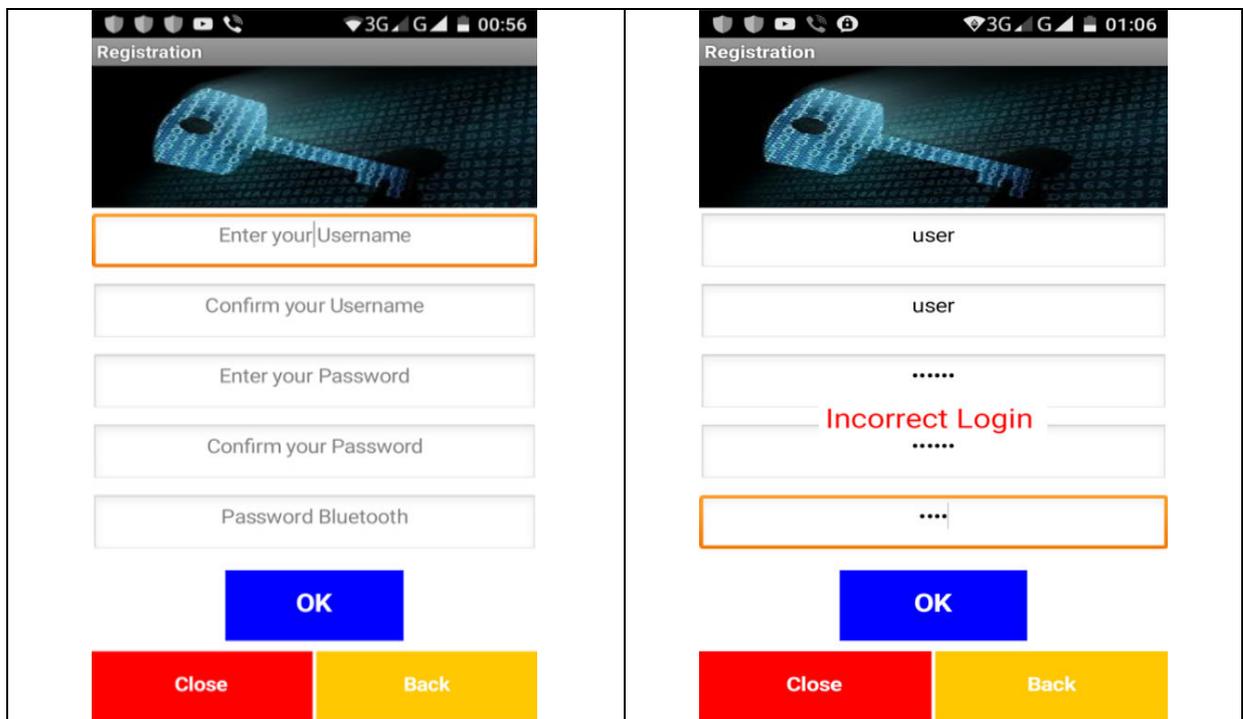


Figure IV.19 : Fenêtre de registration

d. Interface « Menu »

Notre application permet de contrôler notre robot en deux mode « manuel – vocal » comme monter dans la figure suivante :



Figure IV.20 : Fenêtre de menu

e.Interface « Manual Control »

Notre interface contient une touche d'activation et désactivation du Bluetooth donc avant d'envoyer n'importe quel caractère à travers les boutons, il faut d'abord s'assurer que le Bluetooth est activé, pour avoir contrôlé le robot

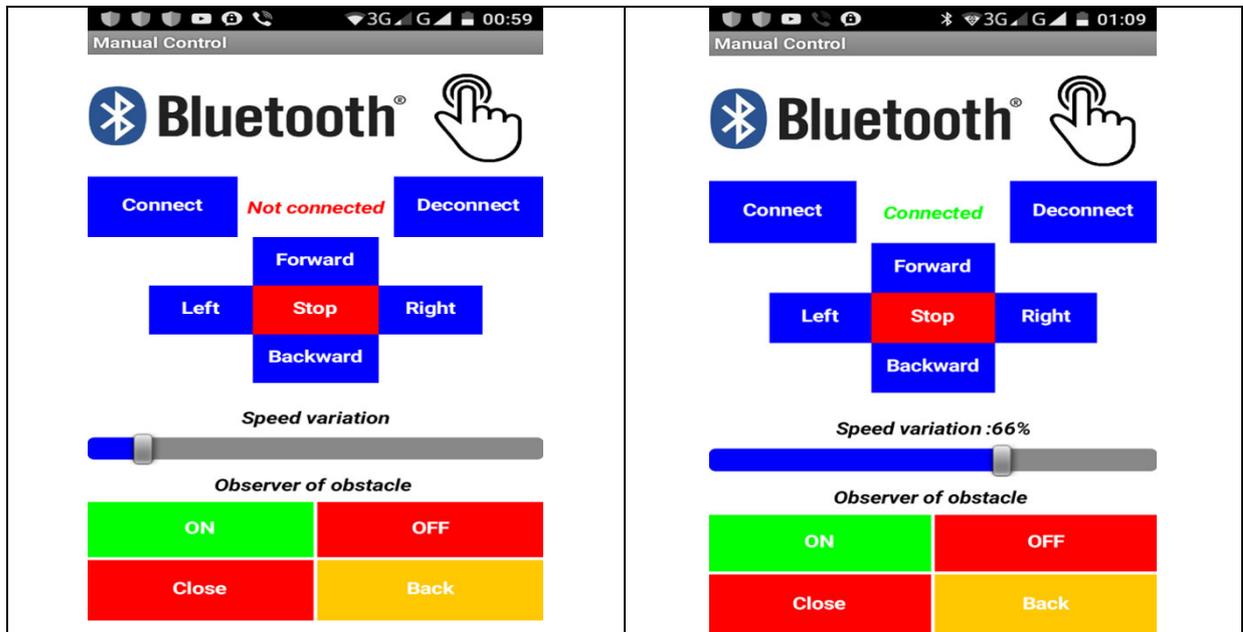


Figure IV.21 : Fenêtre de la commande manuelle

f. Interface « voice Control »

Vous devez connecter au bluetooth pour commander le robot vocalement en utilisant les mots suivants « avant – arrière – droite – gauche – stop – on – off - close pour quitter application ou back pour quitter l'interface en clique sur Botton micro ou bien passe la main devant écran de appareil

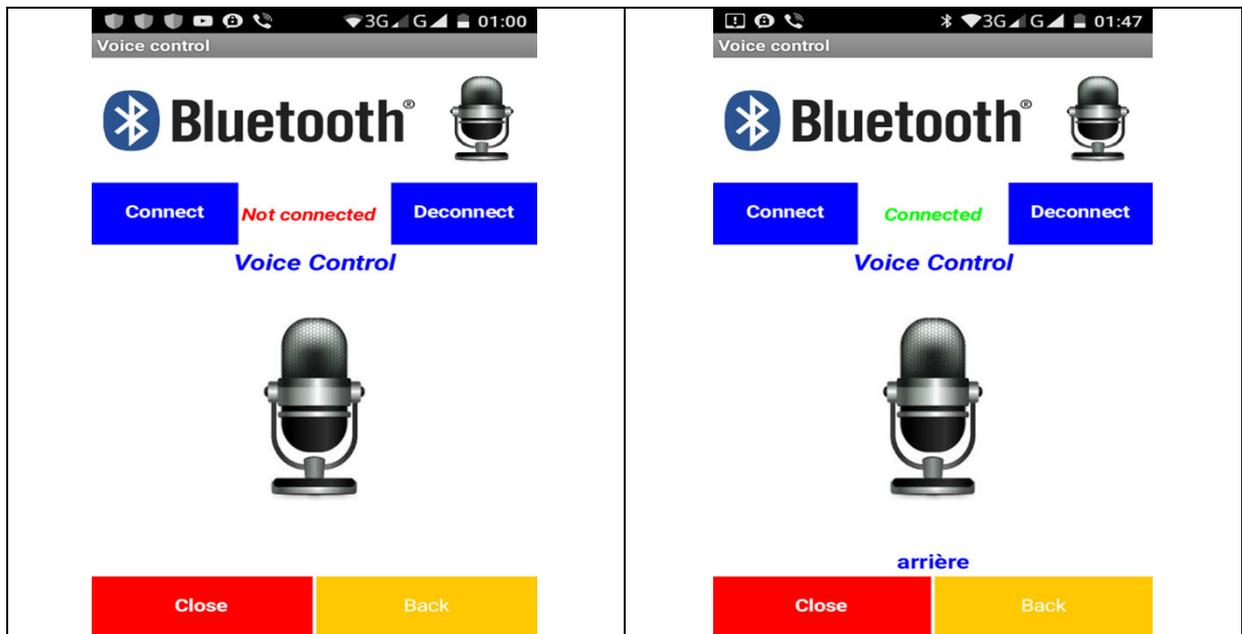


Figure IV.22 : Fenêtre de la commande vocale

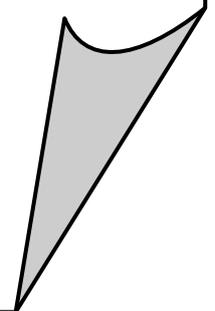
Remarque : Annexe B contient le code block correspondant à notre application

IV.4.3 Conclusion

Dans ce dernier chapitre, nous avons expliqué les différentes étapes qui nous ont permis de réaliser ce robot, ces composants utilisés et leurs tâches, on a présenté aussi une description générale du programme arduino implémenté sur la carte électronique, aussi l'application que nous avons développée « «M2II » ce qui nous a permis de contrôler le robot à distance.

La réalisation de ce robot a été une bonne occasion pour nous, d'une part d'acquérir des nouvelles connaissances, et d'autre part, d'assimiler les différents outils acquis durant cette période en matière de développement « Arduino, MIT app inventor ... etc. ».

Conclusion
&
Perspectives



La robotique est un domaine pluridisciplinaire. Durant ce projet nous avons pu acquérir une certaine expérience pour la conception et la réalisation des différentes parties du robot mobile. Nous avons avant tout, appris à bâtir un projet en respectant un cahier des charges, et ce par la construction d'une stratégie idée par idée jusqu'à maturité.

La partie réalisation nous a été d'un profit inestimable, que ça soit .De même pour la partie électronique qui a nécessité la réalisation d'une carte de commande " à base d'arduino" et la partie informatique qui a nécessité elle-même la création d'une application androïde "à base de MIT app Inventor", dans lesquelles nous avons pu mettre en pratique nos connaissances théoriques acquises tout au long de notre cursus de formation.

Notre robot mobile doit être se déplacer vers les quatre sens et évitera les obstacles avec une vitesse variable, la carte arduino " block de commande " lui permet d'effectuer ses tâches.

La particularité de notre travail réside dans l'utilisation d'une application androïde " M2II" qui permet de contrôler le robot à distance.

Pour cela, nous avons utilisé deux types de capteurs et un driver L293D :

1. capteur ultrasons pour la détection des obstacles
2. capteur bluetooth "module HC06" pour la commande à distance
3. on a choisi spécialement le L293D qui permet de commander un moteur à courant continu en marche « Avant - Arrière», grâce à 2 broches numériques de la carte Arduino.

Nous avons donc pu réaliser l'objectif de notre projet qui est la réalisation d'un robot mobile commandé à distance par bluetooth et l'utilisation de logiciel arduino et leur carte électronique arduino « Uno » pour manipuler notre robot mobile qui a trois roues en fin la création de notre application.

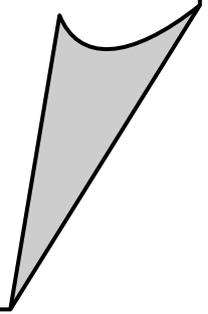
En perspective on souhaitera

- Utiliser d'autres langages de programmation comme Java, Micro C...etc
- Utilisation d'autre type de carte arduino ou bien des PICs.

Conclusion & Perspectives

- Enrichir notre projet par d'autres fonctionnalités comme la commande à partir d'une connexion Wifi
- Implémenter d'autres techniques de commande «télécommande à base de l'infrarouge wifi ...etc»

Sommaire



Sommaire

Résumé	
Remerciment	
Liste des tableaux	i
Liste des figures.....	ii
Liste d'abréviations	iv
Liste des nomenclateurs.....	v
Introduction générale	
Contexte.....	2
Problématique	2
Objectifs de notre travail.....	2
Organisation de mémoire.....	3
Chapitre I: Généralité sur la robotique	
I.1 INTRODUCTION	5
I.2 BREF HISTORIQUE SUR LA ROBOTIQUE	5
I.3 C'EST QUOI UN ROBOT ?.....	6
I.3.1 Étymologie.....	6
I.3.2 Définition.....	6
I.3.3 Trois lois de la robotique.....	7
I.4 TYPE DES ROBOTS	7
I.4.1 Robots manipulateurs.....	7
I.4.1.1 Types des robots manipulateurs	8
I.4.2 Robots mobiles	8
I.4.2.1 L'architecture des robots mobiles	8
I.4.2.2 Classification selon le type de locomotion	9
I.4.3 Domaine d'application des robots mobiles.....	12
I.5 CONCLUSION	14
Chapitre II: Capteurs & Actionneurs	
II.1 INTRODUCTION	16
II.2 CAPTEURS	16
II.2.1 C'est quoi un capteur ?.....	17
II.2.2 Chaîne de mesure.....	17
II.2.3 Classification des capteurs	18
II.2.3.1 Capteur actif.....	19
II.2.3.2 Capteur passif	19
II.2.3.3 Capteur composite.....	20
II.2.3.4 Capteur intégré.....	20
II.2.4 Caractéristiques d'un capteur	20
II.2.5 Capteur utilisé dans la réalisation.....	21
II.2.5.1 Capteur ultrasonique.....	21
II.2.5.2 Capteur Bluetooth.....	23
II.3 ACTIONNEURS	24
II.3.1 C'est quoi un actionneur ?.....	24
II.3.2 Familles d'actionneurs	24
II.3.2.1 Actionneurs électriques	24

Sommaire

II.3.2.2 Actionneurs pneumatiques.....	25
II.3.2.3 Actionneurs hydrauliques.....	25
II.3.3 Capteur utilisé dans la réalisation.....	25
II.3.3.1 Moteur à courant continu	25
II.4 CONCLUSION.....	27

Chapitre III: Microcontrôleur & Arduino

III.1 INTRODUCTION.....	29
III.2 MICROCONTROLEUR	29
III.2.1 Bref historique sur les Microcontrôleurs	29
III.2.2 C'est quoi un microcontrôleur ?.....	30
III.2.2.1 Microprocesseur	30
III.2.2.2 Mémoire de données	30
III.2.2.3 Mémoire programmable	30
III.2.3 L'architecture des microcontrôleurs.....	31
III.2.3.1 RISC et CISC l'architecture des microcontrôleurs	32
III.2.4 Langage de programmation d'un microcontrôleur.....	33
III.2.4.1 C.....	33
III.2.4.2 C++	33
III.2.4.3 Java	33
III.2.4.4 L'assembleur.....	33
III.2.5 Domaine d'applications des microcontrôleurs	34
III.3 ARDUINO	34
III.3.1 Bref historique de l' Arduino.....	34
III.3.2 C'est quoi Arduino ?.....	34
III.3.2.1 Application	35
III.3.2.2 Types de cartes	35
III.3.3 Différentes cartes Arduino.....	35
III.3.4 Les raisons de choisir Arduino.....	37
III.3.4.1 prix	37
III.3.4.2 liberté	37
III.3.4.3 Compatibilité.....	37
III.3.4.4 Communauté	38
III.3.5 Les outils Arduino.....	38
III.3.5.1 Matériel.....	38
III.3.5.2 Logiciel.....	41
III.3.6 Exemples d'usage	42
III.4 CONCLUSION	42

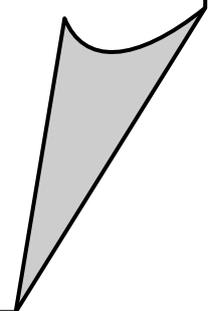
Chapitre IV: Conception & Réalisation de notre outil

IV.1 INTRODUCTION	44
IV.2 LES OUTILS UTILISES POUR LE DEVELOPPEMENT	44
IV.2.1 Proteus Professionnel	44
IV.2.1.1 Composants principaux de Proteus.....	44
IV.2.2 Arduino.....	46
IV.2.3 MIT app Inventor.....	46
IV.2.3.1 Commencement avec App Inventor	47

Sommaire

IV.2.4 Enterprise Architect	49
IV.3 PARTIE ELECTRONIQUE.....	49
IV.3.1 Schéma bloc	49
IV.3.1.1 Bloc de traitement et de contrôle.....	50
IV.3.1.2 Bloc de commande	50
IV.3.1.3 Bloc de puissance.....	50
IV.3.1.4 Bloc d'alimentation.....	53
IV.3.1.5 Capteur bluetooth.....	54
IV.3.2 Simulation de notre carte électrique	55
IV.3.3 Réalisation pratique.....	56
IV.3.3.1 Composants utilisés	56
IV.3.3.2 Robot mobile assemblé.....	56
IV.3.4 Présentation du programme IDE.....	57
IV.4 PARTIE INFORMATIQUE.....	61
IV.4.1 Cahier des charges et analyse des besoins fonctionnels	61
IV.4.1.1 Description des choix techniques.....	62
IV.4.1.2 Cahier des charges	62
IV.4.1.3 Analyse des besoins fonctionnels	62
IV.4.2 Implémentation de l'outil M2II	66
IV.4.2.1 Présentation de l'application	67
IV.4.3 Conclusion	71
Conclusion & perspectives	

Biographie
&
Web graphie



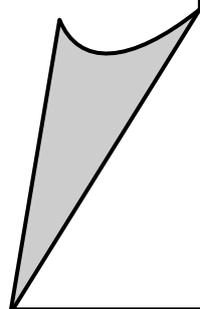
Biographie

1. Introduction à la robotique , auteur : Laetitia Matignon Université de Caen, France [2]
 2. Robotics and control, Auteur: Mittal & Nagrath, ISBN 0-07-048293-4[3]
 3. La double contrainte: L'influence des paradoxes de Bateson en Sciences humaines; Auteur : Jean-Jacques Wittezaele[4]
 4. Mécatroniques, Auteur: V.S.Bagad, ISBN9788184314908, Edition 2008[5]
 5. Robotique générale, Ateur : ALAIN PRUSKI, édition marketing,Paris-1988 ISBN 2-7298-8838-1[6]
 6. Robotique Mobile, Ateur : B. BAYLE,Ecole Nationale supérieur de Strasbourg, France, 2008-2009[7]
 7. Robotics Auteur : Appuu Kuttan, International Publishing House Pvt. Ltd. (2007)[10][11]
 8. Abrégé de mesure physique, Auteur : Aurélien PIERRE ; 2éme Edition ISBN979-10-92277-01-2 [13]
 9. Les capteurs en instrumentation industrielle ; Auteur : Georges Asch,7éme edition[14]
 10. Informatique et automation: automatismes programmables contrôlés par ordinateur ,Auteur : Jean Mbihi; ISBN2553007981[17]
 11. Théorie élémentaire et pratique de la commande par les régimes glissants, Auteur : Pierre Lopez,Ahmed Saïd Nouri,ISBN10 3-540-31003[18]
 12. Advanced PIC Microcontroller Projects in C: From USB to RTOS with the PIC ... Auteur: Dogan Ibrahim[20]
 13. Microcontrollers: Theory and Applications, Auteur: Ajay V Deshmukh, ISBN0-07-058595-4[21]
 14. Microcontrollers: Features and Applications, Auteur: D. S. Yadav,A. K. Singh[22]
 15. Microcontroller and Embedded System , Auteur: A.K. Singh[23]
 16. Microcontrollers: Architecture, Programming, Interfacing and System Design, Auteur: Raj Kamal.[24]
 17. Le Chemin De Réussite En Programmation Structurée Algorithmiques & Langage C ,Auteur : Samir ELACHOURI[25]
 18. Le langage C , Auteur : Peter Aitken, Bradley Jones[26]
 19. Le langage C++ , Auteur : Marius Vasiliu[27]
 20. Java Programming: Comprehensive Concepts and Techniques , Auteur: Gary Shelly,Thomas J. Cashman,Joy Starks,Michael Mick[28]
 21. Arduino applications avancées ,Ateur :C. Tavernier, Version Dunod[36]
-

Webographie

1. <https://www.gotronic.fr/ins-histoire-de-la-robotique-49.htm> [1]
2. http://gte.univ-littoral.fr/sections/documents-pdagogiques/chapitre-8-mesure/downloadFile/file/Les_capteur[12]
3. <https://knowledge.parcours-performance.com/arduino-bluetooth-hc-05-hc-06/>[16]
4. http://lpmei.com/cd_bac_mei/eleve/cours/Automatisme/223%20Les%20actionneurs.pdf[19]
5. http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php[29]
6. <https://framablog.org/2011/12/10/arduino-histoire/>[30]
7. <https://www.arduino.cc>[31]
8. Arduino pour bien commencer en électronique et en programmation - Astalaseven ,Eskimon et olyte - cite du Zéro <http://www.siteduzero.com>[32]
9. <http://fr.rsonline.com/web/generalDisplay.html?id=infozone&file=electroniques/comparing-arduino-boards>[33]
10. <http://www.01net.com/telecharger/windows/Programmation/creation/fiches/131502.html>[34]
11. http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php[35]
12. <http://www.generationrobots.com/fr/152-arduino>[37]
13. <https://www.labcenter.com>[38]
14. http://sig.fgranotier.info/IMG/pdf/debuter_app_inventor[39]
15. <http://ai2.appinventor.mit.edu>[40]
16. <http://www.01net.com/outils/telecharger/windows/Programmation/langage/fiches/tele9345.html>[41]
17. <https://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/uml-c-est-quoi>[342]
18. <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>[43]
19. <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/sequence-diagram>[44]

Annexe



```
:10000000C94FC000C9424010C9424010C94240105
:100010000C9424010C9424010C9424010C942401CC
:100020000C9424010C9424010C9477050C94E9059C
:100030000C9424010C9424010C9424010C942401AC
:100040000C9424010C9424010C9424010C9424019C
:100050000C9424010C9424010C9424010C94260883
:100060000C9424010C9424010C9424010C9424017C
:100070000C9424010C9424010C9424010C9424016C
:100080000C9424010C9424010C9424010C9424015C
:100090000C9424010C9424010C9424010C9424014C
:1000A0000C9424010C9424010C94240108030B03E8
:1000B0000FA02FE02032C032C032C030F0313038A
:1000C00017031D0321032C032703A303A803AD0378
:1000D000B703C1030B040B040B04CB03D503DF03ED
:1000E000E903F6030B04FE03000000024002700D0
:1000F0002A00230000000000000250028002B0001
:1001000000C9424010C9424010C9424010C9424016E
:10011000000C000000000000000000000000000D3
:1001200000000080B00002020201000904000001A7
:1001300002020000524001001052401010104242D
:10014000020605240600010705810310004009048A
:100150000100020A00000007050202400000070536
:10016000830240000004030904556E6B6E6F776EC6
:100170000041726475696E6F204C656F6E617264C8
:100180006F20455448001201000200000040032A7D
:10019000408000010102030112010002EF02014050
:1001A000032A40800001010203010000000230037
:1001B000260029002C002F00040404040403040575
:1001C0000202020204030202020202060606060606F4
:1001D00004040202020404040802011040804010DA
:1001E00020408040808020401804020100201105D
:1001F000801020404020C20A11241FBECFEFDAE059
:10020000DEBFCDBF21E0A8E4B1E001C01D92AC3B50
:10021000B207E1F711E0A0E0B1E0E6E7F6E102C0E5
:1002200005900D92A834B107D9F710E0CECFD0E0DB
:1002300004C02197FE010E94D70ACB3FD107C9F7E1B
:100240000E94B5080C94390B0C940000FC018091BD
:1002500006B01811103C08FEF9FEF08959FB789452
:1002600082E08093E9002091F20030E0121613063C
:1002700014F421E030E02115310559F0289884E686
:1002800080936A018091F10080838091F20088233D
:1002900019F09FBFC90108958BE68093E800F9CF5C
:1002A0008091620181110DC082E080935E0184E043
:1002B00080935F01109261011092600181E0809350
:1002C00062018E591E008954091540150915501ED
:1002D000209152013091530142175307B4F4909189
:1002E000E8009570E1F39091E80092FD19C080923D7
:1002F000F100809154019091550101968F739927D9
:10030000892B19F48EEF8093E8008091540190912D
:1003100055010196909355018093540181E0809511
:1003200080E00895CF92DF92EF92FF920F931F9398
:10033000CF93DF931F92CDB7DEB7182F062FE42E91
:10034000862F880F8E5F99830E94640183E00E944C
:100350006401F12EC12E9981D92E8C2D8F19801711
:1003600098F4F601E7FE02C0849101C080810E94EA
:100370006401182F80E00E9464018123FFEF9CF1AEF
:10038000DF0A8111EACF01C081E00F90DF91CF91A8
:100390001F910F91FF90EF90DF90CF900895DF9223
:1003A000EF92FF920F931F93CF93DF93D82E8A0182
:1003B0007B01E40EF51EEB01CE15DF0571F0D7FED3
:1003C00003C0FE01849101C088810E9464012196CE
:1003D0008111F2CF8FEF9FEF01C0C801DF91CF9164
:1003E0001F910F91FF90EF90DF9008950F931F93F4
:1003F000CF93DF931F92CDB7DEB782E0898342E4CB
:1004000050E063E271E080E80E94CF010E94500159
:10041000DC0112960D911C9113970115110569F0DD
:10042000D801ED91FC9101280F381E02DBE016F5F58
:100430007F4FC801099597FF07C089810F90DF9111
:10044000CF911F910F910895F80100851185E5CF97
:10045000615030F02091F100FC0120830196F8CF2F4
:10046000289884E680936A010895CF92DF92FE92B
:10047000FF920F931F93CF93DF936C017A018B014F
:10048000C0E0D0E0CE15DF0589F0D8016D918D0177
:10049000D601ED91FC910190F081E02DC601099506
:1004A000892B11F47E0102C02196ECCFC701D9091A8
:1004B000CF911F910F91FF90EF90DF90CF90089513
:1004C000CF93DF931F92CDB7DEB76983DC01ED9147
:1004D000FC910280F381E02D41E050E0BE016F5FAE
:1004E0007F4F09950F90DF91CF91089583E080931E
:1004F000E9008091F20082319F08AE38093E800F4
:100500000895CF93DF931F92CDB7DEB7FC018485AA
:10051000958597FD05C02FEF3FEF358724870BC0EA
:10052000CE0101960E942601019719F4898190E07D
:1005300002C08FEF9FEF0F90DF91CF9108950F93D1
:100540001F93CF93DF931F92CDB7DEB78C01FC01D3F
:100550008485958597FF0E0C0E0101960E942601E5
:10056000019719F4298130E002C02FEF3FEF380125
:1005700035872487F8018485958597F90DF91CF9189
:100580001F910F910895FC018485958597D0BC0FF
:100590009FB7F89482E08093E9008091F2009FBFBFA
:1005A00090E0019608959FB7F89482E08093E90067
:1005B0008091F2009FBF90E00895362F90E0FC01FB
:1005C000EC5FE4F4491FC01E952FE4F2491FC0187
:1005D000E854FE4F8491882309F451C04423E1F18B
:1005E00050E0FA013197EF30F105B0F5EA5AFF4FC
:1005F0000CF94D70A909180009F7707C0909180005B
:100600009F7D03C090918000977F9093800024C0CD
:1006100094B59F7702C094B59F7D94BD1DC0909105
:1006200090009F7702C0909190009F7D03C09091A
:100630009000977F90939000FC09091C0009F779B
:1006400003C09091C0009F7D9093C00005C0909121
:10065000C200977F9093C200E82FF0E0EE0FFF1FDB
:10066000EA50FF4FA591B4918FB7F894EC913111F6
:1006700003C020952E2301C02E2B2C938FB0895ED
:10068000CF93DF9390E0FC01E952FE4F2491FC01EF
:10069000E854FE4F8491882309F451C04423E1F18B
:1006A000FC01E851FFF4FC5891D491FC01E50FF4F86
:1006B000A591B491611109C09BF7F89488812095E4
:1006C00082238883EC912E230CB0623061F49B7A4
:1006D000F8943881822F809583238883EC912E2B88
:1006E0002C939FBF06C08FB7F894E8812E2B2883E8
:1006F0008FBFDF91CF9108951F93CF93DF93182F72
:10070000EB0161E00E944003209711F460E004C017
:10071000CF3FD10539F461E0812FDF91CF911F9157
:100720000C94DD02E12FF0E0EC5FFE4FE4914E2FE0
:1007300050E0FA013197EF30F10508F06C04E0B5949
:10074000FF4F0C94D70A84B5806884BDC7B6D8C0CC
:1007500084B5806284BDC8B6D63C08091800080681C
:1007600080938000D093800C09380059C0809105
:100770008000806280938000D093800C0938A00B9
:100780004FC0809180008868080938000D0938D005E
:10079000C0938C0045C08091900080688093900049
:1007A000D0939000C09390003BC0809190008062E4
:1007B00080939000D0939000C0939A0031C08091A9
:1007C00090008868080939000D093900C0939C001F
:1007D00027C08091C00080688093C0008091C000D5
:1007E0008F7B8093C000C093CF001AC08091C0005F
:1007F00080628093C000C093D00012C08091C2007C
:10080000886808093C2008091C2008B7F8093C20079
:10081000C093D20005C0C038D1050CF07CF76CF94
:10082000DF91CF911F9108953FB7F89480914D01CA
:1008300090914E01A0914F01B091500126B5A89B17
:1008400005C02F3F19F00196A11DB11D3FBFBA2F62
:10085000A92F982F8827820F911DA11DB11DBC01C2
:10086000DC0142E0660F771F881F991F4A95D1F787
:1008700008958F929F92AF92BF92CF92DF92EF92A4
:10088000FF926B017C010E9414044B015C01C114B6
:10089000D104E104F104F100E9414044D01C10B0165
:1008A00088199909A09B08983E9340A105B10599
:1008B00070F321E0C21AD108E108F1088EE880E31
:1008C00083E0981EA11CB11CC114D104E104F10401
:1008D00019F7DDCFF90EF90DF909FB09FA9F092F
:1008E0009F908F9008958F929F92AF92BF92CF92D8
:1008F000DF92EF92FF920F931F93CF93DF935C01F0
:100900006B018A0180910B01882309F45EC08091FC
:100910006B01882309F459C08091580180FF05C0FC
:100920008091E00082608093E000E8018AEFF82E79
:1009300093E0E92E2FE3822E2EA3932E209709F4D9
:100940003DC08FB7F894E092E9009091E80095FFE0
:1009500006C09091F200282D291B922F01C090E033
:100960008FBF91110AC09A94FF2079F161E070E025
:1009700080E090E00E943904E1CF292F30EC217D7
:10098000D3070CF49C2F8FB7F894E092E9002091E4
:10099000E80025FF11C0292F30EC217D30BF60160
:1009A000915020F041914093F100FACFC20E31E36
:1009B0009091E800951F1308FBFC0CF5D9884E68B
:1009C000809356011016110644F081E090E0F50185
:1009D0009383828380E090E005C080103C09092B9
:1009E000E800EACDF91CF911F910F91FF90EF9038
:1009F000DF90CF90BF90AF909F908F900895B0C01F3
:100A000009791F0FC01019000209EF73197AF01C8
:100A1000481B590BE0916C01F0916D010280F3814C
:100A2000E02D8CE691E0099480E090E00895CF936A
:100A3000DF930E94FF04EC0187E291E00E94FF0433
:100A40008C0F9D1FDF91CF9108958AE291E00E9463
:100A5000170560E082E00E94DD0261E083E00E9411
:100A6000DD0260E084E00E94DD0261E087E00E943A
:100A7000DD0282E391E00E94D170560E084E00E94BD
:100A8000DD0261E087E00E94DD0261E082E00E9419
:100A9000DD0261E083E00E94DD0288E391E00E94D6
:100AA000170560E082E00E94DD0260E083E00E94C2
:100AB000DD0261E084E00E94DD0261E087E00E949E
:100AC000DD028DE391E00E94170560E083E00E9463
:100AD000DD0261E082E00E94DD0260E087E00E94CA
:100AE000DD0261E084E00E94DD02020E40A1F92C2
:100AF000F920FB60F9211248F939F938091E10074
:100B00009091E100937F9093E10083FF0FC01092DA
:100B1000E90091E09093EB001092EC0092E3909347
:100B2000ED0010926B0198E09093F00082FF20C0DE
:100B300093E09093E09093F1F200992319F0A03E1
:100B40009093E80090915601992339F090915601C5
:100B5000915090935601992389F190916A019923CB
:100B600039F090916A01915090936A01992341F173
:100B700084FF10C08091E2008E7E81608093E2004D
:100B80008091E1008F7E8093E100809158018E7EFC
:100B9000806111C080F16C08091E2008E7E80616E
:100BA0008093E2008091E1008E7E8093E10080914D
:100BB00058018E7E81608093580104C05D9ACDF2C
:100BC000289AD6CF9F918F910F900F0E9091F90B4
:100BD00018951F920F920FB60F921124CF92DF92A9
:100BE000EF92FF920F931F932F933F934F935F9337
:100BF0006F937F938F939F93AF93BF93CF93EF93FF93E5
:100C0000CF93DF93CDB7DEB76C97DEBFCDBF109229
:100C1000E9008091E80083FFFAC168E0CE014596C3
:100C20000E94280282EF8093E80088987FF05C02B
:100C30009091E8009093E009FFCFC0809EF9093E800F6
:100C4000982F907609F0C6C09E892F89188D911132
:100C50000CC0803829F480915901893F10002C0C2
:100C60001092F100192F10047C1422F50E0512B39
:100C7000913051F4811140C14130510509F03CC11E
:100C8000809159018D7F0BC0933061F4811134C183
:100C90004130510509F030C1809159018260809343
:100CA00059012AC1953041F48091E80080FFFCCFC2
:100CB00020682093E30020C1963009F05FC0EB8CE0
:100CC000FC81092E90010925501109254011230E0
:100CD00091F510925301109252010E94F60199E091
:100CE000E0616F5F7F4FD0E1E92F1D92EA95E9F7A7
:100CF00099831A8391E09E8309EA98879AEF998767
```


Screen 1

```

when close .Click
do close application

when next .Click
do open another screen screenName " Screen2 "

```

Screen 2

```

when close .Click
do close application

when back .Click
do open another screen screenName " Screen1 "

when next .Click
do
  if v . Visible = true
  then open another screen screenName " Screen6 "
  else if x . Visible = true
  then call Notifier1 .ShowAlert
        notice " Incorrect Login "
        set Notifier1 . BackgroundColor to 
        set Notifier1 . TextColor to 
  else call Notifier1 .ShowAlert
        notice " Invalid Username and Password !!!!!!! "
        set Notifier1 . BackgroundColor to 
        set Notifier1 . TextColor to 
arnings

```

```

when register .Click
do open another screen screenName " Screen3 "

when Screen2 .OtherScreenClosed
otherScreenName result
do
  if get otherScreenName = " Screen3 "
  then
    call TinyDB1 .StoreValue
      tag select list item list get result
      index 1
      valueToStore " user "
    call TinyDB1 .StoreValue
      tag select list item list get result
      index 2
      valueToStore " password "

  initialize global user to " "
  initialize global password to " "

```

```

when login.Click
do
  set global user to call TinyDB1.GetValue
  tag TextBox1.Text
  valuelTagNotThere ""
  set global password to call TinyDB1.GetValue
  tag PasswordTextBox1.Text
  valuelTagNotThere ""
  if
    get global user = "user" and get global password = "password"
  then
    set x.Visible to false
    set v.Visible to true
    call Notifier1.ShowAlert
    notice "Correct Login Click to Next"
    set Notifier1.BackgroundColor to 
    set Notifier1.TextColor to 
  else
    set x.Visible to true
    set v.Visible to false
    call Notifier1.ShowAlert

when reset.LongClick
do
  call TinyDB1.ClearAll

```

Screen 3

```

when close.Click
do
  close application

when back.Click
do
  open another screen screenName "Screen2"

when ok.Click
do
  if
    TextBox1.Text = TextBox2.Text and PasswordTextBox1.Text = PasswordTextBox2.Text
  then
    close screen with value result
    make a list
    TextBox1.Text
    PasswordTextBox1.Text
  else
    call Notifier1.ShowAlert
    notice "Incorrect Login"
    set Notifier1.BackgroundColor to 
    set Notifier1.TextColor to 

```

Screen 4

```

when close .Click
do close application

when cmph .Click
do open another screen screenName " Screen7 "

when cvph .Click
do open another screen screenName " Screen8 "

when back .Click
do open another screen screenName " Screen1 "

```

Screen 5

```

when close .Click
do close application

when back .Click
do open another screen screenName " Screen6 "

when connect .BeforePicking
do set connect . Elements to BluetoothClient1 . AddressesAndNames

when connect .AfterPicking
do set connect . Selection to call BluetoothClient1 . Connect
address connect . Selection
set Label1 . BackgroundColor to 
set Label1 . Text to " Connected "
set Label1 . TextColor to 

when disconnect .Click
do call BluetoothClient1 . Disconnect
set Label1 . BackgroundColor to 
set Label1 . Text to " Not connected "
set Label1 . TextColor to 
]

```

```

when off .Click
do if BluetoothClient1 . IsConnected
then call BluetoothClient1 . SendText
text " s "
if not BluetoothClient1 . IsConnected
then call Notifier1 . ShowAlert
notice " Bluetooth is not connected !!! "
set Notifier1 . BackgroundColor to 
set Notifier1 . TextColor to 

when on .Click
do if BluetoothClient1 . IsConnected
then call BluetoothClient1 . SendText
text " e "
if not BluetoothClient1 . IsConnected
then call Notifier1 . ShowAlert
notice " Bluetooth is not connected
set Notifier1 . BackgroundColor to 
set Notifier1 . TextColor to 

```

```

when backward .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
    text "b"
  if not BluetoothClient1 .IsConnected
  then
    call Notifier1 .ShowAlert
    notice "Bluetooth is not connected !!!"
    set Notifier1 .BackgroundColor to 
    set Notifier1 .TextColor to 

when forward .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
    text "f"
  if not BluetoothClient1 .IsConnected
  then
    call Notifier1 .ShowAlert
    notice "Bluetooth is not connected"
    set Notifier1 .BackgroundColor to 
    set Notifier1 .TextColor to 

when right .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
    text "r"
  if not BluetoothClient1 .IsConnected
  then
    call Notifier1 .ShowAlert
    notice "Bluetooth is not connected !!!"
    set Notifier1 .BackgroundColor to 
    set Notifier1 .TextColor to 

when left .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
    text "l"
  if not BluetoothClient1 .IsConnected
  then
    call Notifier1 .ShowAlert
    notice "Bluetooth is not connecte"
    set Notifier1 .BackgroundColor to 
    set Notifier1 .TextColor to 

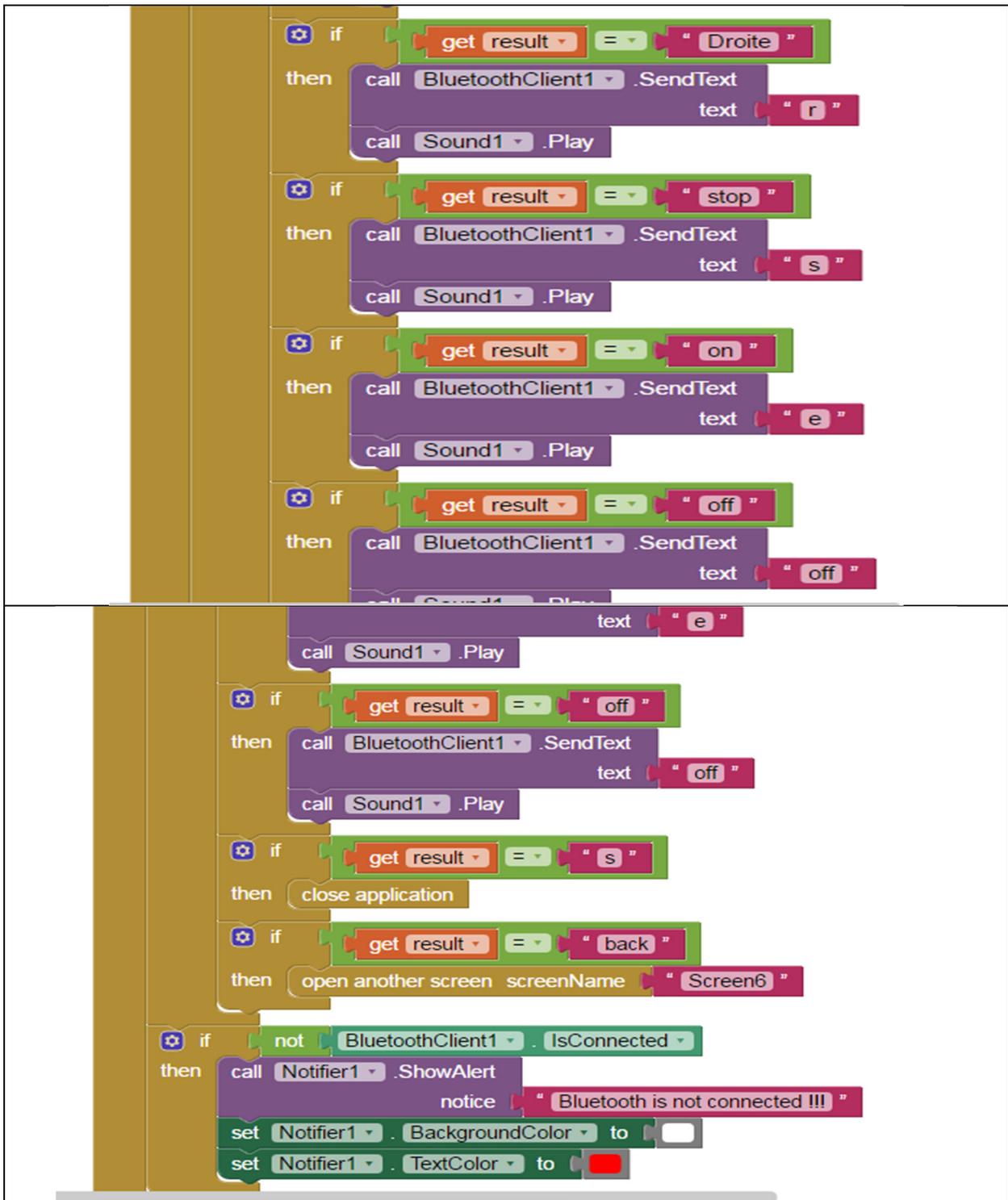
when stop .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
    text "s"
  if not BluetoothClient1 .IsConnected
  then
    call Notifier1 .ShowAlert
    notice "Bluetooth is not connected !!!"
    set Notifier1 .BackgroundColor to 
    set Notifier1 .TextColor to 

when Slider1 .PositionChanged
thumbPosition
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .Send1ByteNumber
    number join round Slider1 .ThumbPosition
    set Label3 .Text to join "Speed variation : "
    get thumbPosition
    "%"
  if not BluetoothClient1 .IsConnected
  then
    call Notifier1 .ShowAlert
    notice "Bluetooth is not connected !!!"
    set Notifier1 .BackgroundColor to 
    set Notifier1 .TextColor to 
  
```

Screen 6

The screenshot displays a visual programming interface with the following code blocks:

- Screen Navigation:**
 - when `close` .Click do `close application`
 - when `back` .Click do `open another screen` screenName `" Screen6 "`
- Bluetooth Connection Management:**
 - when `connect` .BeforePicking do `set connect` .Elements to `BluetoothClient1` .AddressesAndNames
 - when `connect` .AfterPicking do:
 - `set connect` .Selection to `call BluetoothClient1` .Connect address `connect` .Selection
 - `set Label1` .BackgroundColor to `white`
 - `set Label1` .Text to `" Connected "`
 - `set Label1` .TextColor to `green`
 - when `disconnect` .Click do:
 - `call BluetoothClient1` .Disconnect
 - `set Label1` .BackgroundColor to `white`
 - `set Label1` .Text to `" Not connected "`
 - `set Label1` .TextColor to `red`
- Proximity and Voice Control:**
 - when `ProximitySensor1` .ProximityChanged distance:
 - do `if BluetoothClient1` .IsConnected and `ProximitySensor1` .Distance = `0` then `call SpeechRecognizer1` .GetText
 - `if not BluetoothClient1` .IsConnected then `call Notifier1` .ShowAlert notice `" Bluetooth is not connected !!! "`
 - `set Notifier1` .BackgroundColor to `white`
 - `set Notifier1` .TextColor to `red`
 - when `cv` .Click do:
 - `if BluetoothClient1` .IsConnected then `call SpeechRecognizer1` .GetText
 - `if not BluetoothClient1` .IsConnected then `call Notifier1` .ShowAlert notice `" Bluetooth is not connected "`
 - `set Notifier1` .BackgroundColor to `white`
 - `set Notifier1` .TextColor to `red`
- Voice Command Logic:**
 - when `SpeechRecognizer1` .AfterGettingText result:
 - do `if BluetoothClient1` .IsConnected then:
 - `set Label3` .Text to `get result`
 - `if get result` = `" Avant "` then `call BluetoothClient1` .SendText text `" f "`
 - `call Sound1` .Play
 - `if get result` = `" Arrière "` then `call BluetoothClient1` .SendText text `" b "`
 - `call Sound1` .Play
 - `if get result` = `" Gauche "` then `call BluetoothClient1` .SendText text `" l "`
 - `call Sound1` .Play



The image displays a sequence of Scratch code blocks for controlling a mobile robot via Bluetooth. The code is organized into two vertical columns. The first column contains four conditional blocks: 'if get result = "Droite"', 'if get result = "stop"', 'if get result = "on"', and 'if get result = "off"'. Each 'if' block is followed by a 'then' block containing 'call BluetoothClient1 .SendText' and 'call Sound1 .Play'. The second column continues with a 'call Sound1 .Play' block, followed by another 'if get result = "off"' block with 'then' actions 'call BluetoothClient1 .SendText' and 'call Sound1 .Play'. This is followed by an 'if get result = "s"' block with a 'then' action 'close application'. Next is an 'if get result = "back"' block with a 'then' action 'open another screen screenName "Screen6"'. The final block is an 'if not BluetoothClient1 .IsConnected' block with a 'then' action 'call Notifier1 .ShowAlert', followed by 'set Notifier1 .BackgroundColor to' (with a white color swatch) and 'set Notifier1 .TextColor to' (with a red color swatch).

```
if [get result] = ["Droite"]
then
  call BluetoothClient1 .SendText
  text ["r"]
  call Sound1 .Play

if [get result] = ["stop"]
then
  call BluetoothClient1 .SendText
  text ["s"]
  call Sound1 .Play

if [get result] = ["on"]
then
  call BluetoothClient1 .SendText
  text ["e"]
  call Sound1 .Play

if [get result] = ["off"]
then
  call BluetoothClient1 .SendText
  text ["off"]
  call Sound1 .Play

call Sound1 .Play

if [get result] = ["off"]
then
  call BluetoothClient1 .SendText
  text ["off"]
  call Sound1 .Play

if [get result] = ["s"]
then
  close application

if [get result] = ["back"]
then
  open another screen screenName ["Screen6"]

if [not BluetoothClient1 .IsConnected]
then
  call Notifier1 .ShowAlert
  notice ["Bluetooth is not connected !!!"]
  set Notifier1 .BackgroundColor to [white]
  set Notifier1 .TextColor to [red]
```