



République Algérie Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique  
Université Akli Mohand Oulhadj- Bouira



Faculté des Sciences et des Sciences Appliquées  
Département : **Génie Electrique**

**Mémoire de fin d'étude**

Présenter par :

**KECHADI Assia**

**BAHLOUL Ouazna**

En vue de l'obtention du diplôme master

Filière : **Electronique**

Spécialité : **Electronique des Système Embarqués**

**Intitulé du sujet :**

Réalisation d'un prototype Self Cleaner à  
base d'Arduino

**Date de soutenance : 06/11/2021**

**Devant le jury composé de**

Président	Mr. NOURINE Mourad	UAMOB
Examinatrice	Mm. AGGOUN Ghania	UAMOB
Encadreur	Mr. AIT ABBAS Hamou	UALOB

**Année Universitaire 2020/2021**

# Remerciement

Nous tenons à remercier notre Dieu, le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.

Nos remerciements les plus chaleureux vont à nos **chers parents** pour leurs encouragements, leur patience, et leur grand soutien durant toutes ces années d'études.

Nos remerciements s'adressent également au docteur **AIT ABBAS.H**, pour son encadrement, sa motivation professionnelle, son orientation afin de donner notre mieux.

Nos profondes gratitudes aux membres du jury qui ont l'honneur d'évaluer ce travail de fin d'études.

Tous nos infinis remerciements vont à tous les enseignants qui ont contribué à notre Formation durant notre cursus universitaire en particulier, monsieur **ISSAOUNI.S, AIT ABBAS.H, SAOUD.B, MOUDACHE.S, MEDJEDOUB.S.**

Enfin, nous remercions tous ceux qui ont participé de près ou de loin à la réalisation de ce Mémoire, en particulier monsieur **BOUACHERINE.Y.**

Last but not least, we wanna thank us

We wanna thank us for believing in us

We wanna thank us for doing all this hard work

We wanna thank us, for never quitting

We wanna thank us for tryna do more right than wrong.

---

# Dédicace

Ce mémoire représente bien plus que de simples travaux. Ce mémoire est la finalité de cinq longues années d'études. C'est pourquoi je tiens à exprimer ma reconnaissance et ma sympathie bien que je dédie ce travail :

A mes chers parents, mes piliers, que dieu me les gardes

A mes frères et mes sœurs que j'adore

**AISSA, ZAHRA, THELLELI, YOUVA**

A mon âme sœur **YASMINE**

A celles que je considère comme des sœurs

**ASSIA, MARINA, FANI, NADIA, SARA**

A la mémoire de ma grand-mère maternelle

A ma binôme **ASSIA** bien sûr, qui a patienté avec moi, milles merci et sa famille

A tous les membres de ma grande famille

...

           **Ouazna**

# Dédicace

Ce mémoire représente bien plus que de simples travaux. Ce mémoire est la finalité de cinq longues années d'études. C'est pourquoi je tiens à exprimer ma reconnaissance et ma sympathie bien que je dédie

ce travail :

A ma précieuse source de bonheur **ma mère**

A **mon cher père** qui n'a pas cessé de donner à fond

A mes sœurs

**THAMAZGHA, YASMINE, LAMIS, SAFAA**

A mes chères amies que j'adore

**OUAZNA, MARINA, SARA, FANI, NADIA**

**DYHIA, NACIMA, G. SARA**

A ma binôme **OUAZNA**

A ma grand-mère maternelle

A ma grand-mère paternelle

...

Assia

# Table des matières

Liste des figures.....	I
Liste des tableaux .....	II
Introduction Générale.....	01

## Chapitre I : Généralité sur la robotique

I.1 Introduction .....	04
I.2 Définitions .....	04
I.2.1 Définition de la Robotique .....	04
I.2.2 Etymologies (origine) du mot Robot .....	04
I.2.3 Définition d'un Robot .....	04
I.3. Application de la robotique .....	05
I.3.1 La robotique d'intervention .....	05
I.3.2 Robotique de service professionnelle.....	06
I.3.3 Robotique personnelle .....	06
I.4 Les types de robots .....	06
I.4.1 Les humanoïdes .....	06
I.4.2 Les robots industriels (manipulateurs).....	06
I.4.2.1 Robot à séquence de travail évolutive .....	07
I.4.2.2 robots copieur .....	07
I.4.2.3 robots à commande numérique .....	07
I.4.2.4 robot intelligent .....	07
I.4.3 Les robots mobiles (explorateurs) .....	07
I.4.3.1 Les robots à roues .....	07
I.4.3.2 Les robots à pattes.....	10
I.5. Domaines d'application .....	12
I.6 Conclusion .....	13

## Chapitre II : ROBOT MOBILE

II.1 Introduction .....	15
II.2 Définition d'un robot mobile.....	15
II.3 Grandes classes des robots mobiles.....	15
II.4 Mode de fonctionnement .....	16
II.5 Autonomie d'un robot mobile .....	16
II.5.1 Contrôle.....	18

II.5.2 Actionneurs .....	18
II.5.3 Effecteurs .....	19
II.5.4 Capteurs .....	19
II.6 Constitution d'un robot mobile autonome .....	19
II.6.1 Structure mécanique .....	20
II.6.2 Charge utile .....	21
II.6.3 Structure de commande .....	21
II.6.3.1 Perception de l'environnement .....	21
II.6.3.2 Communication homme-machine .....	22
II.6.3.3 Commande .....	22
II.7 Navigation autonome des robots mobiles .....	22
II.7.1 Planification de mouvement .....	23
II.7.2 Localisation .....	23
II.7.3 Suivi de trajectoire .....	24
II.7.4 Évitement d'obstacles .....	24
II.7.5 Parking .....	24
II.8 Architectures de contrôle des robots mobiles .....	25
II.8.1 Approche délibérative .....	25
II.8.2 Approche réactive .....	26
II.8.3 Approche hybride .....	28
II.8.4 Approche collective .....	28
II.9 Conclusion .....	29

## **Chapitre III : Etude et Conception du Robot**

III.1 Introduction .....	31
III.2 Cahier de charge .....	31
III.3 Les structure de robot mobile .....	32
III.3.1 Structure mécanique .....	32
III.3.1.1 le Châssis .....	32
III.3.1.2 Moyens de locomotion .....	33
III.3.1.2.1 Roues .....	33
III.3.1.3 Moteur à courant continu (MCC) .....	34
III.3.1.4 La pompe .....	35
III.3.2 Structure électronique .....	36
III.3.2.1 Arduino Uno .....	36

III.3.2.2 Le module Bluetooth .....	40
III.3.2.3 Driver Moteur L293D .....	41
III.3.2.4 Le capteur ultrasonique .....	43
III.3.2.5 Un relais électronique .....	45
III.3.3 Présentation des Logiciel .....	46
III.3.3.1 Le logiciel Arduino IDE .....	46
III.3.3.1.1 Approche et utilisation de logiciel .....	47
III.3.3.1.2 Structure d'un programme Arduino .....	49
III.3.3.2 Logiciel Proteus.....	52
III.3.3.2.1 ISIS .....	52
III.4 Tableau devis .....	53
III.5 Conclusion .....	53

## **Chapitre IV : Réalisation du robot**

IV.1 Introduction .....	56
IV.2 Schéma synoptique .....	56
IV.3 Déroulement de projet .....	56
IV.4 Réalisation de projet .....	57
IV.4.1 Structure/Châssis .....	57
IV.4.2 Opérateur mécanique .....	57
IV.4.3 Unité de contrôle .....	58
IV.4.4 Entrées/ Capteurs .....	58
IV.4.5 Alimentation .....	59
IV.4.6 Branchement des différentes parties de projet .....	59
IV.4.7 Schéma électrique du projet .....	62
IV.6 Application Android.....	63
IV.7 Conclusion.....	64

<b>Conclusion Générale.....</b>	<b>71</b>
---------------------------------	-----------

### **Bibliographies**

### **Annexes**

---

## Liste Des Figures

Figure I.1: Architecture d'un Robot.....	5
Figure I.2: Robot de type Unicycle.....	8
Figure I.3: Robot de type tricycle.....	8
Figure I.4: Robot de type voiture.....	9
Figure I.5: Robot de type Omnidirectionnel.....	9
Figure I.6: Robot a patte.....	6
Figure I.7: Robot bipède.....	12
Figure I.8: Robot Quadrupède de MIT leg LAB.....	12
Figure II .1 : Interaction entre le robot et l'environnement.....	17
Figure II.2 : Etapes de traitement automatique.....	18
Figure II.3: Différents types d'actionneurs.....	18
Figure II.4: Différents types des capteurs.....	19
Figure II.5: Schéma synoptique de principe d'un robot.....	19
Figure II.6: Décomposition d'un robot mobile autonome.....	20
Figure II.7: Processus de navigation autonome.....	23
Figure II.8: Architectures de contrôle pour les robots mobile.....	25
Figure II.9 : Architecture Hiérarchique.....	25
Figure II.10: Architecture réactive.....	27
FigureII.11 : Architecture modulaire de Subsumption.....	27
Figure III.1: Schéma bloc détaillés de cahier des charges.....	32
Figure III.2: Les Roues.....	33
Figure III.3: Moteur à courant continu.....	34

---

Figure III.4: pompe électronique 12v.....	35
Figure III.5: Microcontrôleur ATmega 328.....	38
Figure III. 6 : Constitution de la carte Arduino Uno.....	40
Figure III.7: Module Bluetooth HC-06.....	41
Figure III.8: Driver Moteur L293D.....	42
Figure III.9: Pont-H L293D.....	42
Figure III.10: Capteur sonar à ultrasons HC-SR04.....	44
Figure III.11: Fonctionnement du capteur ultrason.....	45
Figure III.12: Le module relais 5V.....	46
Figure III.13: L'interface de l'IDE Arduino.....	47
Figure III.14: Le menu fichier.....	48
Figure III.15: Le menu outils.....	48
Figure III.16: La barre d'outils.....	49
Figure III.17: Exemple d'un programme Arduino.....	49
Figure III.18: Fenêtre du module de simulation Proteus ISIS.....	53
Figure IV.1: Schéma synoptique de projet.....	56
Figure IV.2: Préparation et installation des moteurs.....	58
Figure IV.3: La connexion entre L'Arduino et le driver moteur et les moteurs sur Proteus.....	59
Figure IV.4: Branchement et simulation de Bluetooth avec Arduino.....	59
Figure IV.5: Montage réel de capteur HC-SR04 avec Arduino.....	60
Figure IV.6 : Simulation d'Arduino avec capteur HC-SR04.....	60
Figure IV.7: Simulation du robot intelligent sur Proteus.....	61
Figure IV.8: Montage réel de robot intelligent.....	61

---

---

Figure IV.9: Schéma électrique de projet.....	62
Figure IV.10: Organigramme d'éviteur d'obstacle / de commande de la pompe à eau.....	62
Figure IV.11: Capteur d'écran de l'interface de l'application .....	63
Figure IV.12 : Réalisation finale.....; ; ;	63

---

## Liste des Tableaux

Tableau I.1 Avantage et Inconvénient Des Robots.....	10
Tableaux I.2: Domaines d'application.....	13
Tableau III.1 : Caractéristique de la carte Arduino.....	37
Tableau III.2 Les commandes du programme Arduino.....	51
Tableau III.3 Les fonctions du programme Arduino.....	52
Tableau III.4 Les différentes variables du programme Arduino.....	52

# **Introduction Générale**

Les systèmes technologiques sont au cœur de nombreuses applications permettant d'aider l'humain dans des tâches dangereuses, trop complexes ou impossibles. Parmi ces applications, certaines peuvent blesser des humains, ou provoquer des dégâts sur les biens matériels ou l'environnement. De tels systèmes sont qualifiés de systèmes à sécurité critique et leur utilisation est conditionnée par la confiance que l'humain leur accorde.

Comme la robotique est l'étude de la conception d'automatismes intelligents capables d'imiter ou de remplacer certaines fonctions relevant de l'humain afin d'interagir avec le monde physique. Aussi, un robot mobile est un système mécanique, électronique et informatique agissant physiquement sur son environnement en vue d'atteindre un objectif qui lui a été assigné. Cette machine douée d'une intelligence artificielle, doit être capable de s'adapter à certaines variations de ces conditions de fonctionnement. Ainsi le robot est une machine qui peut apprendre à faire des travaux répétitifs, dangereux ou aller dans des endroits inaccessibles (très petits) ou hostiles (comme dans l'espace ou sur la mer). Le robot est donc une machine ou un dispositif qui fonctionne de façon automatique ou en réponse à une commande à distance. Cette machine est dotée de faculté de perception, de décision et d'action.

L'objectif principal de notre travail est de concevoir et de réaliser un robot mobile contrôlé par smartphone qui nettoie le sol c'est-à-dire le robot aura comme tâche principale le nettoyage (humidifier et faire sécher le sol avec des vadrouilles) de grandes et petites surfaces. Le système qu'on a décidé de concevoir est essentiellement basé en deux phases majeure : la conception d'une base mécanique à quatre roues puis l'intégration d'une plateforme robotique avec l'électronique de commande (Arduino) et microcontrôleur.

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 est consacré à l'étude d'un robot mobile. Un aperçu général sur le mode de fonctionnement et les grandes classes des robots mobiles, les constitutions d'un robot mobile autonome seront abordées ainsi, les types des robots mobiles à roues et la modélisation de chaque type.
- Le troisième chapitre est basé sur la conception matérielle et logicielle du robot et sa commande à distance.
- Le quatrième chapitre détaille la simulation et la réalisation de notre prototype robot. Nous terminerons ce manuscrit par une conclusion générale récapitulant qui a été fait et expose les perspectives de ce travail.

# Chapitre I

## **Généralité sur la robotique**

## I.1 Introduction

Quotidiennement nous effectuons de nombreuses tâches répétitives, pénibles, dangereuses, complexes, que ce soit au domicile ou au travail. C'est la raison pour laquelle, les chercheurs et les scientifiques se sont intéressés à fabriquer un nouveau type de machines capables d'exécuter une variété de tâches sur commande ou selon les instructions programmées à l'avance.

Dans ce chapitre, nous allons voir des généralités sur la robotique et les robots mobiles ainsi ses domaines d'application.

## I.2.Définitions:

### I.2.1. Définition de la Robotique :

La robotique définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles [1].

### I.2.2. Etymologies (origine) du mot Robot :

Le terme « Robot » a été introduit en 1920 par l'écrivain tchèque Karel Capek dans sa pièce de théâtre « Rossum's Universal Robots ». Ce terme, provenant du tchèque ROBOT « travail forcé », désignait à l'origine une machine androïde capable de remplacer l'homme dans toutes ses tâches. En 1942, le mot « robotique » fut introduit dans la littérature par " **Isaac Asimov**" dans son livre "**Runaround**" [1]. Il y énoncé les « trois règles de la robotique », qui sont comme suit

- **Loi 01** : Un robot ne peut blesser un être humain ni par son action ni par son inaction.
- **Loi 02** : Un robot doit obéir aux ordres donnés par les êtres humains sauf si tels ordres sont en contradiction avec la première loi.
- **Loi 03** : Un robot protège sa propre existence aussi longtemps qu'une telle protection n'est pas en contradiction avec la première et/ou la deuxième loi.[01]

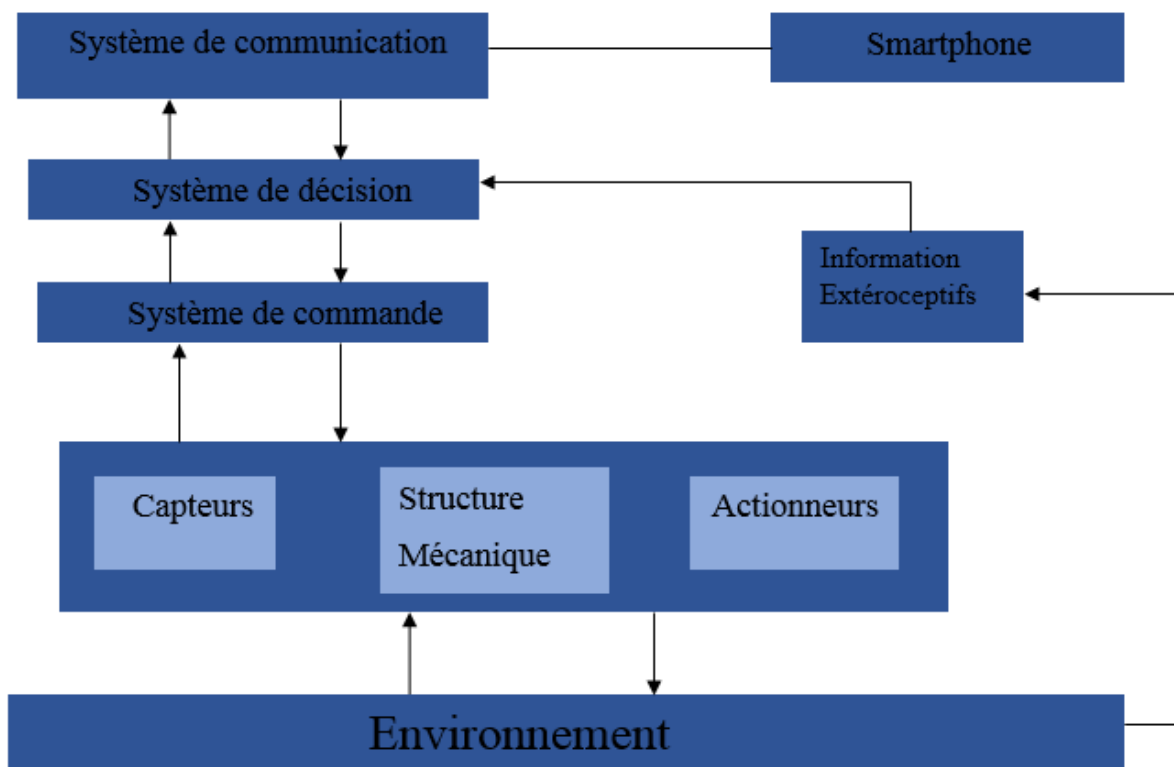
### I.2.4. Définition d'un Robot :

Un robot est un appareil dirigé à l'aide d'un système de commande automatique à base de microprocesseurs et qui été conçu pour accomplir une tâche précise, dans le domaine industriel, scientifique ou encore domestique [02].

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

- **Mécanique** : conception, réalisation, modélisation des robots.
- **Electronique** : mise en place de composants pour les robots en vue de les commander à distance en utilisant les concepts de télécommunications.

- **Informatique** : création de programmes destinés à la gestion du transfert d'informations entre les différents composants du robot.
- **Automatique** : commande, calibrage des capteurs, des effecteurs, identification des paramètres.
- **Traitement du signal** : analyse des informations enregistrées par les capteurs du robot.
- **Mathématiques** : modèles mathématiques pour la prise de décisions ou/et l'apprentissage, calcul de trajectoires, localisation, planification.
- **Sciences cognitives** : interactions homme-machine, machine-machine, prise de décisions.



**Figure I.1** Architecture d'un Robot.

### I.3. Application de la robotique :

Les applications de la robotique peuvent être classées selon trois grandes catégories à savoir : la robotique d'intervention, la robotique de service professionnel et la robotique personnelle [04].

#### I.3.1. La robotique d'intervention :

La robotique d'intervention est caractérisée en général par des robots télé opérés (opérés à distance) à travers des commandes directes (joysticks, bras maître et autres organes de commandes

physiques ou virtuels), ou des commandes par des ordres de haut niveau pour réaliser et enchaîner des tâches (« va à telle position », « prends cette pièce désignée sur un écran »...). L'opérateur est donc systématiquement dans la boucle de commande du robot afin d'interagir et l'aider à réaliser sa mission en fonction des évolutions de l'environnement matériel et humain.

La robotique d'intervention est en général utilisée pour remplir des tâches dans des environnements difficiles d'accès ou dangereux pour les humains (industries nucléaire et militaire, inspection et maintenance, interventions dans des catastrophes naturelles, exploration marine, encore exploration spatiale (mission sur mars), ...) ou encore lorsque l'absence d'humain rend l'exploitation plus aisée ou plus efficace [4].

### **I.3.2. Robotique de service professionnelle :**

Ses fonctions sont principalement de soulager les professionnels de tâches répétitives ou dangereuses (dans une optique proche de la robotique industrielle), ou de les assister dans des interventions qui requièrent un niveau de précision ou des qualités inaccessibles à l'opérateur humain (agriculture, nettoyage, construction et démolition, transport automatique de personnes et de biens, relations publiques, médecine, assistance au geste, sécurité et surveillance...) [4].

### **I.3.3. Robotique personnelle :**

Constitue un champ potentiellement aussi foisonnant que la robotique de service professionnel. Les utilisations sont possibles de répondre à un grand nombre de besoins et d'attentes (réalisant des tâches domestiques d'entretien de la vie courante, assistance aux personnes en perte d'autonomie, éducation, jeux, surveillance domestique...) [4].

## **I.4. Les types de robots :**

Aujourd'hui la classification des robots est faite dans l'objectif de donner une idée de la présence actuelle de la robotique et de la portée de ce domaine dans les années à venir...

On va aborder aujourd'hui les trois catégories de robots les plus communes.

### **I.4.1. Les humanoïdes :**

C'est la catégorie la plus connue, en grande partie grâce à leur promotion faite par la science-fiction, elle regroupe tous les robots anthropomorphes, ceux dont la forme rappelle la morphologie humaine. Ces robots ont généralement un torse, une tête, deux bras et deux jambes. Parfois, certains de ces robots ne représentent qu'une partie du corps. Lorsqu'un robot anthropomorphe imite non seulement l'apparence physique, mais aussi les comportements humains, on l'appelle un androïde.

### **I.4.2. Les robots industriels (manipulateurs) :**

La majorité de ces robots sont à base fixe. Quand la base n'est pas fixe, elle est généralement montée à un rail. On retrouve dans cette catégorie les robots de manipulation, type « Pick And Place

», des robots soudeurs ou encore des robots de peintures. Cela représente la majorité des robots actuellement en état de service. [.....]

#### **I.4.2.1 robot à séquence de travail évolutive :**

Ce type de robot a les mêmes caractéristiques que le robot précédent, Il se positionne automatiquement de façon éventuellement répétitive en des sites prédéterminés, cependant dans ce système ; la séquence des diverses opérations que doit effectuer le robot est facilement modifiée [5].

#### **I.4.2.2 robots copieur :**

Ce type de robot est capable de reproduire des mouvements qui ont été exécutés sous contrôle d'un opérateur humain, et qui ont été enregistrés dans une mémoire, par exemple la position du robot à un instant donné et aussi la tâche qu'il doit réaliser en chaque point de sa trajectoire [5].

#### **I.4.2.3 robots à commande numérique :**

Ce robot présente la particularité d'être contrôlé de façon numérique, il exécute des opérations enregistrées comme un robot copieur et il est directement programmable [5].

#### **I.4.2.4 robot intelligent :**

Ce type de robot est doté de moyens sensoriels de perception de l'environnement comme le sens visuel, le sens tactile et il est capable de s'adapter aux changements éventuels en continuant à exécuter une tâche donnée, programmée ou enregistrée à l'avance [05].

### **I.4.3 Les robots mobiles (explorateurs) :**

Un robot mobile est un système mécanique, électronique et informatique agissant physiquement sur son environnement en vue d'atteindre un objectif qui lui a été assigné. Cette machine est polyvalente et capable de s'adapter à certaines variations de ses conditions de fonctionnement. Elle est dotée de fonctions de perception, de décision et d'action. Ainsi, le robot devrait être capable d'effectuer des tâches diverses, de plusieurs manières, et accomplir correctement sa tâche, même s'il rencontre de nouvelles situations inattendues.

#### **I.4.3.1 Les robots à roues :**

La très grande majorité de robots mobiles développés jusqu'à ce jour sont équipés de roues plutôt que de jambes. Les principales raisons de ce choix technologique sont les suivantes : les roues sont plus faciles à contrôler, elles procurent une meilleure stabilité au véhicule, elles dissipent moins d'énergie et elles permettent au véhicule de se déplacer plus rapidement.

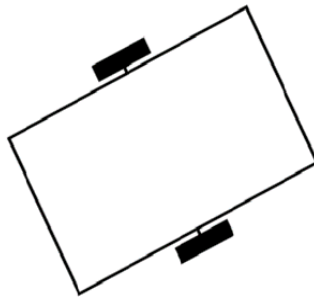
Par contre, les roues comportent aussi un certain nombre d'inconvénients face aux mécanismes à jambes. Entre autres, elles ne sont utilisables que sur des terrains relativement plats et dont la surface est dure. Sur des terrains accidentés, les roues doivent être plus grandes que les obstacles qu'elles risquent de rencontrer. Dans le cas de surfaces moins stables, les roues peuvent glisser ou s'enliser, ce qui pose des difficultés de contrôle importantes.

La disposition des roues d'un robot mobile peut varier sensiblement d'un véhicule à l'autre en fonction des caractéristiques recherchées [06].

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

➤ **Robot Unicycle :**

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. C'est un robot non-holonome, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion. Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites (Figure I.8)

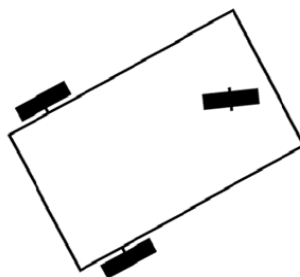


**Figure I.2** Robot de type Unicycle

➤ **Robot Tricycle :**

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues axes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues axes et de l'axe de la roue orientable.

C'est un robot non holonome. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues axes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limite de la roue orientable (Figure I.9).



**Figure I.3** Robot de type tricycle

➤ **Robot Voiture :**

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues axes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe.

Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé (Figure I.10).

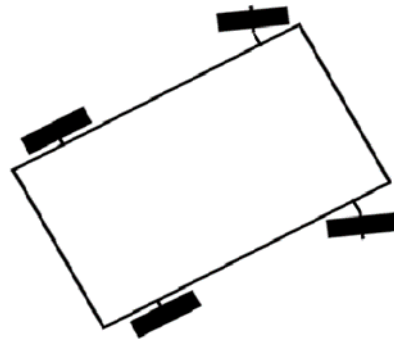


Figure I.4 Robot de type voiture

➤ **Robot omnidirectionnel :**

Un 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.

L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande (Figure I.11).

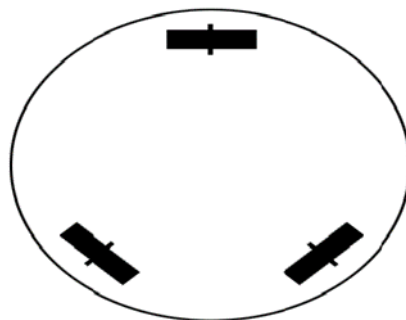


Figure I.5 Robot de type Omnidirectionnel

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

<p><b>Robot Unicycle</b></p>	<p>(-) Non-holonome (+) Stable                  (+) Rotation sur soi-même                  (+) Complexité mécanique faible</p>
------------------------------	--

<b>Robot Tricycle</b>	(-) Non-holonome (-) Peu stable (-) Pas de rotation sur soi-même (+) Complexité mécanique modérée
<b>Robot Voiture</b>	(-) Non-holonome (+) Stable (-) Pas de rotation sur soi-même (+) Complexité mécanique modérée
<b>Omnidirectionnel</b>	(+) Holonome (+) Stable (+) Rotation sur soi-même (-) Complexité mécanique importante

**Tableau I.1:** Avantage et Inconvénient Des Robots.

#### I.4.3.2 Les robots à pattes :

Des plates-formes à deux, quatre ou six pattes peuvent également être utilisées. Les plates-formes à six pattes sont relativement pratiques car le robot est en équilibre permanent, ce qui facilite le contrôle. Les plates-formes à deux ou quatre pattes sont plus complexes à commander et le simple contrôle de la stabilité et d'une allure de marche correcte reste aujourd'hui difficile, ce qui les rend en général relativement lentes. Ces différents facteurs font que ces plates formes sont rarement utilisées quand l'application visée a un besoin précis de positionnement et de navigation.

Les premiers dessins ou mécanismes marcheurs ont été réalisés vers la fin du 18<sup>ième</sup> siècle. Une évolution simultanée des robots marcheurs bipèdes et à plus de deux pattes s'est déroulée depuis les années 1960. L'intérêt des chercheurs pour les robots marcheurs n'est pas nouveau. Dans les années 1960, la compagnie General Electric, a présenté le prototype d'un camion marcheur à quatre pattes complètement hydrauliques avec des commandes analogiques. Ce prototype était commandé par un opérateur dans la cabine de pilotage. Il ajustait la position des pattes avant et arrières avec ses bras et ses jambes. Ce robot ne possédait aucun ordinateur. Il utilisait un moteur à combustion de 67 kW qui lui permettait de marcher à une vitesse de 5 M/h et de transporter une charge de 230 kg. Par la suite, dans les années 1970, les robots à apparence d'insecte comme les fourmis sont apparus, possédant de six à huit pattes. Ce nombre de pattes leur permet d'avoir des démarches statiquement stables, plus naturelles et plus rapides. Les systèmes électriques n'étaient pas tous intégrés sur le robot car les composantes étaient encore trop grosses. Au début des années 1990, un groupe à l'Université Carnegie Mellon a développé un robot marcheur assez différent, une table marchante. Ces robots sont très stables grâce à leur grand nombre de pattes qui agissent ensemble. Cependant, ils sont lents et ont une démarche saccadée.

L'évolution des technologies à partir de 1994 a amené des chercheurs comme Kimura à concevoir des robots à forme d'animaux dans le but d'atteindre une plus grande fluidité de mouvement. Le contrôle de ces robots se fait par des réseaux de neurones. Une méthode qui tente d'imiter non seulement la démarche, mais aussi la structure de contrôle que les êtres vivants utilisent. Les robots à pattes sont considérés pour leur mobilité supérieure obtenue grâce à un nombre élevé de degrés de liberté. Le contact avec le sol est discret, ce qui permet une sélection des points d'appui en fonction des conditions locales du terrain. La charge utile est relativement faible, essentiellement à cause d'une puissance installée intrinsèquement élevée. Divers prototypes expérimentaux ont été mis au point, la plupart du temps, pour des activités d'inspection ou d'exploration de milieux peu accessibles. Néanmoins, malgré leurs possibilités attrayantes en matière de mobilité, l'intérêt de l'industrie pour ce type de véhicule se révèle être, à ce jour, encore très limité. Toutefois, ils comportent aussi des avantages. Ils peuvent franchir des obstacles que les roues ne peuvent surmonter, monter et descendre des escaliers, circuler sur des terrains très accidentés, et finalement, ils ont moins tendance à s'enfoncer dans un sol instable. Les robots à pattes ont l'avantage de pouvoir s'adapter à la plupart des terrains. La bipédie n'en est qu'un exemple, puisque l'on retrouve également des robots à 4,6 ou 8 pattes [08].

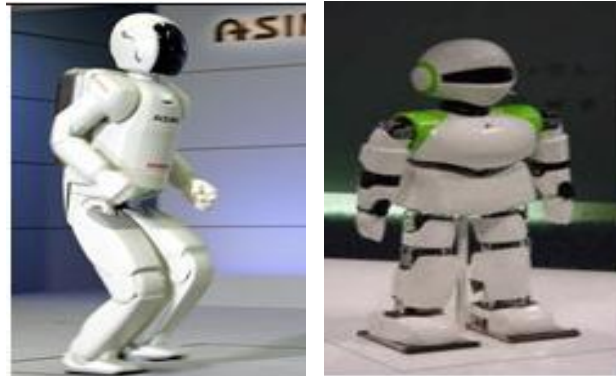


**Figure I.6** Robot a patte.

➤ **Les robots bipèdes :**

Le robot bipède est considéré comme le plus agile parmi les robots marcheurs puisqu'il imite la marche humaine ; dont la souplesse et l'efficacité achevée en font le modèle par excellence de la locomotion à pattes. Ainsi, ce type de robot recueille une grande attention à travers le monde.

L'étude des robots bipèdes reste difficile vu qu'ils ne sont pas statiquement stables comme leurs homologues quadrupèdes ou hexapodes [06].



Asimo

Pino

**Figure I.7** Robot bipède

➤ **Les robots quadrapèdes :**

Imitent la morphologie animale et sont plus stables que les bipèdes.



**Figure I.8** Robot Quadrapède de MIT leg LAB

➤ **Les robots hexapodes :**

Les robots hexapodes sont les plus inspirés du monde des insectes. Par rapport aux bipèdes et aux quadrapèdes, les hexapodes sont plus stables alors qu'ils possèdent un grand nombre de degrés de liberté.

**I.5. Domaines d'application de la robotique :**

Le tableau ci-après résume de manière non exhaustive les diverses applications de la robotique.

Industrie nucléaire	- Surveillance de sites. - Manipulation de matériaux radioactifs. - Démantèlement de centrales.
Sécurité civile.	- Neutralisation d'activité terroriste.

	<ul style="list-style-type: none"> <li>- Déminage.</li> <li>- Pose d'explosif.</li> <li>- Surveillance de munitions.</li> </ul>
Militaire.	<ul style="list-style-type: none"> <li>- Surveillance, patrouille</li> <li>- Pose d'explosif.</li> <li>- Manipulation de munitions.</li> </ul>
Chimique.	<ul style="list-style-type: none"> <li>- Surveillance de site.</li> <li>- Manipulation de matériaux toxiques</li> </ul>
Médecine.	<ul style="list-style-type: none"> <li>- Assistance d'urgence.</li> <li>- Aide aux handicapés physiques, aux aveugles.</li> </ul>
Lutte contre l'incendie.	<ul style="list-style-type: none"> <li>- Localisation d'une source d'incendie.</li> <li>- Détection de fumée.</li> <li>- Suppression de flammes.</li> </ul>
Sous-marine.	<ul style="list-style-type: none"> <li>- Pose de câble.</li> <li>- Recherche de nodules.</li> <li>- Recherche de navires immergés.</li> <li>- Inspection des fonds marins.</li> </ul>
Agricole.	<ul style="list-style-type: none"> <li>- Cueillette de fruits.</li> <li>- Traite, moisson, traitement de vignes.....</li> </ul>
Construction BTP.	<ul style="list-style-type: none"> <li>- Projection mortier.</li> <li>- Lissage du béton.</li> </ul>
Nettoyage.	<ul style="list-style-type: none"> <li>- Coque de navire.</li> <li>- Nettoyage industriel.</li> </ul>
Espace.	<ul style="list-style-type: none"> <li>- Exploration.</li> </ul>
Industrie.	<ul style="list-style-type: none"> <li>- Convoyage.</li> <li>- Surveillance.</li> </ul>

**Tableaux I.2:** Domaines d'application de la robotique.

## Conclusion :

La robotique mobile connue une grande révolution. Elle est devenue indispensable et ces applications sont réparties dans presque dans tous les domaines de la vie. La robotique mobile va bien sûr continuer à intégrer les progrès de l'informatique, et en premier lieu bénéficier d'algorithmes de plus en plus complexes, exécutables en temps réel. Mais aussi, via les techniques d'accès aux réseaux Internet par mobiles. Les robots mobiles vont aussi se bénéficier du développement de la mécanique, l'électronique. Il va être composé de capteurs intelligents, micro contrôleurs, ce sont des facteurs de fiabilité et de baisse de coût.

Dans ce premier chapitre, nous avons donné une idée générale sur la robotique. Puis nous sommes passée à l'étude des différents types des robots, en tenant compte de l'explication détaillée de chaque type, à travers tous les développements de la robotique. Nous nous sommes basés sur l'études des robots mobiles à roues nettoyeurs des sols.

# Chapitre II

## **ROBOTS MOBILES**

## II. Introduction

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, spatiale, lutte contre l'incendie, surveillance...) [9].

L'objectif de ce chapitre est de donner un bref exposé sur le domaine de la robotique mobile, en particulier la navigation autonome des robots mobiles à roues et présenter leur autonomie.

### II.2. Définition d'un robot mobile

Il existe diverses définitions du terme robot mobile [10] :

1. Un robot mobile est d'abord une base mobile par opposition aux robots manipulateurs. A ce titre, il peut être utilisé pour le transport, c'est une machine automatique capable de se mouvoir dans un environnement donné. On regroupe sous cette appellation tous les robots autonomes (non Télécommandés) capables de se déplacer, par opposition aux robots attachés à un point fixe.
2. Un robot mobile est un manipulateur reprogrammable multifonctionnel conçu pour déplacer des matériaux, des outils, des pièces ou des composants spécialisés à travers une série de mouvements programmés pour effectuer une tâche précise.
3. Un robot est une machine équipée de capacités de perception, de décisions et d'actions qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception assurée par leurs capteurs.

### II.3. Grandes classes des robots mobiles

Selon l'Association Japonaise de Robots Industriels (AJRI), les robots sont répartis dans les classes suivantes :

- ❖ **Classe 1** : Robot de Manipulation Manuelle : un robot à plusieurs degrés de liberté actionné par l'opérateur.
- ❖ **Classe 2** : Robot à Séquence Fixe : robot de manipulation qui effectue les étapes successives d'une tâche en fonction d'une méthode prédéterminée, qui est difficile à modifier.
- ❖ **Classe 3** : Robot à Séquence Variable : le même type de robot de manipulation que dans la classe 2, mais les étapes peuvent être modifiées facilement.
- ❖ **Classe 4** : Robot Playback : l'opérateur humain effectue la tâche manuellement en conduisant ou en commandant le robot, qui enregistre les trajectoires. Cette information est rappelée lorsque cela est nécessaire, et le robot peut effectuer la tâche en mode automatique.

- ❖ **Classe 5** : Robot à Commande Numérique : l'opérateur humain alimente le robot avec un programme de mouvement plutôt que de lui enseigner la tâche manuellement.
- ❖ **Classe 6** : Robot Intelligent : un robot avec des moyens de comprendre son environnement, et la capacité de mener à bien une tâche malgré les changements dans les conditions ambiantes dans lesquelles celui-ci doit exécuter sa tâche.[11]

## II.4. Mode de fonctionnement

Il existe deux principaux modes de fonctionnement pour un robot mobile : télé-opéré et autonome. En mode télé-opéré, une personne pilote le robot à distance. Elle donne ses ordres via une interface de commande (joystick, clavier/souris...), et ceux-ci sont envoyés au robot via un lien de communication (internet, satellite ...). D'ailleurs, suivant le niveau de télé opération, le terme « robotique » est plus ou moins justifié. Le robot doit donc obéir aux ordres de l'opérateur qui perçoit l'environnement autour du robot, par différents moyens (retour d'image), de manière à donner des ordres adaptés au robot. Dans ce domaine, les efforts de recherche sont beaucoup portés sur les problèmes liés au réseau de télécommunication (retards dans le réseau de communication, problèmes de commande, pertes de données) et sur l'amélioration de la perception de l'environnement par l'opérateur (retours d'efforts).

A l'inverse, en mode autonome le robot doit prendre ses propres décisions. Cela signifie qu'il doit être capable à la fois de percevoir correctement son environnement, mais également de savoir comment réagir en conséquence, suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements il va atteindre son objectif. Les recherches dans ce domaine portent principalement d'une part sur la localisation du véhicule autonome et la cartographie de son environnement, d'autre part sur le contrôle de tels véhicules (structure de contrôle, stratégies de commande, planification).

Cette notion d'autonomie, que nous pourrions qualifier de décisionnelle, ne doit pas être confondue avec celle d'autonomie énergétique (capacité du robot à gérer efficacement son énergie, à la préserver, voire à se ravitailler), même si ces deux notions sont étroitement liées : idéalement une des préoccupations principales d'un robot mobile totalement autonome (du point de vue décisionnel), serait en effet de pouvoir gérer de lui-même ses réserves d'énergie [12].

## II.5. Autonomie d'un robot mobile

Le robot mobile est un agent physique réalisant des tâches dans son environnement, doté de capacités de perception, de décision et d'action. L'objectif est de permettre au robot d'interagir rationnellement avec son environnement automatiquement (sans intervention humaine) [13]. Cette nouvelle machine est caractérisée par sa capacité à être programmée pour réaliser des tâches très

diverses. Mettant en œuvre en particulier un ensemble de capteurs et un ensemble d'actionneurs [14]. Ses capacités en matière de manipulation d'objets lui ont permis de s'intégrer dans des lignes de production industrielle, où elle se substitue à l'homme dans les tâches difficiles, répétitives ou à risque pour l'être humain.

La figure II.1 présente l'interaction du robot avec son environnement. La manière dont le robot mobile gère ces différents éléments est définie par son architecture de contrôle, qui peut éventuellement faire appel à un modèle interne de l'environnement ou une stratégie intelligente pour lui permettre de planifier ses actions à long terme.

Nous considérons qu'un système est autonome si [15] :

- Il est capable d'accomplir sans intervention humaine les objectifs pour lesquels il a été conçu,
- Il est capable de choisir ses actions afin d'accomplir ses missions.

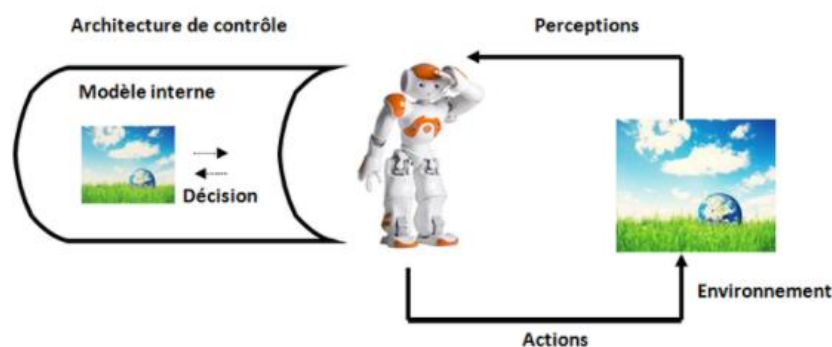
L'activité d'un tel robot se ramène aux tâches suivantes comme illustrée sur la figure II.2 :

[16]

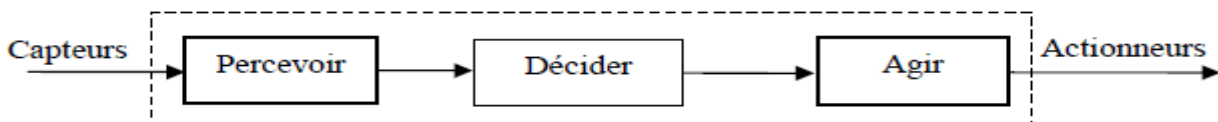
- **Perception** : le robot doit acquérir des informations sur l'environnement dans lequel il évolue par l'intermédiaire de ses capteurs. Ces informations permettent de mettre à jour un modèle de l'environnement (architectures hiérarchiques ou délibératives) ou peuvent être directement utilisées comme entrées de comportement de bas niveau (architecture purement réactive) ;

- **Décision** : le robot doit définir des séquences d'actions résultant d'un raisonnement appliqué sur un modèle de l'environnement ou répondant de manière réflexe à des stimuli étroitement liés aux capteurs

- **Action** : il doit exécuter les séquences d'actions élaborées en envoyant des consignes aux actionneurs par l'intermédiaire des boucles d'asservissements.



**Figure II .1** : Interaction entre le robot et l'environnement.



**Figure II.2 :** Etapes de traitement automatique de robot mobile.

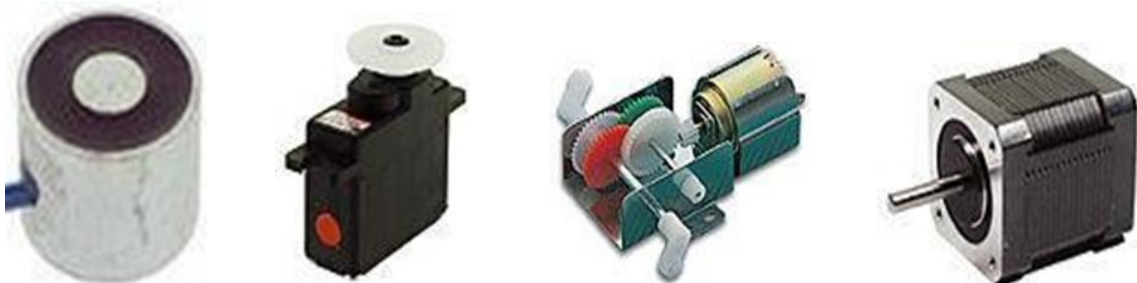
### II.5.1. Contrôle :

C'est une opération qui consiste à asservir les variables relatives au mouvement (qu'on notera  $x$ ) à des valeurs désirées (notées  $x_d$ ) ; par ce moyen, il est fait en sorte que la valeur  $x$  soit commandée à partir de la valeur désirée [17] et que l'on ait en fin de compte  $x = x_d$ . En d'autres termes, le contrôleur récupère l'information sensorielle (par des capteurs), prend des décisions intelligentes par rapport aux actes à accomplir, et effectue ces opérations en envoyant les commandes adéquates aux actionneurs.

### II.5.2. Actionneurs :

Ce sont des mécanismes qui permettent à /aux effecteur(s) d'exécuter une action, de convertir les commandes logicielles (Software) en des mouvements physiques ; leur but primaire est de produire assez de force pour provoquer le mouvement du robot, celle-ci représente la transformation d'une énergie source (énergie électrique) en énergie mécanique [18].

La Technologie des actionneurs est étroitement liée à l'énergie de base utilisée (pneumatique, hydraulique, électrique) [19]. Il existe beaucoup d'actionneurs principalement réalisés à base de moteur électrique, ces derniers permettent à votre robot de réagir suivant ce que vous lui avez demandé (Figure II.3).



**Figure II.3:** Différents types d'actionneurs électroniques.

### II.5.3. Effecteurs :

Ce sont tous les mécanismes à travers lesquels le robot peut effectuer des changements propres à lui, ou relatifs à l'environnement ; ces changements se font grâce aux actionneurs.

### II.5.4. Capteurs :

Ce sont des outils de perception qui permettent de gérer les relations entre le robot et son environnement. Il existe deux types de capteurs tels que : les capteurs proprioceptifs qui mesurent l'état mécanique interne du robot (comme les capteurs de position, de vitesse ou d'accélération), et les capteurs extéroceptifs qui recueillent des informations sur l'environnement (comme la détection de présence, mesure de distance... etc). Les capteurs ont comme fonction de lire les variables relativement au mouvement du robot pour permettre un contrôle convenable [18]. Il existe de nombreux types de capteurs qui nous permettent à notre robot d'avoir des informations sur l'extérieur (Figure II.4).



Figure II.4: Différents types des capteurs

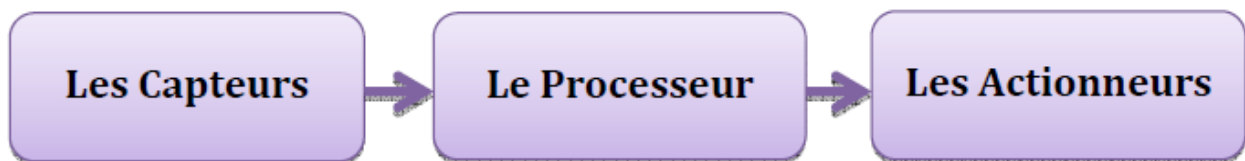
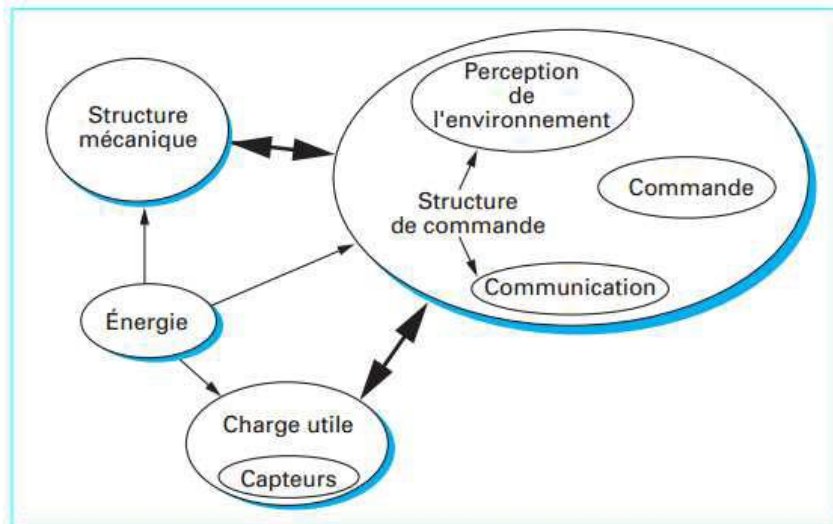


Figure II.5: Schéma synoptique de principe d'un robot

## II.6. Constitution d'un robot mobile autonome :

Un robot mobile autonome se décompose en trois sous-systèmes comme le montre la figure II.5[20]

- La structure mécanique.
- La charge utile.
- La structure de commande.



**Figure II.6:** Décomposition d'un robot mobile autonome.

### II.6.1. Structure mécanique

Elle rassemble les éléments mécaniques et électriques permettant d'assurer la fonction de mobilité. Le mouvement est obtenu en associant la propulsion et l'appui sur le milieu. La propulsion, très liée à l'application du robot, est assurée par des techniques variées (électrique, thermique, pneumatique ou hydraulique) en fonction de sa taille, sa masse et de son lieu d'évolution. Le milieu dans lequel évolue le mobile détermine en général le mode d'appui.

On dénombre deux grands types de milieux naturels : le milieu solide et le milieu fluide. La majorité des applications des robots mobiles autonomes relève du premier cas. Le second, plus restreint, concerne les robots sous-marins ou robots volants. Nous nous intéressons plutôt au premier type qui ouvre des voies d'applications plus nombreuses. Trois mécanismes de locomotion sont utilisés sur sol solide :

- Mobiles à roues
- Mobiles à chenilles
- Mobiles à pattes
- La motricité et l'énergie

Les déplacements des robots sont réalisés par des moteurs de types électrique, thermique ou hydraulique.

L'énergie électrique la plus fréquemment employée offre l'avantage d'une commande aisée.

Par contre le transport et la génération présentent des difficultés. Plusieurs méthodes sont employées :

- Par batteries qui sont soit rechargées périodiquement de manière automatique ou manuelle, soit par un échange avec d'autres lorsqu'elles sont déchargées.
- Par groupe électrogène embarqué dont l'inconvénient constitue la masse élevée. L'énergie de base est alors thermique.
- Par cordon ombilical qui réduit l'autonomie du robot.

L'énergie thermique est essentiellement employée par des véhicules de forte puissance comme énergie de base pour la traction ou pour activer un compresseur hydraulique.

### **II.6.2. Charge utile**

La charge utile est la charge que peut porter le robot, elle est nettement inférieure à la charge maximale que peut porter le robot qui est directement dépendante des actionneurs.

La mobilité du robot autonome n'est pas une fin en soi. Le déplacement est dicté par une action à réaliser sur l'environnement. La charge utile concerne directement l'application du robot. Il s'agit en général de véhiculer soit un outil (ex : brosse pour le robot nettoyeur) , soit un instrument de mesure (ex : une caméra pour le robot d'inspection) ou un objet quelconque (ex : conteneur pour le robot de transport).

### **II.6.3. Structure de commande**

La structure de commande repose sur trois modules fonctionnant de manière indépendante entre eux. Il s'agit des modules :

- De perception de l'environnement ;
- De communication homme-machine ;
- De commande.

#### **II.6.3.1. Perception de l'environnement**

La capacité d'autonomie d'un robot mobile est liée à sa faculté de percevoir et d'interpréter son environnement. Les variations d'états aléatoires de l'espace d'évolution demandent au robot une constante adaptation et des réactions différentes de celles prévues lors de la planification du mouvement. Le module de perception se décompose en plusieurs éléments : le transducteur, la modélisation, la détermination du type d'événement et la prise de décision. L'acquisition d'un état de l'environnement s'effectue soit par la mesure d'une énergie propre issue de l'espace d'évolution, soit par la réflexion d'énergie émise par une source artificielle. La donnée à traiter est contenue dans les caractéristiques de l'énergie réfléchie.

### II.6.3.2. Communication homme-machine

La structure de commande intègre les aspects de communication entre le robot et l'ordonnateur de tâches. La communication est réalisée à l'aide de multiples supports : écrit, visuel, sonore. En fait, ce module de communication paraît prendre de plus en plus d'importance à l'heure actuelle. Cela semble paradoxal avec le principe d'autonomie de notre robot. En fait, l'idée du robot mobile tout autonome semble perdre du terrain. La majorité des développements en cours, industriels ou dans les laboratoires s'orientent vers des fonctions autonomes des robots permettant de faciliter la tâche d'un opérateur. Pour cette raison, la communication avec l'homme est un maillon essentiel. Des interfaces de plus en plus conviviales sont développées.

### II.6.3.3. Commande

La planification des actions et le contrôle des déplacements se situent dans trois référentiels différents. Du plus abstrait au plus concret, nous les avons repérés : monde idéal, monde perçu et monde réel. Cette décomposition est classique pour les systèmes automatiques de commande. L'élément nouveau, s'agissant des robots mobiles, est que le monde idéal a un champ très étendu, car se référant, éventuellement, non seulement à des représentations de type intégré-différentiel, mais aussi à des représentations et classifications de type sémantique (par exemple des notions de corridor, de pièces, etc.). Ainsi, le monde idéal dans lequel évolue le robot sera l'objet d'au moins trois modélisations, correspondant à des niveaux de planification différents. Il s'agit de la modélisation sémantique, de la modélisation topologique et de la modélisation géométrique.

## II.7. Navigation autonome des robots mobiles

La navigation d'un robot mobile est une tâche qui consiste, généralement, à trouver un mouvement libre dans l'espace de configuration (environnement de travail) sans collision avec les obstacles proche du robot. L'espace de configuration est l'ensemble des paramètres caractérisant la position du robot dans son environnement (position et orientation).

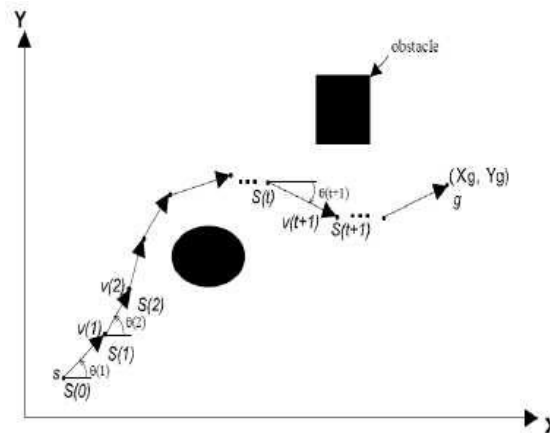
Ce mouvement amène le robot d'une configuration initiale  $q_0 = q(t_0) = (x_0, y_0, q_0)$  à une autre finale désirée  $q_f = q(t_f) = (x_f, y_f, q_f)$  [21] [22] [23].

La figure II.4 présente une description de la tâche de navigation d'un robot mobile.

Comme montré, le robot démarre d'une situation initiale  $s$ , il doit exécuter les actions de mouvement, qui sont généralement la vitesse et l'angle de braquage ( $v(t)$  et  $a(t)$ ) lui permettant de se mouvoir vers une nouvelle situation  $s(t+1)$ . La navigation est obtenue à travers un processus itératif comme suit :

1. À chaque instant  $t$ , avec  $t = 0, 1, \dots, k, \dots$ , le robot doit mesurer les distances aux obstacles de l'environnement  $d_i$  et les positions: courante et finale:  $P_g(x_g, y_g)$  ;

2. Le système de contrôle détermine les variables de commande adéquate  $v_r(t+1)$  et  $a(t+1)$ ,
3. Le robot exécute ses actions en déplaçant vers les nouvelles coordonnées,
4. Répéter le même processus (les étapes : 1, 2 et 3) de détection de la situation et génération des actions jusqu'à la destination finale appelée but.



**Figure II.7:** Processus de navigation autonome.

On peut donner des définitions différentes de la tâche de navigation selon le but recherché. Il existe néanmoins une très grande variété de travaux et des méthodes permettant d'aborder ce problème difficile. Les stratégies de navigation permettant à un robot mobile de se déplacer pour rejoindre un but sont extrêmement diverses. Pour exécuter une tâche de navigation autonome, un robot mobile doit mettre en œuvre un certain nombre de fonctionnalités, parmi lesquelles [24] :

### II.7.1 Planification de mouvement

L'un des principaux objectifs du robot mobile est de pouvoir évoluer dans un environnement complexe encombré d'obstacles pour atteindre son but final. Il a besoin de construire une trajectoire définie comme une séquence de déplacement sans collision avec ces obstacles entre la position initiale (point de démarrage) et le point but ou cible (figure I.8). La planification de trajectoire dans sa formulation classique est le problème du calcul de ce chemin, dans un modèle géométrique de l'environnement cela est fait en introduisant le concept d'espace des configurations qui permet de transformer le problème de la recherche d'un chemin pour un système à  $n$  degrés de liberté dans l'espace euclidien en celui du mouvement d'un point dans un espace à  $n$  dimensions où le robot est représenté par un point. Plusieurs approches sont proposées pour la planification de trajectoire. Cependant, les plus utilisées sont la planification globale et locale.

### II.7.2 Localisation

Afin d'exécuter le mouvement planifié, le robot doit se localiser dans son environnement en estimant la position et l'orientation par rapport à un repère fixe. L'estimation peut s'effectuer soit par une mesure des déplacements du robot, soit par une mesure de sa position absolue dans

l'environnement. Plusieurs approches de localisation sont utilisées en robotique mobile. Elles peuvent être classées en trois catégories : localisation basée sur une carte, localisation par rapport à des balises et localisation par rapport à d'autres robots.

Du fait des incertitudes sur les mesures utiles pour la localisation, elle est généralement modélisée dans un cadre probabiliste.

### **II.7.3 Suivi de trajectoire**

Le suivi de trajectoire est une mission importante pour un robot mobile. Il consiste à calculer les commandes envoyées aux actionneurs permettant de réaliser le mouvement planifié. Un robot est un système dynamique commandé par bouclage pour la poursuite de sa trajectoire de référence. On s'intéresse dans ce travail à l'étude de cette fonctionnalité.

### **II.7.4 Évitement d'obstacles**

Le suivi de la trajectoire planifiée ne permet pas de garantir l'absence des collisions avec les objets statiques ou dynamique existants. L'évitement d'obstacles est un comportement de base présent dans quasiment tous les mouvements des robots mobiles. Ces collisions peuvent se produire lors de l'exécution de la trajectoire, dues à une localisation imparfaite, un plan imprécis ou des obstacles qui n'étaient pas dans le modèle de l'environnement utilisé pour la planification de trajectoire. Le robot mobile autonome doit avoir une capacité d'évitement d'obstacles efficace.

### **II.7.5 Parking**

Le parking est la phase finale de la navigation autonome, car l'objectif d'une mission de navigation est souvent d'atteindre une configuration finale spécifiée avec une grande précision, Le succès de navigation dépend de la réalisation de cet objectif de stationnement.

Tous ces éléments ou tâches font que le mouvement initialement planifié doit être adapté lors de son exécution et que des stratégies d'évitement réactives d'obstacles doivent être mises en œuvre. On adoptera des stratégies différentes en fonction du type de système, de sa vitesse, et de champ d'application. Après cet aperçu sur les éléments constitutifs et nécessaires pour faire naviguer un robot mobile dans son environnement. Il s'agit maintenant d'utiliser au mieux la motricité du robot et sa localisation pour accomplir la tâche de navigation autonome ; le robot doit être doté d'un système de contrôle adéquat. Pour cela, dans ce qui suit, on présente quelques types des systèmes de contrôle existant [24].

## II.8 Architectures de contrôle des robots mobiles

Un robot est un système complexe qui doit satisfaire des exigences variées en utilisant un ensemble logiciel appelé architecture de contrôle du robot. Cette architecture permet donc d'organiser les relations entre les trois grandes fonctions : la perception, la décision (planification) et l'action. Ces architectures peuvent néanmoins être classées en trois groupes (approches).[24]

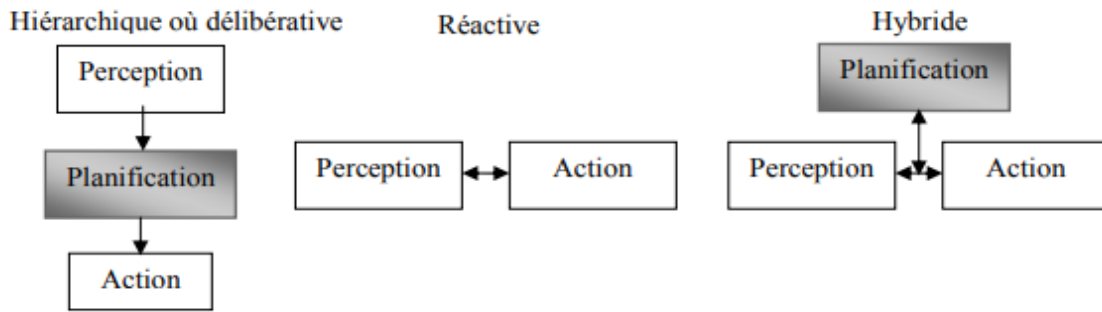


Figure II.8: Architectures de contrôle pour les robots mobile

### II.8.1 Approche délibérative (penser puis agir)

Les premiers robots mobiles dérivés des recherches en intelligence artificielle utilisaient des contrôleurs hiérarchiques, dont le fonctionnement repose essentiellement sur la disponibilité d'un modèle de l'environnement permettant de représenter toutes les informations pertinentes pour le déplacement du robot. Elles ont l'avantage de prouver l'existence d'une solution optimale permettant au robot d'atteindre le but assigné. Mais un tel modèle peut être insuffisant dans un environnement inconnu ou dynamique car au moment de la réalisation de l'action, l'environnement peut avoir changé et la décision n'est plus valide. Le traitement est décomposé en une série d'opérations successives comme décrit sur la figure I.9.

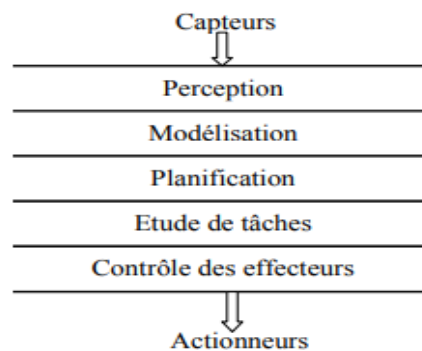


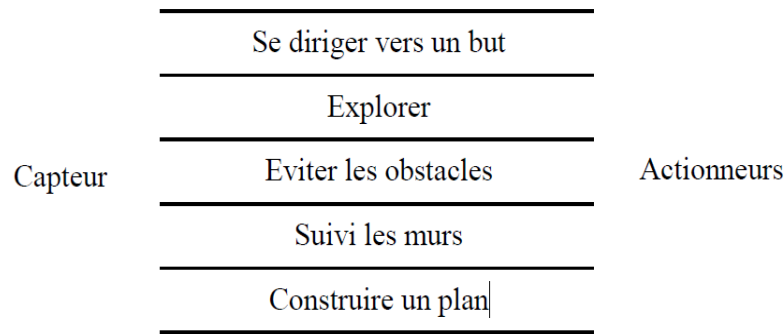
Figure II.9 : Architecture Hiérarchique

Dans ce type d'architecture, le robot utilise toutes les informations sensorielles disponibles et toutes les connaissances internes sauvegardées et raisonne sur les prochaines actions à réaliser. Le robot doit construire et ensuite évaluer tous les plans possibles pour trouver celui qui atteint le but. La planification requiert l'existence d'une représentation interne du monde qui permet au robot d'avoir une idée de futur et de prédire les résultats d'actions possibles dans les différents états perçus. Le modèle interne doit donc être précis et récent. Quand le temps est suffisant afin de générer un modèle du monde précis, cette approche permet au robot d'agir stratégiquement en sélectionnant la meilleure action pour une situation donnée.

La partie la plus importante est celle qui concerne la modélisation et la planification qui peuvent prendre un temps assez long, et ceci notamment en environnement peu structuré. En cas d'insuffisance du modèle ou capacité de traitement, cette approche ne peut être appliquée. Les écarts entre le modèle et l'environnement ne peuvent être pris en compte que via un nouveau cycle "perception-modélisation-planification", ce qui est peu réactif et nécessite l'utilisation de nouvelles méthodologies assez puissantes. Ces méthodes doivent établir un couplage entre la perception et l'action et le traitement local des différentes tâches.

### **II.8.2 Approche réactive** (ne pas penser mais agir)

En 1986, Brooks a proposé une approche réactive qui se distingue par l'abandon des phases de *modélisation* et de *planification*. Le contrôle du robot se fait alors sans utiliser le modèle de l'environnement (figure II.10). Les stratégies de navigation réactives n'utilisent que les valeurs courantes des capteurs et non des données provenant d'un modèle interne, pour décider de l'action à effectuer. Le principe de l'approche réactive est basé sur la décomposition de la tâche de navigation en un ensemble de comportements actifs de base (explorer, aller au but, éviter les obstacles, suivi des murs,...). Pour guider le robot, il faut donc choisir à chaque instant lequel de ces comportements est à activer ? Ce problème est connu dans la littérature scientifique sous le nom de *sélection de l'action*. La solution proposée par Brooks est de diviser la tâche globale en un ensemble des sous tâches secondaires. Cette architecture est appelée "*subsumption*", utilise une hiérarchie des comportements qui se déclenchent donc selon un ordre de priorité en fonction des données de perception (fusion des comportements).

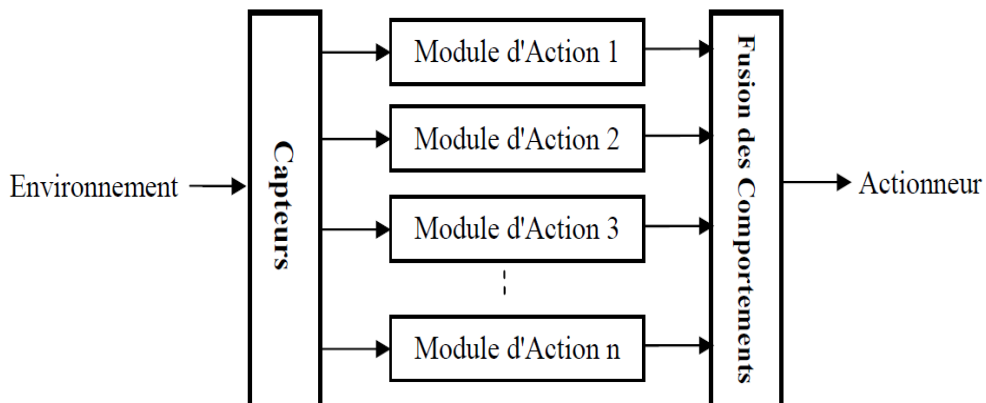


**Figure II.10** Architecture réactive

La figure II.10 décrit les différentes couches comportementales d'un robot en utilisant la structure subsumption. Cette architecture peut être définie comme une hiérarchie de comportements plutôt que comme une hiérarchie fondée sur une abstraction des données. Le concept de base est le suivant : si complexe soit-il, peut être décomposé en comportements élémentaires représentant chacun un niveau de compétence. Ces différents niveaux sont placés en couches parallèles et chaque compétence couple directement la perception avec l'action. Une telle architecture a, au moins, trois avantages :

- D'abord, elle peut réagir aux éventualités en temps réel dû au parallélisme,
- Chaque comportement définit une tâche indiquée ce qui facilitera leur contrôle et modification, par exemple : ajouter ou enlever un comportement,
- Elle présente une bonne robustesse ; le système peut fonctionner même si un ou plusieurs comportements échouent. Elle apporte donc l'avantage d'avoir une trajectoire qui s'adapte aux changements de l'environnement.

Cette architecture est une approche purement réactive vient de son manque de capacités de raisonnement de haut niveau et de modularité.



**Figure I.11** Architecture modulaire de Subsumption

**II.8.3 Approche hybride** (penser et agir indépendamment en parallèle)

Cette architecture combine les meilleurs aspects des architectures réactives et délibératives : elle combine la réponse temps-réel de la réactivité avec la rationalité et l'efficacité de la délibération. En effet, la composante réactive s'occupe des besoins immédiats du robot comme l'évitement d'obstacles et fonctionne ainsi en un temps court et utilise des données et signaux externes directs, alors que la composante délibérative utilise des représentations internes, abstraites et symboliques du monde et fonctionne en un temps très long. Tant que les sorties des deux composantes ne sont pas en conflit, le système n'a pas besoin de plus de coordination. Cependant, ces deux parties interagissent pour tirer les avantages de chacune. La composante réactive doit outrepasser la composante délibérative si le monde présente un certains déficit immédiat et imprévisible et la partie délibérative doit informer la partie réactive afin de guider le robot vers des trajectoires et des buts plus efficaces. L'interaction des deux parties de l'architecture requiert une composante intermédiaire dont sa construction est particulièrement le plus grand défi dans la conception hybride. Ainsi, les architectures hybrides sont souvent appelées "architectures à trois couches", composées des couches réactive, intermédiaire et délibérative. Un grand effort de recherche a été conduit dans la manière de concevoir ces composantes et leurs interactions.[24]

**II.8.4 Approche collective**

En cas d'application qui nécessite l'utilisation de plusieurs robots mobiles (la robotique collective), l'architecture de contrôle comportementale collective tire son inspiration de la biologie et essaie de modéliser le comportement des animaux dans leurs environnements complexes d'évolution. Les composantes de base de ces architectures sont des comportements observés d'activité émergeant des interactions entre le robot et son environnement. De tels systèmes sont construits d'une façon ascendante commençant par un ensemble de comportements de survie tel que l'évitement de collisions qui couple les entrées perçues aux actions du robot. Des comportements sont ajoutés pour fournir au robot plus de capacités lui permettant de réaliser des tâches de plus en plus complexes telle que le suivi de mur, la poursuite d'une cible, l'exploration ou le "homing" (rentrez chez-soi). Ces nouveaux comportements sont introduits dans le système incrémentale des plus simples aux plus compliqués jusqu'à ce que leurs interactions aboutissent aux capacités désirées du robot. Comme les architectures hybrides, les architectures comportementales peuvent être organisées en couches mais à l'inverse de ces architectures, les couches ne diffèrent pas beaucoup les unes des autres en termes de temps et représentation utilisées. Toutes les couches sont représentées par des comportements, des processus qui récoltent des entrées et s'envoient des sorties.[24]

## II.9 Conclusion

La robotique est la branche de l'intelligence artificielle concernée par l'étude des systèmes automatiques capables d'interagir directement avec le monde physique. C'est une automatisation de ses machines, ou l'objectif est d'augmenter les capacités de localisation et de navigation dans son espace de travail. Ce chapitre nous a permis de présenter un exposé sur le domaine de la robotique mobile, en définissant la navigation autonome des robots mobiles, les types des robots et les capteurs utilisés. On a présenté aussi les principales architectures utilisées pour le contrôle des robots mobiles en citant les avantages et les inconvénients de chacune. Les robots mobiles à roues ont les robots mobiles les plus répandus, à cause de ses structures mécanique simple et ses commandes relativement plus faciles que les autres robots qui diffèrent par leur moyen de locomotion. La commande d'un robot mobile se divise généralement en trois étapes principales : perception, décision et action. La dernière étape concerne l'exécution des mouvements planifiés, c'est une étape qu'il doit maîtriser efficacement pour accomplir ses missions avec succès.

Après cet aperçu, dans le chapitre suivant nous présenterons la conception de robot en citant tout le matériel utilisé ainsi les logiciels de programmation et de schématisation.

Chapitre III

**Etude et Conception**

**Du Robot**

### III.1 Introduction

La conception est une étape cruciale dans notre travail qui consiste à la création de notre prototype robot nettoyeur des sols à base d'Arduino via un smartphone qui se déplace sur un terrain plat. Ainsi, nous avons réfléchi aux fonctionnalités à mettre en place de façon raisonnable. La fonction première envisagée est d'offrir à ce robot une autonomie dans ses déplacements, en évitant les obstacles qui peuvent être rencontrés sur son chemin, et ce grâce à un système embarqué sur batteries. La deuxième fonction ciblée est la commande à distance et sans file de robot.

Ce chapitre est consacré à la présentation de l'architecture et de la conception du projet en question, la conception est décomposée en trois parties essentielles (Electronique, Mécanique et Logiciel) et avant de présenter ces trois types de conception, on doit commencer par un cahier de charge et ces différentes fonctions, par la suite on va présenter les composants matériels et logiciels.

### III.2 Cahier de charge :

Nous avons fixé notre propre cahier des charges en proposant ce sujet. L'idée était donc de créer notre propre robot mobile et nettoyeur idéalement commandé à distance. Nous avons donc pu distinguer les différentes fonctions de notre cahier des charges comme suit :

- Fonction principale : construire un système qui simplifie la vie des gens.
- Fonction secondaire 1 : Être contrôlable et pilotable à distance.
- Fonction secondaire 2 : jet d'eau, élimine les saletés.
- Fonction secondaire 3 : Disposer d'une autonomie dans ces déplacements.

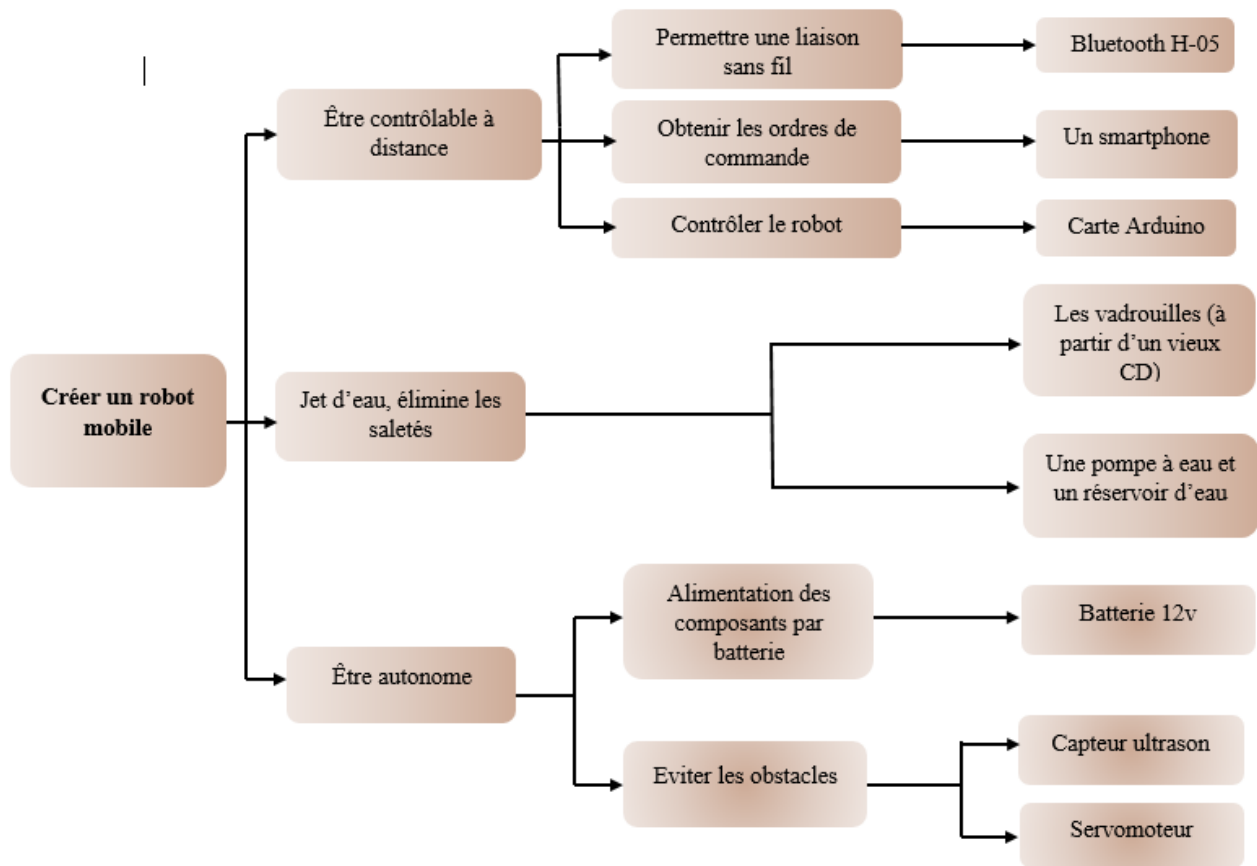


Figure III.1: Schéma bloc détaillés de cahier des charges.

### III.3 Les structures de robot mobile :

On peut distinguer les différentes parties fonctionnelles de robot :

- Structure mécanique
- Structure électronique
- Logiciel

#### III.3.1 Structure mécanique :

##### III.3.1.1 le Châssis :

Un châssis est une plaque sur laquelle sont posés tous les composants du robot et permettant de le faire tenir en équilibre. Parmi les différents matériaux que l'on peut utiliser pour réaliser le châssis d'un robot, le bois est le plus solide. Cette matière présente de nombreux avantages, elle existe en plaque, carré ou rond, ou sous d'autres formes.

Dans notre cas, on a choisi un châssis constitué d'une plaque en bois de 27 cm de diamètre, et d'une épaisseur 7cm.

### III.3.1.2. Moyens de locomotion

Pour une base mobile, la locomotion est très importante, dans le cadre de sa stabilité. Les principes constructifs pour déplacer un robot mobile sont nombreux et variés. On peut distinguer comme grandes catégories les roues et les chenilles [14],[15].

#### III.3.1.2.1. 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 rencontre principalement trois types de roues :

- Les roues fixes dont l'axe de rotation, passe par le centre de la roue ;
- Les roues centrées orientables, l'axe d'orientation est perpendiculaire au sol, il passe par le centre de la roue ;
- Les roues décentrées orientables, souvent appelées roues folles, pour lesquelles l'axe d'orientation est perpendiculaire au sol, il ne passe pas par le centre de la roue.

Nous avons opté pour les roues fixes, les principales raisons de ce choix technologique sont les suivantes : les roues sont plus faciles à contrôler, elles dissipent moins d'énergie et elles permettent au véhicule de se déplacer plus rapidement. Pour donner plus d'équilibre au robot on va utiliser quatre roues motrices présentées par la figure III.2.



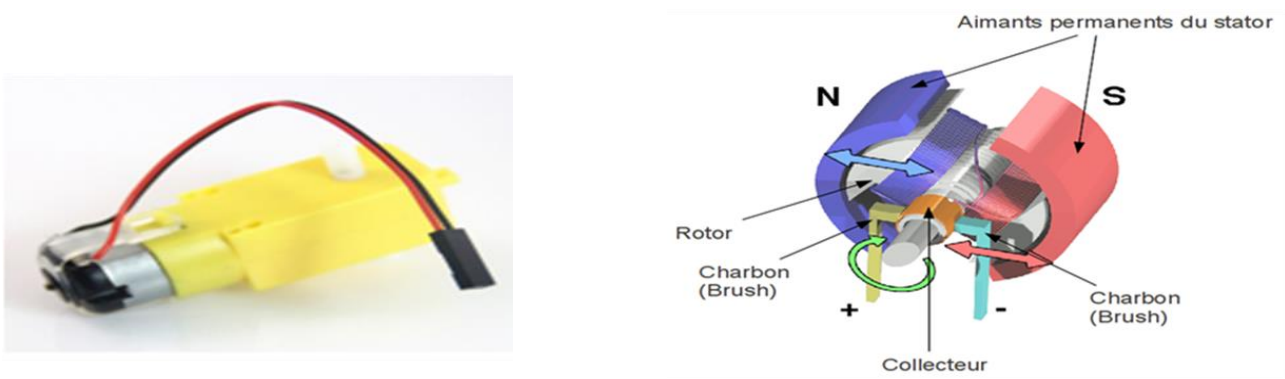
**Figure III.2:** Les Roues de robot.

Pour mettre les roues en mouvement on a besoin d'un système de rotation qui est le moteur, Chaque roue sera entraînée par un moteur.

### III.3.1.3 Moteur à courant continu (MCC) :

Le moteur à courant continu est composé de deux parties principales : le rotor (partie qui tourne) et le stator (partie fixe). En électrotechnique le stator s'appelle aussi inducteur et le rotor s'appelle l'induit. Sur l'image au-dessous, nous pouvons observer au milieu – entouré par les aimants bleu et rouge qui constituent le stator – le rotor composé de fils de cuivre enroulés sur un support lui-même monté sur un axe. Cet axe, c'est l'arbre de sortie du moteur.

C'est lui qui va transmettre le mouvement à l'ensemble mécanique (pignons, chaîne, actionneur...) qui lui est associé en aval



**Figure III.3:** Moteur à courant continu.

#### ➤ **Fonctionnement :**

Le stator est constitué d'un aimant permanent qui engendre un champ magnétique. Le rotor, ou induit, est alimenté par un courant continu. Le courant électrique traverse le rotor. Ses conducteurs sont immergés dans le champ. Soumis à la force  $F$  (Laplace), le rotor tourne et le couple du moteur est généré. Le courant transmis au collecteur par les balais traverse le conducteur rotorique et change le sens de commutation ce qui conserve la magnétisation du rotor perpendiculaire à celle du stator. Le couple est créé par l'interaction magnétique entre le stator et le rotor où le champ statorique est presque inexistant sur les conducteurs. La magnétisation transversale du rotor reste la même durant la rotation. Parce que le pôle statorique exerce une action sur le pôle rotorique, le moteur tourne. On peut calculer facilement le couple en utilisant la force de Laplace créée par le champ statorique. Cette force exerce une action sur les conducteurs rotoriques traversés par le courant d'intensité. Elle est issue de l'action réciproque et est semblable en module pour deux conducteurs rotoriques diamétralement opposés. Cependant, les sens des courants étant inverses, les forces sont également

opposées. La force créée est alors fonction de l'intensité et des champs magnétiques comme le couple moteur est proportionnel à ces deux grandeurs.

### III.3.1.4 La pompe :

Une pompe est un dispositif permettant d'aspirer de refouler un liquide (les compresseurs véhiculent des gaz compressible). Les pompes sont électriques et diesel, il y a des différents types de pompe avec différent caractéristique on mention :

- **Pompe rotative centrifuge** : le mouvement du liquide résulte de l'accroissement d'énergie qui lui est communiqué par la force centrifuge.
- **Pompe volumique** : sont des pompes à débit constant sur leurs plages de fonctionnement
- **Pompe par Airflit** : parfois appelé « pompe par injection d'air », c'est un système de pompage simple d'un liquide par injection d'air comprimé dans une canalisation.
- **Pompe à vide** : c'est un outil permettant de faire le vide, c'est-à-dire d'extraire l'aire ou tout autres gaz contenu dans une enceinte close, afin d'en diminuer la pression.
- **Pompe péristaltique** : « pompe à galets » utilise un tube flexible dans lequel le liquide ou le gaz a entraîné un système de cames ou de galets pressant le tube à l'intérieur de la pompe.
- **Pompe à palettes** : ce type de pompe est surtout utilisé pour diminuer ou augmenter la pression des gaz : pompe à vide, compresseur d'air, climatiseur, réfrigérateur...etc.



**Figure III.4:** pompe électronique 12v.

### III.3.2 Structure électronique :

#### III.3.2.1 Arduino Uno :

Est une carte électronique développée par arduino.cc dont la pièce maîtresse est un microcontrôleur de type ATmega 328P-PU fabriqué par la société Atmel ATmega. Elle possède des pistes électriques disposées sur une, deux ou plusieurs couches (en surface et/ou en interne) qui permettent la mise en relation électrique des composants électroniques. Chaque piste relie tel composant à tel autre, de façon à créer un système électronique qui fonctionne et qui réalise les opérations demandées. Autour de lui, on trouvera 14 broches d'entrée/sortie numériques (dont 6 peuvent servir de sorties MLI, ou PWM), 6 entrées analogiques, un oscillateur à quartz de 16 MHz, 2 ko de SRAM et 32 ko de ROM, un connecteur d'alimentation en courant continu, nécessitant 7 à 12V, une embase ICSP, un port USB type B, un bouton d'initialisation (reset), une LED, et un régulateur de tension qui ramènera la tension de fonctionnement à 5V. Pour l'utiliser, il suffit de la relier à un ordinateur avec un câble USB, ou encore de l'alimenter à l'aide d'un bloc secteur externe ou de piles. Le site officiel Arduino est : <https://www.arduino.cc/>

➤ **Avantage de la carte Arduino Uno :**

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant aux personnes intéressées plusieurs avantages cités comme suit :

- **Le prix (Réduit) :** Les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes.
- **Multi plateforme :** Le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Un environnement de programmation clair et simple :** L'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible :** Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (Fonctionne sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui

peut transférer le programme à travers de la liaison série (RS232, Bluetooth ou USB selon le module).

- **Matériel Open source et extensible :** Les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA328, les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût.

➤ **Caractéristiques techniques :**

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limite)	6-20V
E / S numérique Pins	14(dont 6 fournissent la sortie PWM)
PWM numérique E / S Pins	6
Pins d'entrée analogique	6
DC Courant par I O pin	20mA
Courant DC pour 3,3V Pin	50mA
Mémoire flash	32 KB (ATmega328P) dont 0,5 KB utilisé par bootloader
SRAM	2KB (ATmega328P)
EEPROM	1KB (ATmega328P)
Vitesse de l'horloge	16MHz
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g

**Tableau III.1 :** Caractéristique de la carte Arduino.

➤ **Détails techniques :**

La carte Arduino Uno peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée. Une alimentation externe peut provenir soit d'un adaptateur AC-DC ou d'une batterie. L'adaptateur peut être connecté en branchant une prise

2.1mm dans la prise d'alimentation de la carte ou à partir d'une batterie connectée dans le pin (ou broche) GND et V-in (alimentation Externe).

Le processeur peut fonctionner sur une alimentation externe de 6 à 20 volts. Cependant, si la Tension est inférieure à 7V, le pin 5V peut fournir moins de cinq volts et le processeur peut devenir instable. Si la tension est supérieure à 12V, le régulateur de tension peut surchauffer et endommager la carte. La plage recommandée est de 7 à 12 volts. Les pins (ou broches) d'alimentation sont les suivantes :

- **Vin** : Tension d'entrée à la carte Arduino à l'aide d'une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou une autre source d'alimentation régulée). Si l'alimentation en tension est faite par l'intermédiaire de la prise d'alimentation, on pourra y accéder via ce pin.
- **5V** : Cette pin délivre un 5V régulé par la carte. Le processeur peut être alimenté soit à partir de la prise d'alimentation DC (7-12V), le connecteur USB (5V), ou le pin V-in de la carte (7-12). La fourniture d'une tension via les 5V ou 3,3V contourne le régulateur, et peut endommager votre processeur. A déconseiller !
- **3V3** : Une alimentation de 3,3 volts générée par le régulateur. La consommation de courant maximale est de 50 mA.
- GND masse.
- **IREF** : Ce pin sur la carte Arduino fournit la référence de tension avec laquelle le microcontrôleur fonctionne [26].

➤ **Le 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 (III.9) montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.



**Figure III.5:** Microcontrôleur ATmega 328.

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

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

➤ **Mémoire :**

L'ATmega328 a 32 Ko (avec 0,5 KB occupées par le bootloader\*\*). Il a également 2 Ko de SRAM et 1 Ko de mémoire EEPROM\*\*\* (qui peut être lu et écrit avec la bibliothèque de l'EEPROM).

➤ **Entrées et sortie :**

Chacune des 14 broches numériques sur la carte Uno peut être utilisée comme une entrée ou une sortie, en utilisant les fonctions PinMode, digitalWrite, et digitalRead. Ils fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandée et a une résistance de pull-up interne (déconnecté par défaut) de 20-50k ohm. Un maximum de 40mA est la valeur qui ne doit pas être dépassée sur toutes les broches d'Entrée/Sorties pour éviter des dommages permanents au microcontrôleur.

Certaines broches ont des fonctions spécialisées :

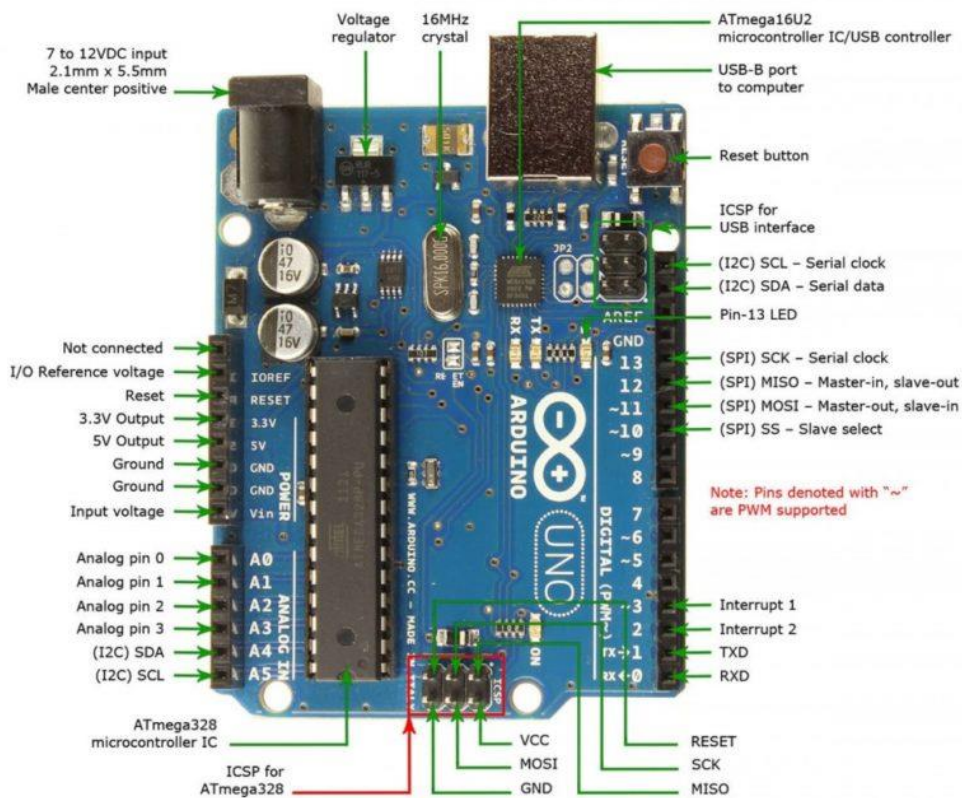
- **Série 0 (RX) et 1 (TX)** : Permet de recevoir (RX) et transmettre (TX) TTL données série. Ces pins sont connectés aux pins correspondants de l'USB-TTL puce Serial ATmega8U2.
- **LED 13** : Il est équipé d'un conduit par la broche numérique 13. LED Lorsque la broche est à la valeur HIGH, la LED est allumée, lorsque la broche est faible, il est hors tension.

L'Uno dispose de 6 entrées analogiques, A0 à A5, dont chacune fournit 10 bits de résolution (ou 1024 valeurs différentes). Par défaut, la tension est de 5 volts. Il est cependant possible de changer la limite supérieure de la gamme en utilisant la broche AREF et la fonction AnalogReference. Autres broches de la carte :

- **AREF** : Tension de référence pour les entrées analogiques. Pin utilisé avec AnalogReference.
- Réinitialiser.

➤ **Communication :**

Arduino a un certain nombre de moyens pour communiquer avec un ordinateur, une autre carte Arduino, ou autres microcontrôleurs. L'ATmega328 fournit UART TTL (5V) en communication série, disponible sur les broches numériques 0 (RX) et 1 (TX).



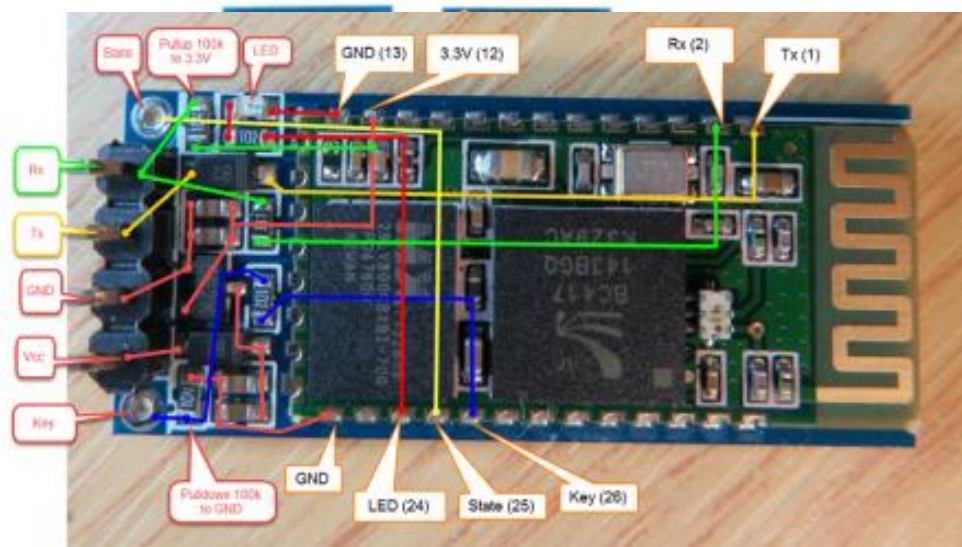
**Figure III. 6 :** Constitution de la carte Arduino Uno.

**III.3.2.2 Le module Bluetooth :**

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une Connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, Très faible portée, faible débit, très bon marché et peu encombrant. Il existe deux modules de Bluetooth le HC05 et HC06. Ils permettent simplement de disposer d'une liaison Bluetooth sur un projet.

Notre choix s'est porté sur le module HC-06, c'est un accessoire indispensable si l'on souhaite communiquer sans fil (avec par exemple un smartphone en Bluetooth) avec une carte Arduino.

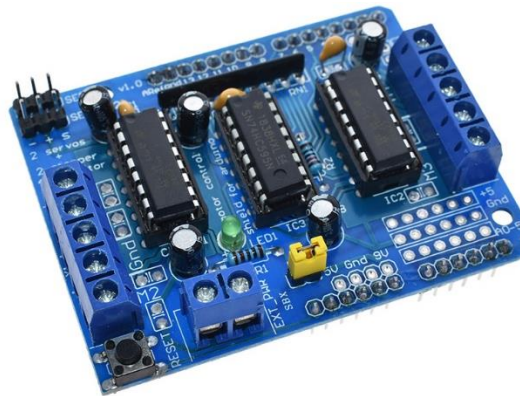
Le module HC06 est différent du module HC05, autre grand classique dans le monde merveilleux d'Arduino. C'est un module "esclave" contrairement au HC05 qui est un module "maître". On ne pourra donc pas faire communiquer deux Arduino en utilisant deux modules HC06 mais par contre, on peut très bien établir une liaison Bluetooth entre un smartphone et un Arduino équipé de ce module HC06. Il possède 6 pins 4 uniquement sont utilisées couramment, VCC et GND pour l'alimentation (5V), RX/TX pour la communication. On y voit aussi une broche « Key » (ou bien Enable « EN ») qui servira à envoyer des commandes de configuration au Module. La dernière broche nommée « Led » (ou bien state) permet de brancher une LED pour obtenir un signal sur l'état du module. Led sert à connecter une LED de statut, vous pouvez la laisser déconnectée cela n'influencera pas le comportement du module. Key sert à utiliser le mode « commande » du module. Avec cette dernière vous pourrez reconfigurer la voie série (vitesse, parité etc...) et d'autres options liées au Bluetooth. Cette broche est connectée à n'importe mode esclave/maître quelle sortie numérique de l'Arduino.



**Figure III.7:** Module Bluetooth HC-06.

### III.3.2.3 Driver Moteur L293D :

Le L293D est un circuit monolithique à 4 canaux intégré haut tension et courant élevé. En gros, cela signifie que grâce à cette puce, vous pouvez piloter des moteurs à courant continu fournisseur jusqu'à 36 volts, et la puce peut fournir un courant maximal de 600mA par canal. La puce L293D est également connue en tant que type de pont en H. Ce pont est typiquement un circuit électrique qui permet d'appliquer une tension aux bornes d'une charge dans les deux sens, donc faire varier la vitesse et le sens de rotation d'un moteur DC.



**Figure III.8:** Driver Moteur L293D.

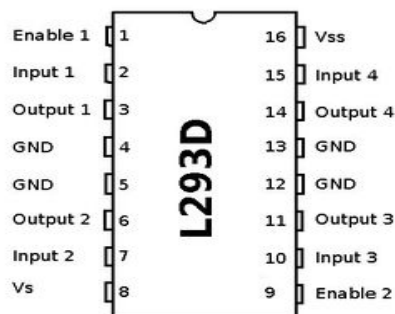
Cela signifie principalement que vous pouvez inverser la direction du courant et donc renverser le sens de rotation du moteur.

Le principe de fonctionnement est basé sur 4 éléments du circuit communément nommés contacts et identifiés comme suit : Haut Côté Gauche, Haut Côté Droit, Bas-Côté Droit et Bas-Côté Gauche.

En utilisant différentes combinaisons de fermetures, il est possible de démarrer, Stopper ou Inverser le courant. La seule chose à laquelle il faut faire attention dans tout cela, ce sont les 2 broches d'entrée (Input Pins) qui commande la logique pour chaque moteur. Le plus important pour nos besoins, c'est que ces entrées sont commandables depuis un Board Arduino.

Il n'est pas vraiment nécessaire de se préoccuper de la régulation de la tension parce que le L293D accepte deux sources de tensions.

- Une source d'alimentation directe (jusqu'a 36 Volts) pour alimenter les moteurs.
- Une autre source de tension (5 Volts) pour alimenter la logique du circuit intégré.
- Cette alimentation peut provenir directement de la carte Arduino.



**Figure III.9:** Pont-H L293D.

**VSS** : Alimentation de la logique de commande (5V).

A raccorder à la borne +5V d'Arduino (donc sur le régulateur d'Arduino).

**VS** : Alimentation de puissance des moteurs.

Par exemple, s'il s'agit d'un ancien véhicule téléguidé que vous avez cannibalisé, il s'agira de la borne positive (+9.2v) de votre accumulateur.

**GND** : Doit être raccordé à la masse (GND) de la source d'alimentation de puissance VS (donc le négatif de l'accumulateur) et à la masse de la source d'alimentation de VSS (donc GND Arduino).

Si vous n'avez qu'une source d'alimentation pour le tout, c'est forcément plus simple.

**OUTPUT1, OUTPUT2** : Broches à raccorder à la charge (le moteur).

**INPUT1, INPUT2** : Broche de commande du Pont-H. Se raccorde à Arduino.

**ENABLE1** : permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 & OUTPUT2.

ENABLE1 commande l'activation du premier Pont-H.

Si ENABLE1 = GND, le pont-H est déconnecté et le moteur ne fonctionne pas.

Si ENABLE1 = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT1 & INPUT2.

### III.3.2.4 Le capteur ultrasonique

Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis. Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter.

Ce capteur de distance à ultrasons permet des mesures de distance allant de (2cm à500cm) avec une précision pouvant aller jusqu'à (3mm). L'angle du cône de mesure est d'environ (15°). Le sonar HC-SR04 comprend un émetteur ultrasons, un récepteur ultrasons ainsi qu'un circuit de contrôle.



**Figure III.10** : Capteur sonar à ultrasons HC-SR04.

➤ **Caractéristiques**

- Dimensions : 45 mm x 20 mm x 15 mm
- Plage de mesure : 2 cm à 400 cm.
- Résolution de la mesure : 0.3 cm.
- Angle de mesure efficace : 15 °.
- Angle de mesure : 30°.
- Poids : 8.5g.
- Signal d'entrée trigger : 10  $\mu$ s TTL impulsion.
- Fréquence d'opération : 40 Hz.
- Tension d'exploitation (Voltage d'entrée) : 5V.
- Courant (Ampérage d'entrée) : 15Ma

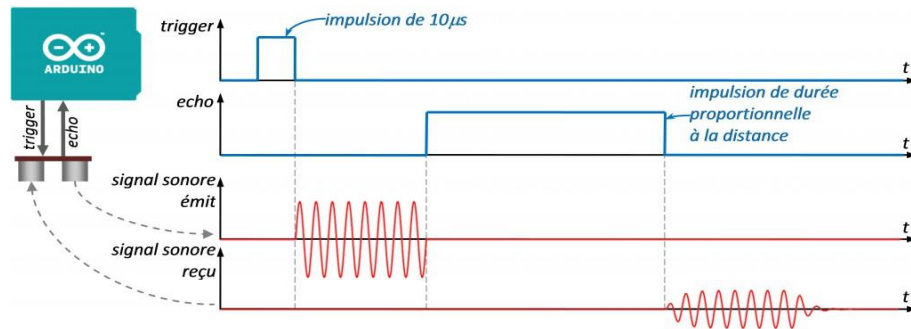
➤ **Broches de connexion**

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

**Remarque** : La borne GND doit être connectée en premier, avant l'alimentation sur Vcc.

➤ **Fonctionnement**

Pour déclencher une mesure, il faut présenter une impulsion "high" (5 V) d'au moins 10  $\mu$  sur l'entrée "Trig". Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal "high" sur la sortie "Echo", dont la durée est proportionnelle à la distance mesurée [27].



**Figure III.11:** Fonctionnement du capteur ultrason.

➤ **Distance de l'objet :**

La distance parcourue par un son se calcule en multipliant la vitesse du son, environ 340 m/s (ou 34'000 cm/1'000'000 µs) par le temps de propagation, soit :  $d = v \cdot t$  (distance = vitesse temps). Le HC-SR04 donne une durée d'impulsion en dizaines de µs. Il faut donc multiplier la valeur obtenue par 10 µs pour obtenir le temps t. On sait aussi que le son fait un aller-retour. La distance vaut donc la moitié.

$$d = 34'000 \text{ cm}/1'000'000 \text{ µs} \cdot 10 \text{ us.valeur} / 2$$

$$\text{En simplifiant : } d = 170'000 / 1'000'000 \text{ cm} \cdot \text{valeur}$$

$$\text{Finalement, } d = 17/100 \text{ cm} \cdot \text{valeur}$$

La formule  $d = \text{durée}/58 \text{ cm}$  figure aussi dans le manuel d'utilisation du HC-SR04 car la fraction 17/1000 est égale à 1/58.8235. Elle donne cependant des résultats moins précis.

**Note :** A grande distance, la surface de l'objet à détecter doit mesurer au moins 0.5 m<sup>2</sup> [27].

### III.3.2.5 Un relais électronique :

Un relais électronique est un type d'interrupteur qui est contrôlé par un circuit électrique au lieu d'une personne. Il est constitué des composants électroniques très simples. Le type le plus basique est doté de contacts qui permettent de le monter sur un circuit imprimé ; une armature reliée aux contacts, un ressort et une armature en fer avec un bobinage. Lorsque le bobinage reçoit une énergie électrique, il crée un champ magnétique. Ce champ magnétique déplace l'armature. Le mouvement de l'armature ouvre ou ferme les contacts, selon la fonction du relais. Selon les besoins, un relais peut être conçu comme normalement ouvert ou normalement fermé. Le relais est caractérisé par : La tension de sa bobine de commande, 5V à 220V.



**Figure III.12:** Le module relais 5V.

### III.3.3 Présentation des Logiciel :

#### III.3.3.1 Le logiciel Arduino IDE :

L'environnement de développement intégré (IDE) Arduino est une application multiplateforme (Pour Windows, MacOS, Linux) écrit en langage de programmation Java. Il est utilisé pour écrire et télécharger des programmes sur des cartes compatibles Arduino.

Le code source de l'IDE est publié sous licence GNU General Public, version 2. L'IDE Arduino supporte les langages C et C++ en utilisant des règles spéciales de structuration de code. L'IDE Arduino fournit une bibliothèque de logiciels du projet Wiring, qui fournit de nombreuses procédures d'entrée et de sortie communes. Le code écrit par l'utilisateur nécessite seulement deux fonctions de base, pour le démarrage de l'esquisse et de la boucle du programme principal, qui sont compilées et liées à un talon de programme `main()` dans un programme exécutable cyclique avec la chaîne d'utils GNU, également incluse dans la distribution IDE.

L'IDE Arduino utilise le programme Avrdude pour convertir le code exécutable en un fichier texte au codage hexadécimal chargé dans la carte Arduino par un programme de chargement dans le microprogramme de la carte.

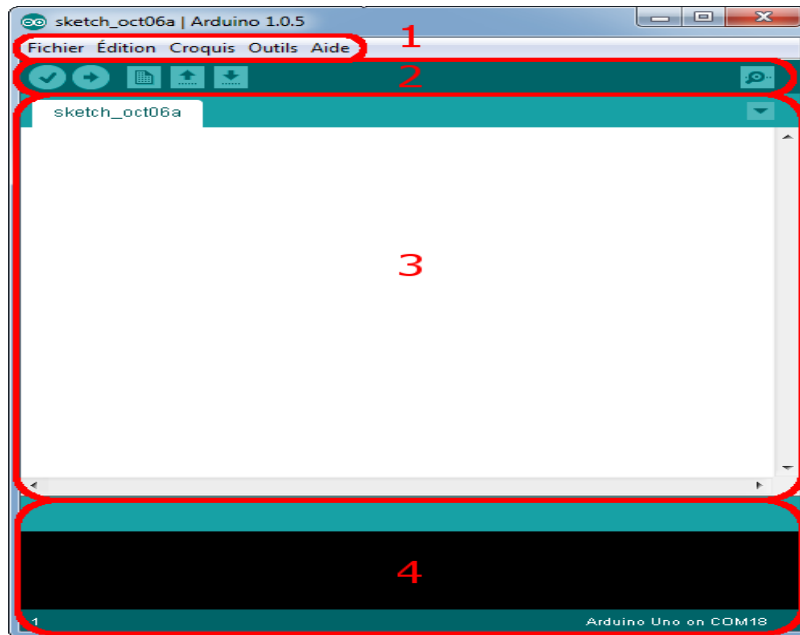
#### ➤ Présentation du logiciel :

**Le cadre numéro 1 :** ce sont les options de configuration du logiciel.

**Le cadre numéro 2 :** il contient des boutons qui vont nous servir lorsque l'on va programmer nos cartes.

**Le cadre numéro 3 :** ce bloc va contenir le programme que nous allons créer.

**Le cadre numéro 4 :** celui-ci est important, car il va nous aider à corriger les fautes de notre programme. C'est le débogueur.



**Figure III.13:** L'interface de l'IDE Arduino.

### III.3.3.1.1 Approche et utilisation de logiciel :

#### ➤ Le menu file :

C'est principalement ce menu que l'on va utiliser le plus. Il dispose d'un certain nombre de chose qui nous vont nous être très utiles. Il a été traduit en français progressivement, nous allons donc voir les quelques options qui sortent de l'ordinaire :

- Carnet de croquis : ce menu regroupe les fichiers que vous avez pu faire jusqu'à maintenant (et s'ils sont enregistrés dans le dossier par défaut du logiciel).
- Exemples : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existants avec ça, vous pourrez vous aider /inspirer pour créer vos propres programmes ou tester de nouveaux composants.
- Téléverser : Permet d'envoyer le programme sur la carte Arduino.
- Téléverser avec un programme : Identique que ci-dessus, mais avec l'utilisation d'un programmeur (vous n'en n'aurez que très rarement besoin).
- Préférences : vous pourrez régler ici quelques paramètres du logiciel.

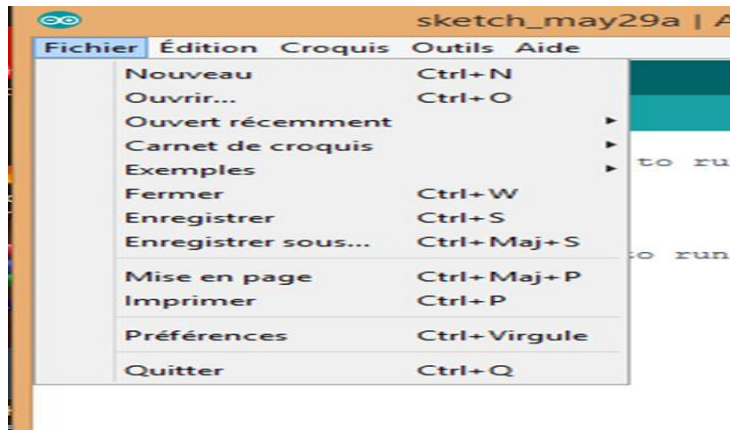


Figure III.14: Le menu fichier.

➤ **Le menu outil :**

Une fois le code écrit (ou collé) dans la fenêtre de programmation, il faut l'envoyer sur l'Arduino. Pour cela, après avoir connecté l'Arduino à l'ordinateur, il faut sélectionner le port et le type de carte (Arduino Uno, dans notre cas).

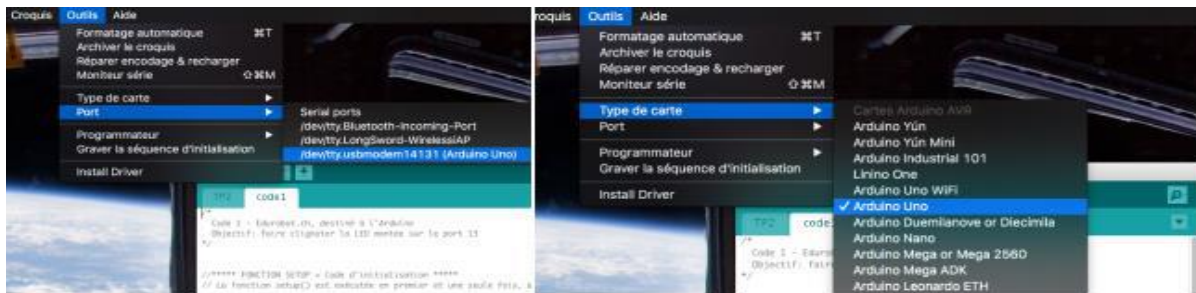


Figure III.15: Le menu outils.

➤ **Les boutons :**

Bouton 1 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.

- Bouton 2 : Charger (téléverser) le programme dans la carte Arduino.
- Bouton 3 : Créer un nouveau fichier.
- Bouton 4 : Ouvre un fichier.
- Bouton 5 : Enregistrer le fichier.
- Bouton 6 : Ouvre le moniteur série.

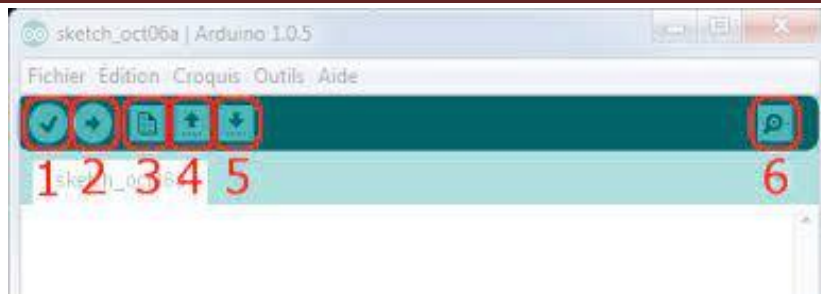


Figure III.16: La barre d’outils.

**III.3.3.1.2 structure d'un programme Arduino :**

Un programme Arduino est une suite d’instructions élémentaires et séquentielles, dans tous les programmes Arduino il y a trois étapes. Pour mettre en évidence ces phases on va présenter la structure d’un exemple simple de programmation qui consiste à allumer une Led pendant 1 seconde puis l’éteindre durant 3 secondes sur la broche N°13 ainsi de suite jusqu’à introduire un nouveau programme ou bien arrêter l’alimentations.

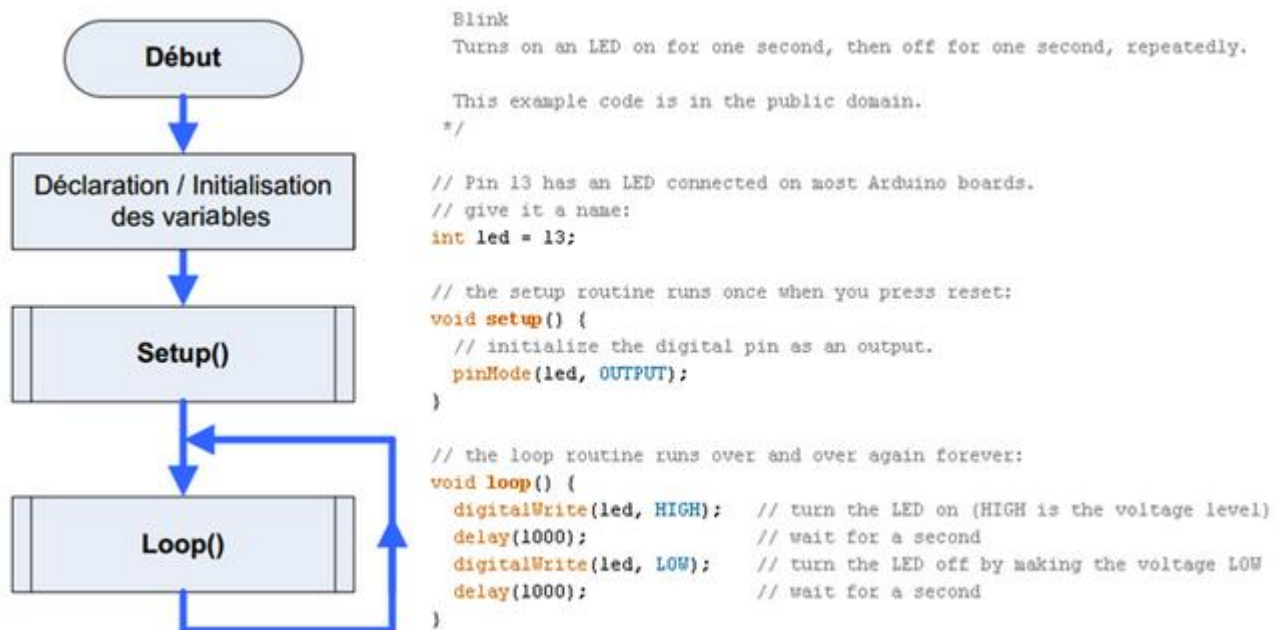


Figure III.17: Exemple d'un programme Arduino.

✓ **Définition et déclaration des variables :**

Pour notre exemple, on va utiliser une sortie numérique de la carte qui est par exemple la 13ème sortie numérique. Cette variable doit être définie et nommée dans cette partie ; On lui donne un nom arbitraire Led.

✓ **Configuration des entrées-sorties voidsetup () :**

Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques. Ici on va configurer Broche LED en sortie PinMode (nom, état) est une des fonctions relatives aux entrées-sorties numériques.

✓ **Programmation des interactions voidloop () :**

Dans cette boucle, on définit les opérations à effectuer, dans l'ordre :

- **DigitalWrite (nom, état) :** Une des fonctions relatives aux entrées-sorties numériques.
- **DeLay (temps en millisecondes) :** La commande d'attente entre deux autres instructions
- Chaque ligne d'instruction est terminée par un point-virgule.
- Les accolades encadrent la boucle [28],[29].

✓ **Commentaires :**

Pour écrire des commentaires sur le programme (cela aide à la relecture du programme et sa compréhension par une autre personne que celle qui l'a fait) on a 2 possibilités :

Soit en multi ligne, en les mettant entre les signes `/** */`.

Soit sur une ligne de code en les séparant du code avec les signes `//`.

Le microcontrôleur de la carte Arduino va recevoir le programme que nous aurons créé et qui va le stocker dans sa mémoire puis l'exécuter. Grâce à ce programme, il va envoyer au robot les tâches à exécuter et des données qu'il reçoit par les capteurs à un ordinateur ...

➤ **Syntaxe du langage Arduino :**

La syntaxe des commandes d'Arduino est donnée dans la table des matières suivante. Chaque instruction est suivie de sa traduction entre-parenthèses [30],[31].

Commandes de structure du programme	
Structure générale	<ul style="list-style-type: none"> <li>• <b>Voidsetup ()</b> (configuration, préparation)</li> <li>• <b>Voidloop ()</b> (exécution)</li> </ul>
Contrôle et conditions	<ul style="list-style-type: none"> <li>• <b>If</b> (si...)</li> <li>• <b>If ...else</b> (si...sinon...)</li> <li>• <b>For</b> (pour...)</li> <li>• <b>Switch case</b> (dans le cas où...)</li> <li>• <b>While</b> (pendant que...)</li> </ul>
Opérations de comparaison	<ul style="list-style-type: none"> <li>• <b>==</b> (équivalent à)</li> <li>• <b>!=</b> (différent de)</li> <li>• <b>&lt;</b> (inférieur à)</li> <li>• <b>&gt;</b> (supérieur à)</li> </ul>

	<ul style="list-style-type: none"> <li>• <code>&lt;=</code> (inférieur ou égal à)</li> <li>• <code>&gt;=</code> (supérieur ou égal à)</li> </ul>
Opérations booléennes	<ul style="list-style-type: none"> <li>• <code>&amp;&amp;</code> (et)</li> <li>• <code>  </code> (ou)</li> <li>• <code>!</code> (et pas)</li> </ul>
Autres commandes	<ul style="list-style-type: none"> <li>• <code>//</code> (commentaires simple ligne)</li> <li>• <code>/**/</code> (commentaires multilignes)</li> <li>• <code>#define</code> (donner une valeur à un nom)</li> </ul>

**Tableau III.2** Les commandes du programme Arduino.

<b>Fonctions</b>	
Entrées-sorties numériques	<ul style="list-style-type: none"> <li>• <code>PinMode (broche, état)</code> (configuration des broches).</li> <li>• <code>DigitalWrite (broche, état)</code> (écrire un état sur une broche numérique).</li> <li>• <code>DigitalRead(broche)</code> (lire un état sur une broche numérique).</li> <li>• <code>Unsigned long pulseIn (broche, état)</code> (lire une impulsion sur une broche numérique).</li> </ul>
Entrées analogique	<ul style="list-style-type: none"> <li>• <code>intanalogRead (broche)</code> (lire la valeur d'une broche analogique).</li> <li>• <code>analogWrite(broche, valeur)</code>(PWM : écrire une valeur analogique sur les broches 9, 10 ou 11).</li> </ul>
Gestion du temps	<ul style="list-style-type: none"> <li>• <code>unsigned long millis()</code> (temps de Fonctionnement du programme)</li> <li>• <code>DeLay(ms)</code> (attente, en millisecondes)</li> <li>• <code>delayMicroseconds(us)</code> (attente, en µs).</li> </ul>
Nombres aléatoires	<ul style="list-style-type: none"> <li>• <code>randomSeed(seed)</code> (aléatoire 'piloté')</li> <li>• <code>Long random(max)</code> (aléatoire à partir de telle valeur).</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Long random (min, max)</b> (aléatoire entre deux valeurs).</li> </ul>
--	---

**Tableau III.3** Les fonctions du programme Arduino.

<b>Variable</b>	
Variables	<ul style="list-style-type: none"> <li>• <b>Char</b> (variable ‘caractère’)</li> <li>• <b>Int</b> (variable ‘nombre entier’)</li> <li>• <b>Long</b> (variable ‘nombre entier de très grande taille’)</li> <li>• <b>String</b> (variable ‘message’)</li> </ul>
Niveaux logiques des connecteurs numériques	<ul style="list-style-type: none"> <li>• <b>HIGH</b> (état 1)</li> <li>• <b>LOW</b> (état 0)</li> <li>• <b>INPUT</b> (configuré en entrée)</li> <li>• <b>OUTPUT</b> (configuré en sortie)</li> </ul>

**Tableau III.4** Les différentes variables du programme Arduino.

**III.3.3.2 Logiciel Proteus**

Cette suite logicielle est très connue dans le domaine de l'électronique. Elle est utilisée dans de nombreuses entreprises et organismes de formation. Outre la popularité de l'outil, Proteus possède d'autres avantages :

- Pack contenant des logiciels faciles et rapides à comprendre et à utiliser ;
- Support technique performant ;
- Outil de création de prototype virtuel permettant de réduire les coûts matériel et logiciel lors de la conception d'un projet.

**III.3.3.3.1 ISIS**

Le module ISIS de Proteus est principalement utilisé pour éditer un schéma structurel d'un circuit électronique (assemblage de composants électroniques dont on fixe les valeurs et les références) reliés par des connexions électrique (fils). Par ailleurs, le logiciel permet également de simuler les différents types de montages ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits. La figure ci-dessous montre la fenêtre du module de simulation Proteus ISIS.

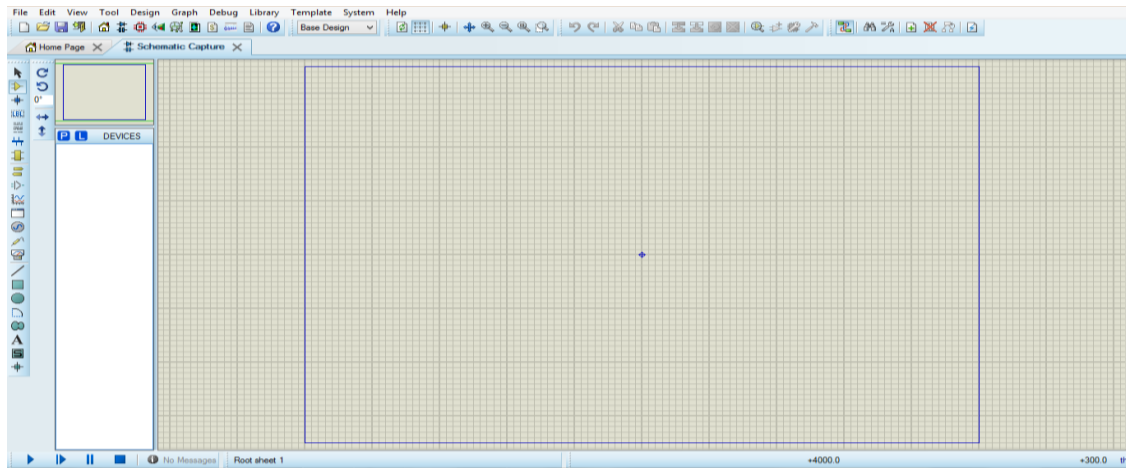


Figure III.18: Fenêtre du module de simulation Proteus ISIS.

### III.4 Tableau devis :

Les composants	Quantité	Prix	Prix total
Arduino Uno	1	2000 DA	11650 DA
Module Bluetooth	1	1300 DA	
Le relais	1	600 DA	
Kit de roue avec moteur DC	3	1200 DA	
Batterie 12v	1	3500 DA	
Pompe 12v	1	500 DA	
Driver Moteur L293D	1	1200 DA	
Capture ultrason HC-SR 04	1	700 DA	
Servomotor	1	650 DA	

### III.4 Conclusion

Dans ce chapitre nous avons étudié les différentes composants que on a utilisés afin de réaliser notre robot, suivi d'une présentation de logiciel de simulation Proteus ISIS et Fritzing ; ainsi que le logiciel de programmation Arduino IDE et la structure de programmation.

Après avoir eu la liste de matériels et logiciels dont on a besoin pour le projet et comprendre leurs fonctionnements, il nous reste qu'assembler ces composants de façon raisonnable pour réaliser un robot qui fonctionne selon nos besoins, c'est ce qu'on va voir dans le chapitre suivant.

Chapitre IV

**Réalisation de projet**

## IV.1 Introduction

Dans le chapitre précédent nous avons fait une conception o on a défini tous les composants matériels et logiciels qu'on va utiliser dans notre réalisation.

Ce chapitre présente un organigramme qui affiche l'objectif de projet, ainsi le mode de fonctionnement de processus.

## IV.2 Schéma synoptique

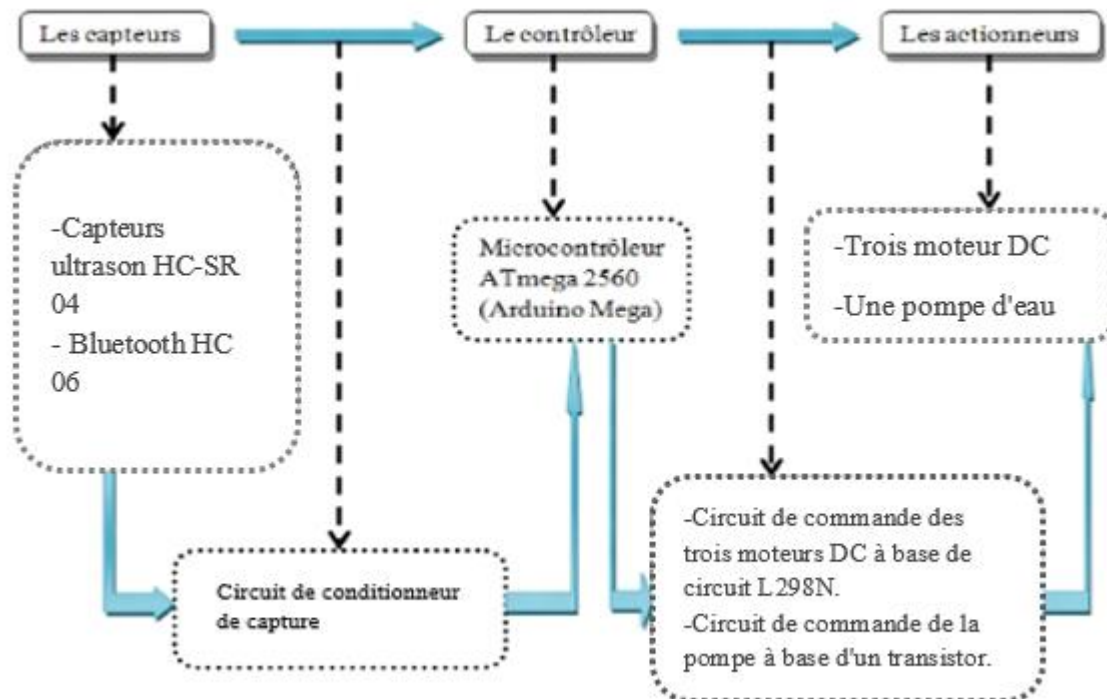


Figure IV.1: Schéma synoptique de projet.

## IV.3 Déroulement de projet

Notre projet de réalisation a été fait en deux parties :

✓ La première partie est la conception assistée par ordinateur CAO, la simulation avec Proteus ISIS.

✓ La deuxième partie est la réalisation, le montage pratique.

## IV.4 Réalisation de projet

Un robot est un système électromécanique capable d'interagir avec son environnement et de prendre des décisions ou actions indépendantes afin de réaliser un travail spécifique.

Le robot est composé des éléments suivants :

1. Structure
- 2- Opérateur mécanique.
- 3- Unité de contrôle.
4. Entrées / Capteurs.
- 5- Alimentation.

Dans les étapes suivantes, nous allons mentionner chacun des éléments ci-dessus, afin qu'ils puissent être facilement compris.

### IV.4.1 Structure/Châssis

Un robot est constitué d'un ou de plusieurs composants physiques qui se déplacent pour effectuer certaines tâches. Dans notre cas, le châssis et les roues représentent la structure du robot.

### IV.4.2 Opérateur mécanique

Un moteur est un dispositif qui convertit l'énergie électrique en énergie mécanique. La plupart des moteurs produisent un mouvement de rotation ou linéaire.

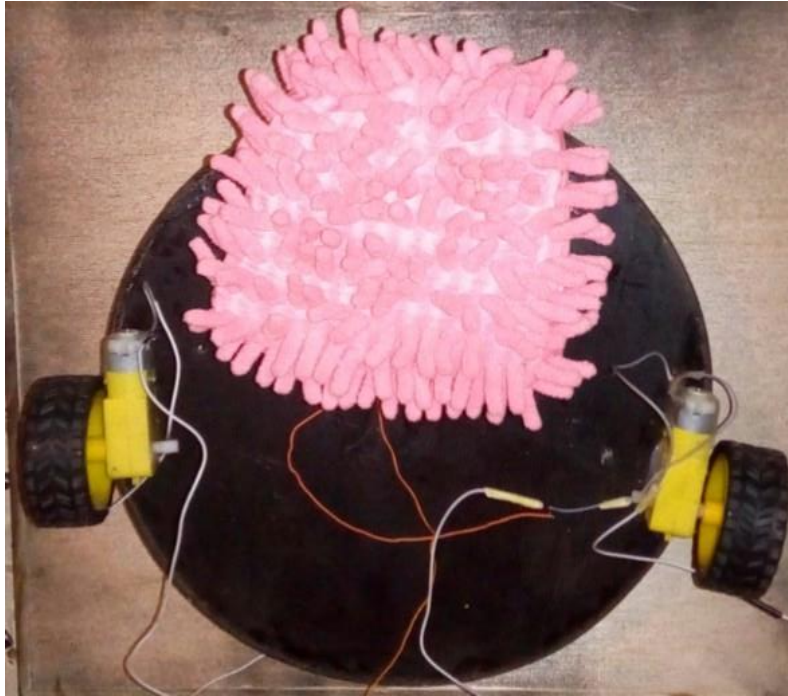
Dans notre cas, le moteur est un moteur à courant continu (DC Motor) et on peut installer un potentiomètre pour régler la vitesse.

#### ✓ Préparation et l'installation des moteurs :

Couper quatre morceaux de fil (rouge et noir) avec une longueur d'environ 5 à 6 pouces. Dénudez le fil de l'isolant à chaque extrémité et soudez-le aux moteurs.

Vous pouvez vérifier la polarité des moteurs en les connectant à la batterie. S'il tourne dans le sens avant (fil rouge avec électrode positive et noir avec électrode négative), cela indique que la conduction a été effectuée correctement.

Notez que les fils de chaque moteur indiquent la direction du centre du châssis. La meilleure partie est que les vadrouilles utilisées étaient faites maison, à partir de vieux CD et de chiffons et elles nettoient parfaitement. Il s'agit d'une version plus petite et peut donc ne pas convenir à une grande



**Figure IV.2** : Préparation et installation des moteurs.

### IV.4.3 Unité de contrôle

Le microcontrôleur est nécessaire pour déplacer le robot d'un endroit à un autre. Le contrôleur a la capacité d'exécuter un programme et est responsable de tous les calculs, de la prise de décision et de la communication. Dans ce projet, le microcontrôleur Arduino sera utilisé comme contrôleur.

Vous pouvez utiliser le contrôleur pour faire tourner les moteurs dans un sens. Mais si vous voulez pouvoir contrôler le moteur dans différentes directions (avant, arrière, droite, gauche) avec le panneau de commande, vous avez besoin de plus de circuits. Vous avez besoin d'un pont en H.

### IV.4.4 Entrées / Capteurs

- Le capteurs HC-SR 04 Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal "high" sur la sortie "Echo", dont la durée est proportionnelle à la distance mesurée.
- La technologie Bluetooth vise à échanger des données sans fil sur des ondes radio à courte portée entre des périphériques formant un réseau personnel avec une distance limitée.

Le Bluetooth HC-06 fonctionne sur une connexion série, à travers laquelle vous pouvez envoyer et recevoir des données. Vous pouvez donc utiliser un terminal série ou un périphérique compatible Bluetooth tel qu'Android pour contrôler le projet.

Bluetooth possède deux ports d'alimentation, un port TX qui transmet des données via Bluetooth et un port RX qui reçoit les données envoyées à Bluetooth.

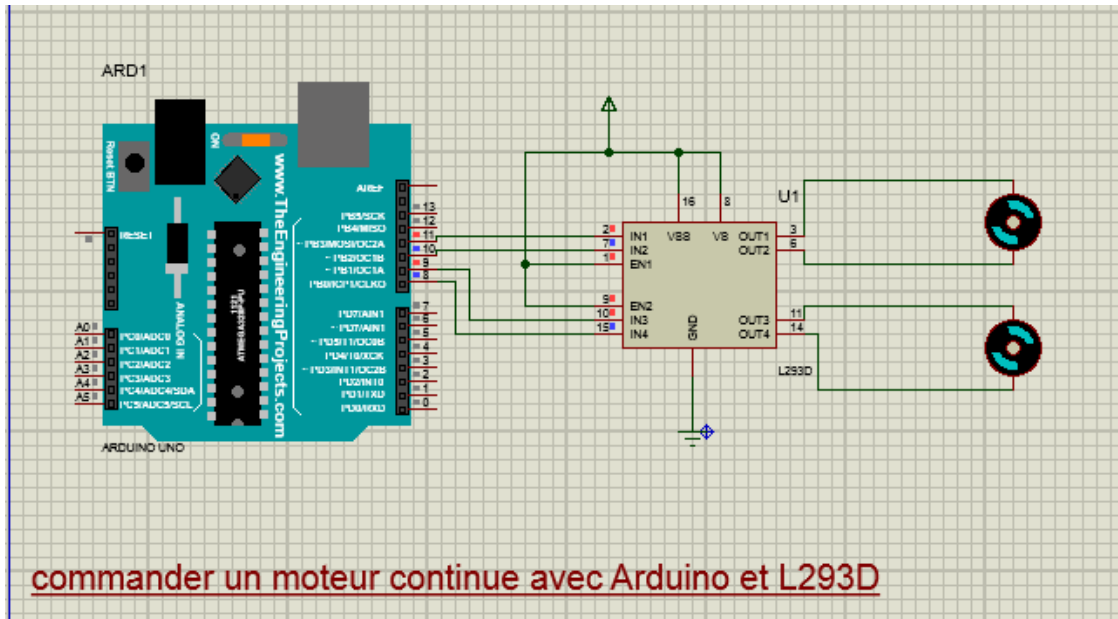
**IV.4.5 Alimentation**

Pour la source d'alimentation on a utilisé trois batteries rechargeables de 3,7 V.

**IV.4.6 Branchement des différentes parties de projet**

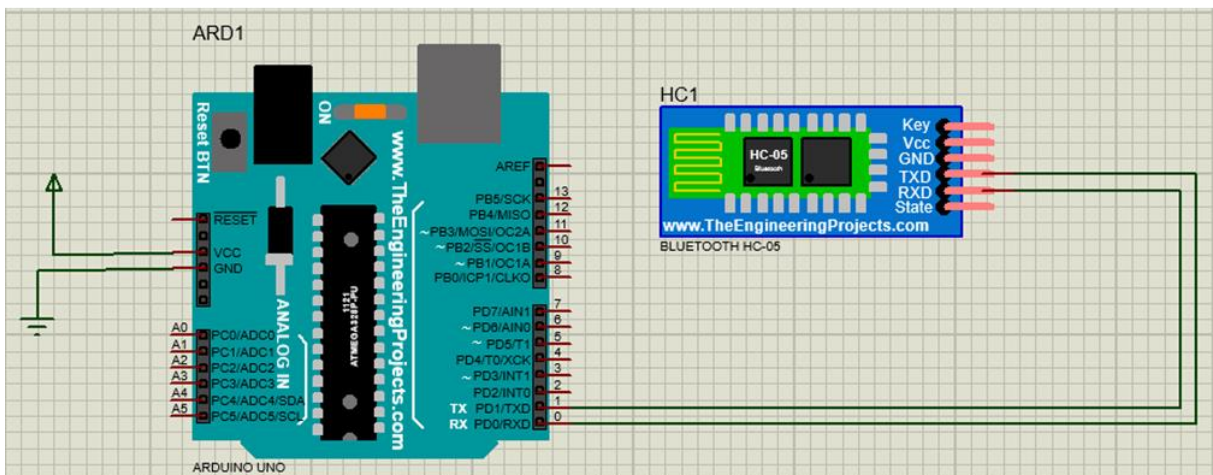
❖ **La connexion entre L'Arduino et les moteurs :**

Pour cette on a utilisé la carte Arduino, deux (02) moteurs à courant continu, un circuit intégré L293D.



**Figure IV.3:** La connexion entre L'Arduino et le driver moteur et les moteurs sur Proteus.

❖ **Circuit Arduino-Bluetooth HC-06 :**



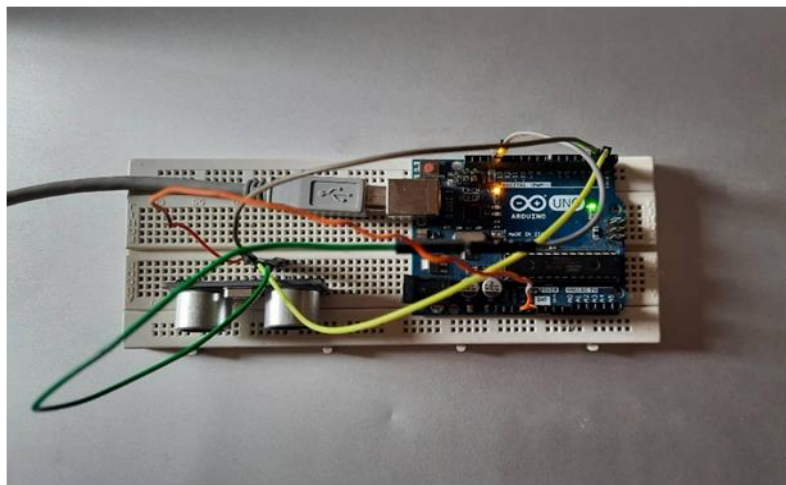
**Figure IV.4:** Branchement et simulation de Bluetooth avec Arduino.

❖ **Intelligent et Eviter les obstacles :**

Dans cette partie on s'intéresse à la réalisation d'un robot qui sera capable de déplacer tout en évitant les obstacles. On a choisi un capteur de distance du type Ultrason HC-SR04. Ce robot est constitué d'un capteur Ultrason, une carte Arduino, deux moteurs à courant continue, un circuit L293D

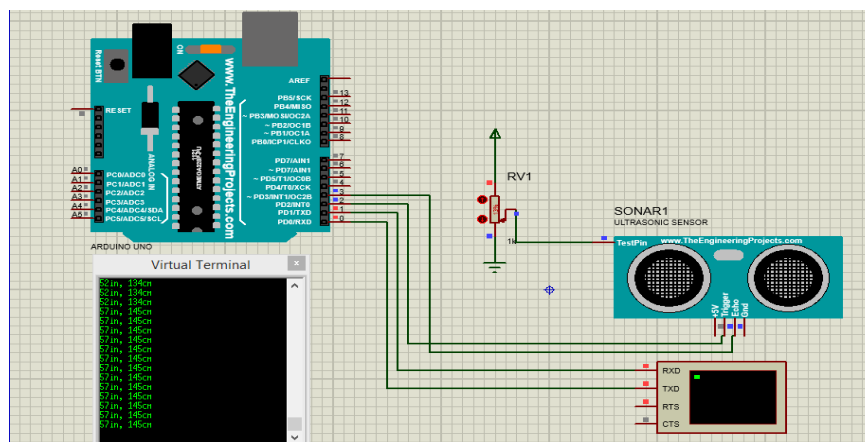
✓ **Circuit d'évitement d'obstacle :**

Le capteur doit être alimenté par une tension de 5V et son GROUND doit être connecté avec le GROUND d'Arduino. Son ECHO et TRIG sont connectés à des PINs digitales.



**Figure IV.5:** Montage réel de capteur HC-SR04 avec Arduino.

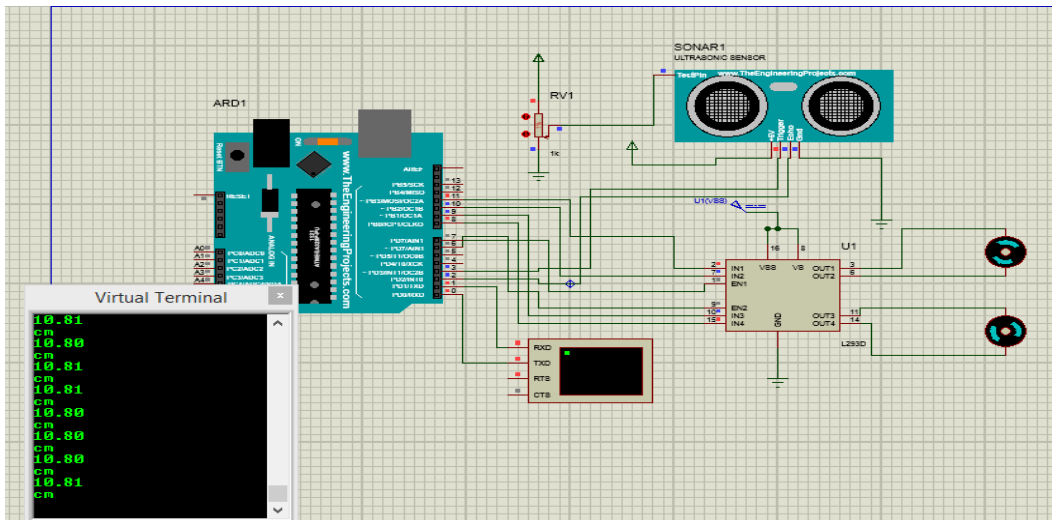
Pour cette simulation on a utilisé la bibliothèque Arduino et Ultrasons. On a dû utiliser une résistance variable pour simuler la présence d'un obstacle et un terminal virtuel où son Rx est connecté avec le PIN digital Tx. Après on doit injecter le fichier de l'extension « .hex » du sketch compilé.



**Figure IV.6:** Simulation d'Arduino avec capteur HC-SR04.

✓ **Circuit du robot intelligent :**

Dans ce circuit on doit intégrer le capteur ultrason avec le circuit intégré L293D qui va commander les deux moteurs.



**Figure IV.7 :** Simulation du robot intelligent sur Proteus.



**Figure IV.8:** Montage réel de robot intelligent.

IV.4.7 Schéma électrique du projet

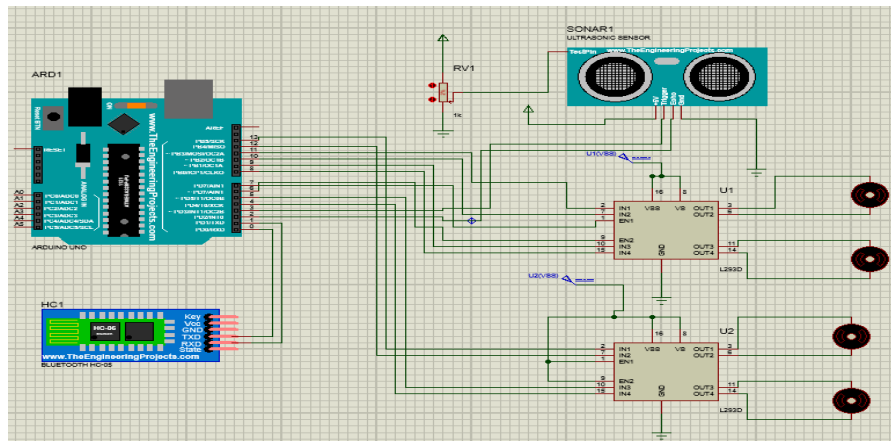


Figure IV.9: Schéma électrique de projet.

IV.5 Programmation

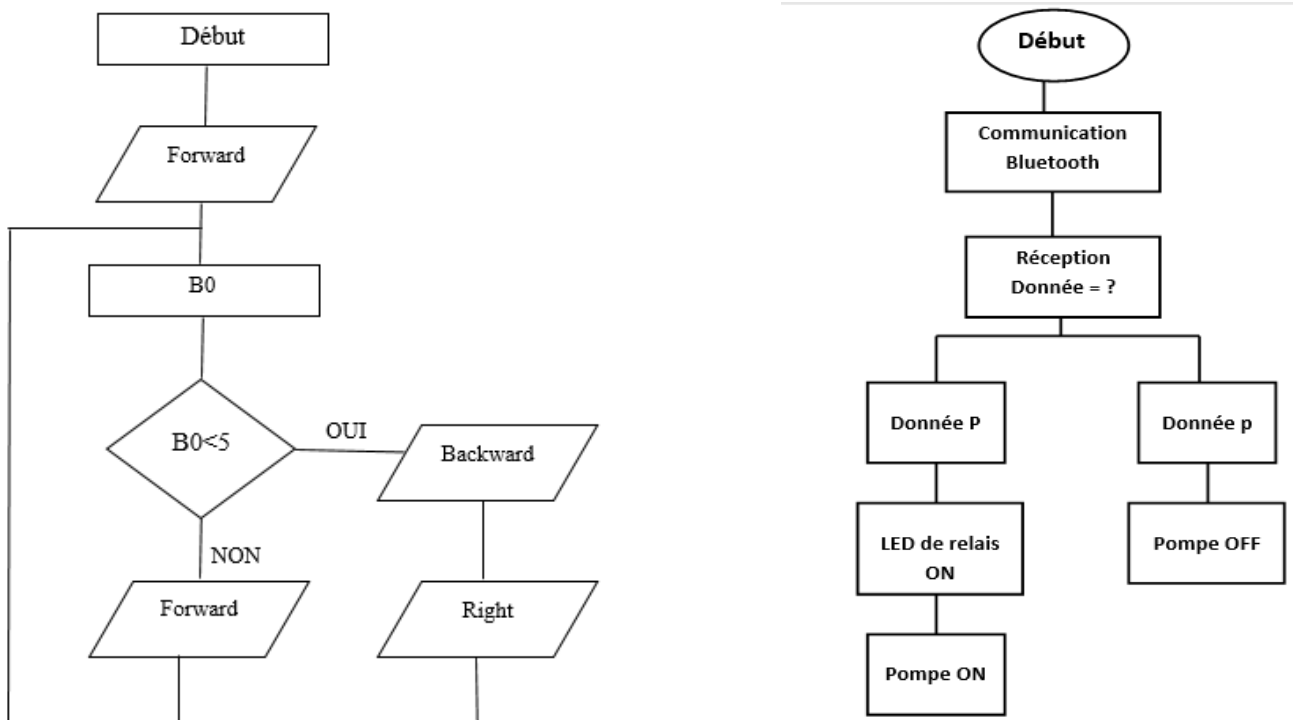
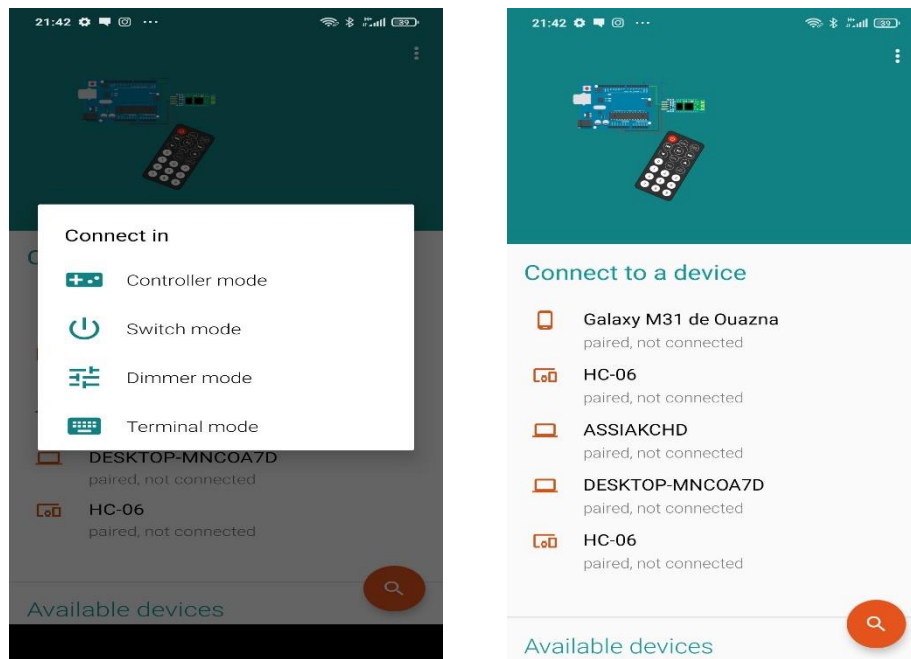


Figure IV.10: Organigramme d'éviteur d'obstacle / de commande de la pompe à eau

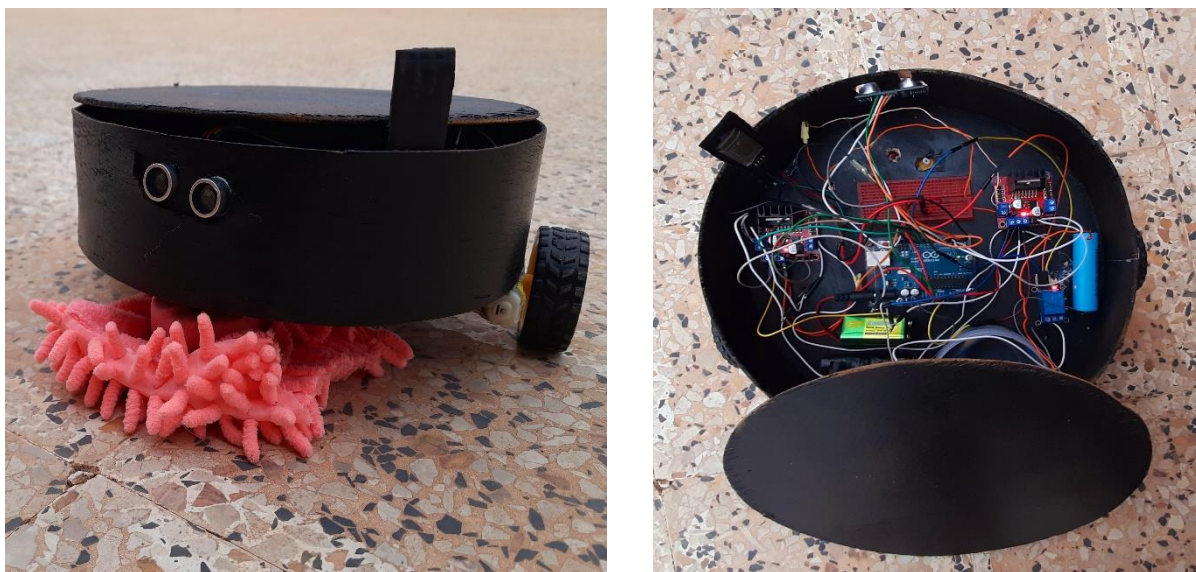
## IV.6 Application Android

Coupler votre appareil avec Bluetooth : HC-06

- Opération Bluetooth. HC-06
- Analyser le périphérique disponible.
- Le couplage à Bluetooth est effectué en saisissant le mot de passe par défaut 1234 ou. 0000. Ouvrez l'application, puis appuyez sur le bouton Appareils associés, puis sélectionnez une unité. Votre Bluetooth.



**Figure IV.11:** Capteur d'écran de l'interface de l'application



**Figure IV.12 :** Réalisation final

## IV.6 Conclusion

Ce chapitre était le noyau de notre projet de fin d'étude. Dans ce dernier, nous avons illustré les étapes suivies pendant la réalisation de notre prototype mobile nettoyeur de sol, en figurant les schémas de simulation de chaque partie réalisée. On a commencé par la partie mécanique (montage des moteurs), puis, les simulations des circuits avec ISIS après l'injection des fichiers « .hex » qui nous avait donné une observation et une confirmation de l'état de ces derniers, en finissant par la réalisation de prototype.

# Conclusion Générale

Ce mémoire nous a permis d'acquérir des connaissances théoriques et pratiques pendant nos recherches et formation, notamment atteindre notre objectif qui consiste à réaliser un robot mobile, qui est un prototype qui pourra être bien évidemment amélioré en diversifiant ses tâches en lui incluant de nouveaux composants afin d'augmenter ses performances.

L'utilisation de la carte Arduino programmable en langage C nous a permis de gagner en temps et en simplicité du programme. Nous avons choisi notre matériel en fonction de nos besoins puis essayé différentes solutions afin d'arriver aux fonctionnalités souhaitées. La réalisation de ce robot nous a également donné l'opportunité de travailler sur plusieurs domaines à la fois, notamment la programmation et l'électronique. Nous avons donc au final pu mener un vrai travail de conception de système, ce qui s'apparente le plus à un travail d'ingénieur. De manière générale, ce projet nous a fait découvrir le monde de la robotique et toutes ses perspectives professionnelles.

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

---

## Bibliographies

- [01] Cyril, D. « Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels ». Thèse de Doctorat en Robotique, Université de technologie de Compiègne, centre de Robotique, d'Electrotechnique et d'Automatique, 2002.
- [02] Robotics, International Federation of World robotics, Editions 2006 à 2011.
- [03] Laetitia Matignon, cours « Introduction à la robotique », Université de Caen, France,2011-2012.
- [04] Tropato Jean-Charles, « La robotique industrielle et de service : La place de l'Italie dans le Monde », Rapport d'étude. Ambassade de France en Italie, 2011.
- [05] Yann leidwanger, « robot mobile intelligents - du capteur au comportement », 2006.
- [06] Oumma Licer, « commande des allures de marche stable d'un robot bipède type compas », université Sidi Mohammed Ben Abdallah, Maroc ,2007.
- [07] Bernard Bayle, « robotique mobile » télécom physique Strasbourg 3A, option ISAV, master IRIV.
- [08] FILLIAT David, « Robotique Mobile », Ecole Nationale supérieur de Techniques avancées, Paris Tech, 2013.
- [09] J. Paul Laumond, « la Robotique Mobile », Editions Hermé, 2001.
- [10] Ulrich. N, «MOBILE ROBOTICS: A PRACTICAL». Springer, Verlag London, 2000.
- [11] Abdelmalek. B, Planification de trajectoire, Département d'électronique, BATNA, 2012.
- [12] Latombe J. C. «Robot Motion Planning», Kluwer Academic Publishers, Norwell, 1991.
- [13] J. Borenstein, «Where am I, Sensors and Methods for Mobile Robot Positioning», University of Michigan, 1996.
- [14] J. Borenstein, «Where am I, Sensors and Methods for Mobile Robot Positioning», University of Michigan, 1996.
- [15] F. Cuesta, « Intelligent Mobile Robot Navigation », Springer-Verlag, Berlin Heidelberg,2005.
- [16] Reignier P, « Pilotage Réactif d'un Robot Mobile, Etude de Lien de la Perception », Thèse de Doctorat, Institut National Polytechnique de Grenoble,1994.
- [17] C. Vibet « Robots Principes et Contrôles » Edition ellipse, 1987.
- [18] W. Khalil E Dombre, « Modélisation, Identification et Commande de Robots » Edition Hermès, 1999.
- [19] A. Pruski « Robotique Générales », Edition ellipse, 1988.

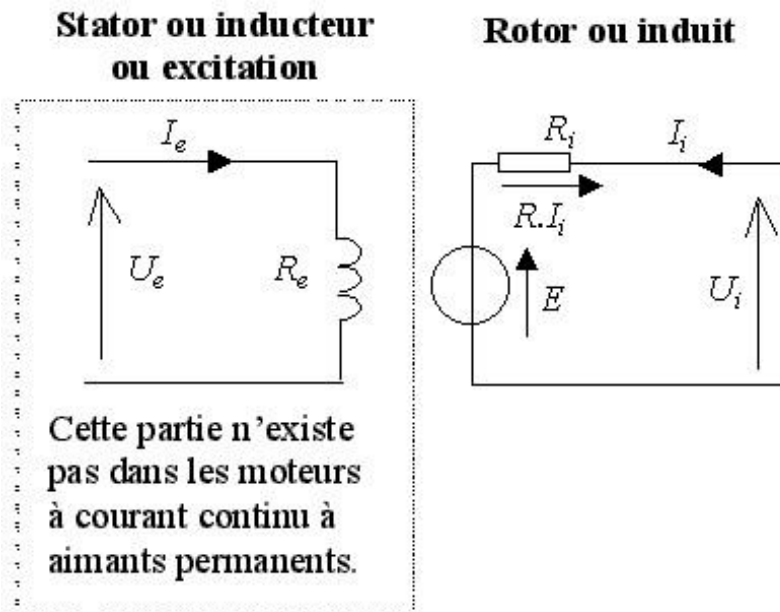
- 
- [20] J.O. B, Statistical Decision Theory and Bayesian Analysis. Springer Verlag Berlin, C. P.S. (1 2011). Service robotics is a key market of the future, 1985.
- [21] B. Bayle, « Robotique Mobile », Ecole Nationale Supérieure de Physique de Strasbourg, Université Louis Pasteur, 2007.
- [22] Y. Cang, N. H, « Aoller with Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance», IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 33, 2003.
- [23] O. Lefebvre « Navigation Autonome sans Collision pour Robots Mobiles non holonomes », Thèse de Doctorat de l'Institut National Polytechnique de Toulouse, 2006.
- [24] CHERROUN Lakhmissi « Navigation Autonome d'un Robot Mobile par des Techniques Neuro-Floues », Thèse de Doctorat en Automatique, Université Mohamed Khider, Biskra ,2014.
- [25] A poros de mchob by-Wiki. Servo-Moteur0URL : <http://dSPACE.univ-tlemcen.dz>.
- [26] Kafi Mohammed Mustafa, Houamed Abdelhak, Un radar de recul à Ultrasons, Mémoire professionnel. Université de Kasdi Merbah\_Ouargla.2018. URL : <https://dSPACE.univ-ouargla.dz>.
- [27] Lucien Bachelard, « HC-SR04 - Module de détection aux ultrasons - Utilisation avec Picaxe », URL : <https://www.gotronic.fr>. Consulté le 06/08/2021.
- [28] SIMON Landranlt, HIPPOLYTE Weiss linger, « Premier pas en informatique embarquée », Edition CC BY-NC-SA,01 Juin 2014, 268p.
- [29] Christian Tavenir, « Arduino, maîtrisez sa programmation et ces cartes d'interface (Shields) », Collection Dunod, Mars 2014, 256p.
- [30] Brian W. Evans «Arduino Programming Notebook », First edition, August 2007,38p.
- [31] <https://www.arduino.cc>.Consulté le 20/08/2021.



---

## Annex II

### Moteur à courant continu



## Annex III

### Programme principale de Robot

```
#include <HCSR04.h>
HCSR04 hc (5,6) ;//initialisation class HCSR04 (trig pin, echo pin)
char command;
int pompe =13;
int incomingByte = 0;
void setup ()
{Serial.begin (9600);
 pinMode (8, OUTPUT);
 pinMode (9, OUTPUT);
 pinMode (10, OUTPUT);
 pinMode (11, OUTPUT);
 pinMode (3, OUTPUT);
 pinMode (4, OUTPUT);
 pinMode (12, OUTPUT);
 pinMode (13, OUTPUT);
}
void loop ()
{
  Serial.println(hc.dist() ) ;
  // en avant
```

---

```
digitalWrite (8, HIGH);
digitalWrite (9, LOW);
digitalWrite (10, HIGH);
digitalWrite (11, LOW);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);

if(hc.dist() <5){
  digitalWrite (8, LOW); // marche arriere
  digitalWrite (9, HIGH);
  digitalWrite (10, LOW);
  digitalWrite (11, HIGH);
  delay (1500);
  // changer la direction de Robot
  digitalWrite (8, LOW);
  digitalWrite (9, LOW);
  digitalWrite (10, HIGH);
  digitalWrite (11, LOW);
  delay (1000);

}
If (Serial.available() > 0) {
  command = Serial.read();
switch(command) {
  case'P':

    digitalWrite (13, HIGH);
    Serial.println ("pump on");
    incomingByte='*';
    break;
  case 'p': // pump off
    digitalWrite (13, LOW);
    Serial.println("pump off");
    incomingByte='*';
    break;

  }
}

} //return current distance in serial
```



## ملخص:

العمل المقدم في هذه الأطروحة عبارة عن تصميم وإنتاج نموذج آلي الروبوت منظم ذاتي يتم توجيهه بواسطة لوحة أرد وينو. تقوم لوحة أرد وينو، المجهزة بمتحكم دقيق ATmega328، بمعالجة المعلومات التي تم الحصول عليها بواسطة مستشعرات HC-SR04، ويستخدم المتحكم الدقيق هذه المعلومات للتحكم في المحركات. الغرض منه هو تجنب العقبات وهو يقوم بتنظيف الأرض (يتحرك للأمام، للخلف، توقف فوري لبضعة أجزاء من الثانية، ينعطف). حيث يمكن طلبه باستخدام تطبيق Android.

يتم تلخيص هذا العمل المتواضع في أربعة فصول: يحتوي الأول على مراجعة بيبليوغرافية عن الروبوتات بشكل عام. والثاني مخصص للتفاصيل والتعمق في تكنولوجيا الروبوتات المتنقلة. يقدم الفصل الثالث وصفًا للمكونات المستخدمة وخصائصها؛ بالإضافة إلى عرض برنامج محاكاة Proteus ISIS. يعرض الفصل الأخير تحقيق المشروع.

**كلمات مفتاحية:** الروبوت، أرد وينو، محاكاة Proteus ISIS، المتحكم

## Résumé :

Le travail présenté dans ce mémoire, est une conception et réalisation d'un robot prototype self cleanser guidé par une carte Arduino Uno. La carte **Arduino**, muni d'un microcontrôleur **ATmega328**, traite les informations acquises par les capteurs **HC-SR04**, le microcontrôleur utilise ces informations pour commander les moteurs. Dont le but d'éviter les obstacles en nettoyant le sol (avancer, marche arrière, arrêt immédiat de quelque milliseconde, tourner). Comme il peut être commandé avec une application Android.

Ce modeste travail est résumé en quatre chapitres : le premier contient une revue bibliographique sur les robots en général. Le deuxième est consacré à détailler et approfondir dans la technologie des robots mobiles. Le troisième chapitre présente la description des composants utilisés et leurs caractéristiques ; ainsi que la présentation de logiciels de simulation Proteus ISIS. Le dernier chapitre présente la réalisation de projet.

**Mot clés :** Robot, Arduino, Simulation Proteus ISIS, Microcontrôleur.

## Abstract :

The work presented in this thesis is a design and production of a self-cleanser prototype robot guided by an Arduino Uno board. The Arduino board, equipped with an ATmega328 microcontroller, processes the information acquired by the HC-SR04 sensors, the microcontroller uses this information to control the motors. The purpose of which is to avoid obstacles by cleaning the ground (forward, reverse, immediately stop for a few milliseconds, turn). As it can be ordered with an Android app.

This modest work is summed up in four chapters: the first contains a bibliography review on robots in general. The second is devoted to detail and deepen in the technology of mobile robots. The third chapter presents the description of the components used and their characteristics; as well as the presentation of Proteus ISIS simulation software. The last chapter presents the realization of the project.

**Keywords:** Robot, Arduino, Proteus ISIS Simulation, Microcontroller.