

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies
Filière : Génie électrique

Spécialité : **Commande des systèmes**

Présenté par
Ahcene HAMOUDI
Karim BERKANI

Thème

Conception et réalisation d'un robot mobile autonome

Mémoire soutenu publiquement le 13/07/ 2016 devant le jury composé de :

M Rabah MELLAH

M.C.A, UMMTO, Président

M Mohand Outahar BENSIDHOUM

M.C.A, UMMTO, Encadreur

M Amar HAMACHE

M.C.B, UMMTO, Examineur

M Mohamed ALI BEY

M.A.B, UMMTO, Examineur

Remerciements

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

Nous tenons à remercier notre promoteur, Monsieur Bensidhoum. M.O, pour sa disponibilité, son orientation et Son soutien moral tout au long de notre travail.

Nos profondes gratitudes, s'adressent à Monsieur MELLAH. R qui a accepté de présider ce jury.

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.

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.

Enfin, nous remercions tous ceux qui ont participé de près ou de loin à la réalisation de ce Mémoire.

Dédicaces

Je dédie ce modeste travail:

A Mes très chers parents pour leur bienveillance et leur soutien tout au long de mon parcours scolaire et universitaire.

A Mes très chers frères Mokrane, Kamel, Achour, Mustapha.

A Ma précieuse sœur Assia et son fils Abderehmane, ainsi que ma tante Ouzna.

A Mon enseignant et ma source d'inspiration « Chikh Nacer » ainsi que toutes sa famille.

A L'ensemble des membres et adhérents de l'association culturelle Igawawen de L.N.I avec qui j'ai partagé une expérience richissime.

A mon ami Hamoudi Ahcene avec qui j'ai partagé ce modeste travail.

A mon ami Belkalem Jugurta pour sa disponibilité et son orientation tout au long de notre travail.

A Salmi Moussa et Hafid Salim pour leur soutiens et aide précieuse.

A toutes la promotion automatique 2016.

Karim.

Dédicaces

Je dédie ce modeste travail:

A Mes parents pour leur bienveillance et leur soutien tout au long de mon parcours scolaire et universitaire.

A Mon petit frère Sofodi.

A Mon amie Arbane Dehbia, pour son soutien, et ses encouragements.

A Mon ami Berkani Karim avec qui j'ai partagé ce travail.

A Belkalem Jugurta pour sa disponibilité et son aide tout au long de notre travail.

A Hafid Salim pour son soutien et son orientation précoce.

Ainsi qu'à tous mes amis et proches.

A toute la promotion automatique 2016.

Ahcene.

Liste des abréviations

ISO : Organisation internationale de normalisation.

IEEE : Institute of Electrical and Electronics Engineers.

Wifi : Wireless Fidelity.

TCP : Transmission Control Protocol.

LAN : Local Area Network.

WAN : Wide Area Network.

Hiper LAN : High Performance Radio Local Area Network.

WPAN : Wireless Personal Area Networks.

IdO : Internet of Things (Internet des objets).

IP : Internet Protocol.

LISTE DES FIGURES

Figure 1 : Robot industrie.....	7
Figure 2 : Robot Curiosity de la NASA.....	8
Figure 3 : Schéma d'un robot de type unicycle.....	13
Figure 4 : Schéma d'un robot de type tricycle.....	14
Figure 5 : Schéma d'un robot de type voiture.....	14
Figure 6 : Google Self-Driving Car Project.....	15
Figure 7 : Schéma d'un robot de type omnidirectionnel.....	16
Figure 8 : Robot mobile à chenilles.....	17
Figure 9 : Robot ASIMO 2000 conçu par HONDA.....	17
Figure 10 : Robot Cheetah de Boston Dynamics.....	18
Figure 11 : Robot Octopodes T8 de Robugtix.....	18
Figure 12 : Robot rampant développer par l'université de Carnegie Mellon.....	19
Figure 13 : Principe de fonctionnement d'un capteur.....	20
Figure 14 : Courbe d'étalonnage d'un capteur.....	21
Figure 15 : Architecture d'un robot.....	22
Figure 16 : Télémètres infrarouges.....	26
Figure 17 : Principe de fonctionnement d'un capteur ultrasonore.....	27
Figure 18 : Obstacle perpendiculaire à l'axe du capteur.....	28
Figure 19 : Obstacle en biais (il ne sera pas détecté par le capteur).....	28
Figure 20 : Obstacle en biais (il sera détecté par le capteur).....	29
Figure 21 : Principe de la stéréovision.....	30
Figure 22 : Moteur à aimants permanent.....	33
Figure 23 : Moteur à réluctance variable.....	34
Figure 24 : Moteur hybride.....	34
Figure 25 : Un servomoteur.....	35
Figure 26 : Architecture comportementale.....	37
Figure 27 : Architecture hybride.....	37
Figure 28 : Le cycle perception/décision/action que doit réaliser un robot.....	38
Figure 29 : Capteur à ultrason HC-SR04.....	42
Figure 30 : Branchement du capteur à ultrason.....	43
Figure 31 : Train d'impulsions émissent par le capteur.....	44
Figure 32 : Principe de fonctionnement d'un capteur ultrasonique.....	44
Figure 33 : Angle d'ouverture du capteur HC-SR04.....	45

LISTE DES FIGURES

Figure 34 : Moteur à courant continu avec balais.....	46
Figure 35 : Vue interne d'un moteur à courant continu.....	47
Figure 36 : Moteur à courant continu sans balais.....	48
Figure 37 : Réducteurs à dentures droite et hélicoïdale.....	49
Figure 38 : Réducteurs Maxon compacts à dentures droite.....	50
Figure 39 : Schéma de fonctionnement d'un réducteur.....	50
Figure 40 : Montage de commande d'un moteur via la carte Arduino.....	51
Figure 41 : Pont en H.....	52
Figure 42 : Illustration la manière dont il faut alimenter le pont en H pour faire tourner le moteur dans les deux sens.....	52
Figure 43 : Montage pont en H diode de roue libre.....	53
Figure 44 : Synoptique de fonctionnement de l'asservissement du servomoteur	54
Figure 45 : Vue intérieur d'un servomoteur.....	54
Figure 46 : La variation du poids que peut supporter un servomoteur SG90.....	56
Figure 47 : Commande de la position du servomoteur.....	56
Figure 48 : Connexion du servomoteur à l'Arduino.....	57
Figure 49 : Carte Arduino UNO Rev 3.....	58
Figure 50 : Schéma explicite de la carte Arduino Uno Rev 3.....	60
Figure 51 : Brochage du microcontrôleur ATmega 328.....	65
Figure 52 : Schéma synoptique du microcontrôleur ATmega 328.....	67
Figure 53 : Schéma synoptique de l'architecture AVR.....	68
Figure 54 : Motor shield Arduino.....	69
Figure 55 : Module ESP8266-07.....	72
Figure 56 : Broches du module Wifi ESP8266.....	74
Figure 57 : Architecture de notre robot mobile.....	78
Figure 58 : La partie responsable de la perception du robot.....	79
Figure 59 : Connecteur du servomoteur.....	79
Figure 60 : Les différentes positions de la partie perception de notre robot.....	80
Figure 61 : Motoréducteur 25 GA 370.....	81
Figure 62 : Brochages du circuit intégré L298P.....	82
Figure 63 : Contrôle de la vitesse via la PWM.....	84
Figure 64 : Schéma synoptique du circuit intégré L298P.....	85
Figure 65 : Shield moteur Arduino.....	87
Figure 66 : Shield moteur monté sur une carte Arduino Uno.....	88

LISTE DES FIGURES

Figure 67 : Branchement des motoréducteurs vers le shield moteur.....	89
Figure 68 : Branchement de l'ESP8266 à l'Arduino.....	90
Figure 69 : Câble USB (fiche A vers B).....	91
Figure 70 : Traduction d'un programme informatique en langage machine.....	91
Figure 71 : Interface de l'environnement de développement Arduino IDE.....	92
Figure 72 : Les icônes de l'IDE.....	93
Figure 73 : Algorithme de fonctionnement du robot.....	95
Figure 74 : Algorithme de téléguidage via Wifi.....	96
Figure 75 : Photos de notre robot.....	107
Figure 76 : Schéma de la carte Arduino Uno Rev 3.....	108
Figure 77: Schéma de l'Arduino Motor Shield.....	109

LISTE DES TABLEAUX

Tableau 1 : Différents types de robot à roues.....	16
Tableau 2 : Comparatifs des moteurs pas à pas.....	35
Tableau 3: Spécifications et limites du HC-SR04.....	43
Tableau 4 : Références hardware de l'Arduino Uno Rev3.....	61
Tableau 5 : Les différents protocoles de communication.....	70
Tableau 6 : Les normes wifi de l'ESP8266.....	72
Tableau 7 : Propriétés de l'ESP8266.....	74
Tableau 8 : Description des Pins de l'ESP.....	75
Tableau 9 : Les Pins du double pont L298P.....	83
Tableau 10 : Variation de la PWM.....	84
Tableau 11 : Branchement du pont A.....	85
Tableau 12 : Branchement du pont B.....	86
Tableau 13 : Liaison de l'Arduino Uno au Shield moteur.....	88

Introduction générale	1
Chapitre I : Etat de l'art de la robotique	4
I. La robotique	5
I.1 Introduction	5
I.1.1 Définition de la Robotique.....	5
I.1.2 Etymologie du terme robot.....	5
I.1.3 Un robot.....	5
I.1.4 Les origines de la robotique.....	5
I.1.5 Les dates marquantes de la robotique.....	6
I.2 Application de la robotique	6
I.2.1 La robotique d'intervention.....	6
I.2.2 La robotique de service professionnel.....	7
I.2.3 La robotique personnelle.....	7
I.3 Les types de robots	7
I.3.1 Les robots humanoïdes.....	7
I.3.2 Les robots industriels (manipulateurs).....	7
I.3.3 Les robots explorateurs (mobiles).....	8
I.4 Conclusion	9
II. La robotique mobile	10
II.1 Introduction	10
II.1.1 Présentation du robot mobile.....	11
II.2 Classification des robots mobiles	11
II.2.1 Classification selon le degré d'autonomie	11
II.2.1.1 Robot télécommandé.....	11
II.2.1.2 Robot semi-autonome.....	11
II.2.1.3 Robot autonome.....	12
II.2.2 Classification selon le domaine d'application	12
II.2.2.1 Les robots industriels et de service.....	12
II.2.2.2 Les robots militaires.....	12
II.2.2.3 Les robots de laboratoire.....	12
II.2.3 Classification selon le type de locomotion	13
II.2.3.1 Les robots mobiles à roues.....	13
II.2.3.2 Robots unicycle.....	13

II.2.3.3 Robots tricycle.....	13
II.2.3.4 Robots voiture.....	14
II.2.3.5 Robots mobiles omnidirectionnels.....	15
II.2.3.6 Les avantages et les inconvénients des différents types de robots à roues.....	16
II.2.3.7 Les robots mobiles à chenilles.....	16
II.2.3.8 Les robots humanoïdes.....	17
II.2.3.9 Les robots mobiles rampants.....	19
II.3 Notion de Perception.....	20
II.3.1 Les capteurs en robotique mobile.....	20
II.3.1.1 Capteurs intéroceptifs.....	22
II.3.1.1.1 Les capteurs de déplacement.....	23
II.3.1.1.1.1 Les odomètres.....	23
II.3.1.1.1.2 Les accéléromètres.....	23
II.3.1.1.1.3 Les capteurs d'attitude.....	24
II.3.1.1.1.4 Les gyroscopes.....	24
II.3.1.1.1.5 Les gyromètres.....	24
II.3.1.1.1.6 Les gyrocompas.....	24
II.3.1.1.1.7 Le magnétomètre.....	24
II.3.1.2 Capteurs extéroceptifs.....	25
II.3.1.2.1 Les capteurs télémétriques.....	25
II.3.1.2.2 Les capteurs lasers.....	25
II.3.1.2.3 Les capteurs infrarouges.....	26
II.3.1.2.4 Les capteurs ultrasonores.....	27
II.3.2 Les systèmes de vision.....	29
II.3.2.1 La stéréovision.....	30
II.3.2.2 Les capteurs à triangulation active.....	30
II.3.2.3 La vision omnidirectionnelle.....	31
II.4 La locomotion.....	31
II.4.1 Les actionneurs.....	31
II.4.1.1 Les moteurs à courant continu.....	32
II.4.1.2 Les moteurs pas à pas.....	32
II.4.1.2.1 Le moteur à aimants permanents.....	33
II.4.1.2.2 Le moteur à réluctance variable.....	33

II.4.1.2.3 Le moteur hybride.....	34
II.4.1.2.4 Comparaison des différents types de moteurs pas à pas.....	35
II.4.1.3 Les servomoteurs.....	35
II.5 Les Architectures décisionnelles d'un robot mobile.....	36
II.5.1 Architecture délibérative.....	36
II.5.2 Architecture comportementale.....	36
II.5.3 Architecture hybride.....	37
II.6 La navigation.....	38
II.7 Conclusion.....	39
Chapitre II : Conception d'un robot mobile.....	40
I. Introduction.....	41
I.1 Structure de notre robot.....	41
I.2 La conception matérielle de notre robot.....	42
I.2.1 Le capteur à ultrason HC-SR04.....	42
I.2.1.1 Caractéristiques et spécifications.....	42
I.2.1.2 Broches de connexion.....	42
I.2.1.3 Spécifications et limites.....	43
I.2.1.4 Le branchement.....	43
I.2.1.5 Principe de fonctionnement.....	43
I.2.1.6 Considérations pratique.....	44
I.2.1.7 Distance de l'objet.....	45
I.2.2 Les motoréducteurs.....	45
I.2.2.1 Les moteurs à courant continu.....	45
I.2.2.1.1 Les moteurs à courant continu avec balais.....	46
I.2.2.1.1.2 Principe de fonctionnement.....	46
I.2.2.1.2 Les moteurs à courant continu sans balais.....	47
I.2.2.1.2.1 Principe de fonctionnement.....	47
I.2.2.2 La mécanique liée au moteur.....	48
I.2.2.2.1 Le couple.....	48
I.2.2.2.2 La vitesse de rotation.....	49
I.2.2.3 Les réducteurs.....	49
1.2.2.4 Commande des moteurs à courant continu.....	51
I.2.2.5 Les protections nécessaires.....	53

I.2.2.5.1 Les diodes de roue libre.....	53
I.2.3 Les servomoteurs.....	53
I.2.3.1 Définition et principe de fonctionnement.....	53
I.2.3.2 La mécanique.....	54
I.2.3.3 La connectique.....	56
I.2.3.4 Le signal de commande des servomoteurs.....	56
I.2.3.5 La librairie Servo.....	57
I.2.4 La conception informatique de notre robot.....	58
I.2.4.1 Arduino.....	58
I.2.4.2 Une carte électronique.....	58
I.2.4.3 La carte Arduino Uno.....	58
I.2.4.3.1 Caractéristiques.....	59
I.2.4.3.2 Alimentation de la carte Arduino Uno.....	61
I.2.4.3.3 Les broche de l'Arduino.....	62
I.2.4.3.3.1 Les broche d'alimentation.....	62
I.2.4.3.3.2 Les broche d'entrées/sorties.....	62
I.2.4.3.4 La communication avec l'extérieur.....	63
I.2.4.3.5 Gestion d'interruptions.....	64
I.2.4.3.6 Timers du microcontrôleur de l'Arduino.....	64
I.2.4.3.7 Le microcontrôleur ATmega 328.....	64
I.2.5 Les Shields Arduino.....	69
I.2.5.1 Le Motor Shield.....	69
I.2.6 La communication Wifi.....	69
I.2.6.1 Les protocoles de communication.....	69
I.2.6.2 Téléguidage Wifi.....	70
I.2.6.3 La technologie wifi.....	70
I.2.6.4 Les normes Wifi.....	71
I.2.7 Le module ESP8266-07.....	72
I.2.7.1 Caractéristiques de l'ESP8266.....	73
I.2.7.2 Les broche de l'ESP8266.....	74
I.2.8 Conclusion.....	76

Chapitre III : Réalisation d'un robot mobile	77
I. Introduction	78
I.1 Architecture du robot mobile	78
I.2 La réalisation matérielle	78
I.2.1 La perception	78
I.2.2 La locomotion	80
I.2.2.1 Les motoréducteurs	80
I.2.2.2 La commande des motoréducteurs	81
I.2.2.3 Le pont H double L298P	81
I.2.2.4 Les PWM	84
I.2.2.5 L'Arduino Motor Shield	86
I.2.3 La communication via Wifi	89
I.2.3.1 Fonctionnement du module	90
I.3 La réalisation logiciel	91
I.3.1 La programmation de l'Arduino	91
I.3.2 Un compilateur	91
I.3.3 Présentation du logiciel	92
I.3.3.1 Correspondance	92
I.3.3.2 Fonctions des icônes de la barre de boutons	93
I.3.4 Programmation sous l'IDE d'Arduino	93
I.4 L'Algorithme de fonctionnement du robot	94
I.4.1 Algorithme du sous-programme de commande par wifi	96
I.5 Conclusion	97
Conclusion Générale	98
Annexes	100
Annexe A	101
Le programme Arduino de notre robot	102
Annexe B	106
Bibliographie	110

INTRODUCTION GÉNÉRALE

Introduction générale :

Depuis bien longtemps, l'Homme rêve de créer des machines intelligentes capables d'effectuer des tâches à sa place. Ainsi les humains auraient plus de temps à consacrer pour les loisirs, ou prendraient moins de risque pour effectuer des tâches dangereuses dans des endroits inaccessibles et hostiles. La robotique est l'ensemble des techniques permettant la conception et la réalisation de ces machines nommé « robot ». Un robot est un appareil effectuant, grâce à un système de commande automatique à microprocesseur, une tâche précise pour laquelle il a été conçu que ce soit dans le domaine industriel, domestique ou scientifique.

Durant cette dernière décennie, on constate qu'un intérêt croissant est porté au sein de la communauté scientifique, au développement des systèmes intelligents autonomes et télé-opérés dans le cadre de la robotique mobile. Un tel intérêt peut être perçu comme une conséquence logique de l'apparition des applications potentielles des machines intelligentes (industrielle, services, manutention, ou encore aide à la mobilité,...). L'objectif est de mettre les robots dans des tâches variées telles les interventions sur des sites accidentés, les manipulations dans des sites sensibles, ainsi que l'exploration spatiale, etc.

L'enjeu principal de la robotique mobile actuelle consiste à développer des systèmes de navigation intelligents, où la navigation autonome, est un axe de recherche qui vise à donner à une machine la capacité de se mouvoir dans un environnement sans assistance, ni intervention humaine pour accomplir un but désiré. La fonction de navigation consiste à donner au robot la possibilité d'obtenir les informations dont il a besoin pour prendre des décisions de manière complètement autonome, ainsi que de le doter de moyens de locomotion adaptés à son environnement. Car si on prend l'exemple des robots explorateurs de planètes, tel le robot Curiosity rover [1] qui explore la planète Mars, il est impossible de contrôler en temps réel un tel robot qui se trouve à une distance de 225 300 000 km de la terre, on ne peut pas négliger le retard que mettrait un signal de commande pour arriver jusqu'à lui, sans parler des bruits (radiations gamma) qui peuvent perturber le signal, c'est pour cela que la capacité d'un robot à prendre lui-même les décisions adéquates est primordiale pour qu'il puisse réaliser et coordonner des missions complexes. Le robot doit sélectionner les meilleures actions à effectuer pour mener à bien sa mission.

Notre travail a pour but la conception et la réalisation d'un robot mobile autonome avec la possibilité d'être télécommandé à distance par Wifi. Le robot sera basé sur une carte Arduino Uno comme composant principal ainsi que différents organes de perception (capteurs), de locomotion (moteurs), ainsi qu'un moyen de communication via Wifi qui est une méthode de contrôle et de feedback à distance.

Notre travail se présente comme suit :

Chapitre I : Etat de l'art de la robotique : dans ce chapitre nous nous intéresserons en un premier lieu à la présentation de la robotique en générale, puis à la robotique mobile de manière plus approfondie.

Chapitre II : Conception d'un robot mobile : dans ce chapitre nous ferons une étude des différents éléments qui constitueront notre robot, nous montrerons également la manière pour les utilisés.

Chapitre III : Réalisation d'un robot mobile : dans ce chapitre nous mettrons en œuvre la réalisation pratique de notre robot, afin que celui-ci réponde au problème de déplacement autonome ainsi que la réalisation de la commande Wifi.

CHAPITRE I : ETAT DE L'ART DE LA ROBOTIQUE

I. La robotique :

I.1 Introduction :

La robotique est l'art d'automatiser des systèmes plus ou moins complexes en s'appuyant sur le savoir-faire acquis à travers les études sur la conception de robots. Elle est aussi décrite comme étant l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots. Ces derniers sont des appareils effectuant, grâce à un système de commande automatique à base de micro-processeur, une tâche précise pour laquelle ils ont été conçus.

La robotique est aujourd'hui primordiale, voire même vitale car elle est l'une des raisons principales qui donne au monde actuel son aspect moderne en se manifestant dans différents champs d'application. On cite au passage, la robotique médicale qui est utilisée dans le cadre d'une application thérapeutique, par exemple lors d'une chirurgie ou au cours d'un programme de réhabilitation neuro-motrice, comme on trouve aussi la robotique industrielle, militaire, et domestique.

I.1.1 Définition de la Robotique :

La robotique est la science qui étudie l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots. C'est un domaine multidisciplinaire ; on y trouve des aspects concernant la mécanique, l'informatique, ainsi que l'électronique.

I.1.2 Etymologie du terme robot :

Le terme « robot » découle de la langue slave et signifie textuellement « esclave » ou bien « Robota » qui signifie travail forcé.

La définition précise d'un robot et en contrepoint de la robotique, est un assemblage complexe de pièces mécaniques et électroniques, le tout piloté via une intelligence artificielle.

I.1.3 Un robot :

Un robot est un dispositif mécanique (alliant mécanique, électronique et informatique) accomplissant automatiquement soit des tâches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains, soit des tâches plus simples mais en les réalisant mieux que ce que ferait un être humain.

Un robot est défini aussi comme un appareil effectuant grâce à un système de commande automatique à base de micro-processeur une action précise pour laquelle il a été conçu dans le domaine industriel, scientifique ou domestique.

I.1.4 Les origines de la robotique :

Les ancêtres des robots sont les automates. Un automate très évolué fut présenté par Jacques de Vaucanson en 1738 : il représentait un homme jouant d'un instrument de musique à vent. Jacques de Vaucanson créa également un automate représentant un canard mangeant et refoulant sa nourriture après ingestion de cette dernière.

Le mot « robot » quant à lui est apparu pour la première fois en 1920 dans une pièce de théâtre de Karel Capek de science-fiction R.U.R (Rossum's Universal Robots) il est inventé par son frère Josef à partir du mot Tchèque « Robota ».

I.1.5 Les dates marquantes de la robotique

- 1774 : Les trois automates de Pierre Jacquet-Droz.
- 1801 : Le métier à tisser Jacquart.
- 1920 : Invention du mot "robot" (pièce de théâtre de Karel Capek).
- 1946 : ENIAC, le premier ordinateur.
- 1950 : Les tortues de Grey Walter, premiers robots mobiles autonomes.
- 1950 : Les robots recueils de nouvelles d'Issac Asimov (les trois lois de la robotique)
- 1956 : Conférence sur l'intelligence artificielle.
- 1961 : Unimate, premier robot industriel aux Etats-Unis.
- 1967 : Shaley, premier robot mobile contrôlé par ordinateur, Stanford Research Institute.
- 1973 : Wabot-1, premier robot humanoïde à marche bipède, Université de Waseda.
- 1992 : Robodoc, premier robot pour une opération de la hanche (Etats-Unis).
- 1993: Cog, Humanoid Robotics Group, MIT (Massachusetts Institute of Technology- Etats Unis).
- 1999 : AIBO-ERS- 110, Sony.
- 2000 : Asimo, Honda.
- 2003 : Les Rovers Spirit et Opportunity sur Mars, NASA.
- 2004 : Le robot serpent amphibie (Japon).
- 2005 : AIBO Mind 3-ERS-7, Sony.
- 2006 : Lego Mindstorm RCX.
- 2009 : Lego Mindstorm NXT.
- 2012 : Le rover Curiosity sur Mars, NASA.

I.2 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.

I.2.1 La robotique d'intervention :

La robotique d'intervention est caractérisée en général par des robots télé opérés par un opérateur à travers des commandes directes (joysticks, bras maître, des organes de commandes physique ou virtuels). L'opérateur est donc systématiquement présent dans la boucle de commande du robot, afin de l'aider à réaliser sa fonction.

Cette forme de robotique est généralement utilisée pour remplir des tâches dans des environnements difficiles d'accès ou hostiles (industries nucléaire, intervention sur des catastrophes naturelles, ainsi que l'exploration marine ou spatiale...).

I.2.2 La robotique de service professionnel :

Cette catégorie est souvent utilisée dans l'industrie professionnelle afin d'accomplir des tâches répétitives et/ou dangereuses, cette robotique requière un niveau de précision et de qualité inaccessibles à un opérateur humain d'où ça grande utilité.

I.2.3 La robotique personnelle :

Elle constitue un champ potentiellement aussi foisonnant que la robotique de service professionnel. Son utilisation est capable de répondre à un grand nombre de besoins telle la réalisation de tâches domestiques, ainsi que l'assistance aux personnes en perte d'autonomie, etc.

I.3 Les type de robots :

La classification des robots est faite de nos jours dans un objectif de donner une idée de la présence actuelle de la robotique et de la portée de ce domaine dans les années à venir, ainsi nous pouvons convenir qu'il existe trois grandes catégories de robots.

I.3.1 Les robots humanoïdes :

C'est sans doute la catégorie de robot la plus connue, ceci est dû en grande partie à la promotion faite pas la science-fiction, elle regroupe ainsi tous les robots anthropomorphes, ceux dont la forme rappelle la morphologie humaine. La ressemblance avec des êtres humains est tellement forte parfois qu'au Japon un robot humanoïde a présenté le journal télévisé du 18 juin 2014. Ce type de robot est communément appelé Androïde.

I.3.2 Les robots industriels (manipulateurs) :

La majorité de ces robots sont à base fixe. Quand ce n'est pas le cas, ils sont généralement montés sur rail. On trouve dans cette catégorie les robots de manipulation, type « Pick And Place », les robots soudeurs ou encore les robots peintres utilisés dans l'industrie automobile. Ils représentent actuellement la majorité des robots.



Figure 1 : robot industriel.

I.3.3 Les robots explorateurs (mobiles) :

De façon générale, cette catégorie englobe aussi bien les robots humanoïdes ayant la faculté de se mouvoir dans leur environnement, que les robots mobiles à roues, ainsi que tous les autres robots capable de déplacement. Les robots mobiles à roues UGV, en anglais (Unmanned Ground Vehicles), regroupe les robots à base actionnée par des roues ou par des chenilles.

Ils sont généralement utilisés pour l'exploration, c'est pour cette raison qu'ils sont aussi appelés des rovers (vagabonds). Les plus connus sont les robots de la NASA (Curiosity, Mars Rovers, etc.). (Voir la figure 2) présente le robot Curiosity qui a été envoyé sur Mars à des fins d'exploration et d'identification du terrain martien.



Figure 2 : Robot Curiosity de la NASA

Les domaines d'application des robots mobiles sont très variés : le transport de courriers et de marchandises, l'exploration en environnement hostile, le déminage, ainsi que d'autres domaines civil.

I.4 Conclusion :

Dans cette brève partie de notre travail, nous avons défini certaines notions fondamentales liées à la robotique en général.

Un robot est en fait une machine programmable mettant en jeu de nombreuses autres disciplines telles (la mécanique, l'électronique, et l'informatique) dans le but d'imiter les actions d'une créature intelligente.

L'homme a tenté de concevoir des machines qui puissent le remplacer, ces machines sont les robots. Avec le temps, ces machines ont évolué en une science appelée la robotique. Dans un passé proche, les robots servaient plus à aider les personnes, alors que de nos jours ils ont un rôle prépondérant dans la vie humaine, et ce sans même que nous nous en rendions compte. Mais rien de tout cela n'aurait été possible sans l'intelligence artificielle.

De nos jours la robotique se développe en deux grandes branches :

- La robotique manufacturière : utilisée comme outil de production. Ce domaine de la robotique regroupe tous les robots industriels.
- La robotique non-manufacturière : utilisée comme outil pour réaliser des tâches impossibles pour l'être humain. Ce domaine regroupe essentiellement les robots de télémanipulations, les robots d'exploration, les robots mobiles et les simulateurs.

II. La robotique mobile :

II.1 Introduction :

D'une manière générale, l'appellation robots mobiles regroupe l'ensemble des robots à base mobile, par opposition aux robots manipulateurs, et on désigne souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent décrits selon leur type de locomotion, qu'ils soient marcheurs, sous-marins ou encore aériens.

On peut estimer que les robots mobiles à roues constituent le gros des robots mobiles, c'est pour cela qu'on s'intéressera volontairement à une présentation approfondie des robots mobiles à roues et aux problèmes associés à leur déplacement autonome.

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 encore en milieu hostile (nucléaire, marin, spatiale, lutte contre les incendies, surveillance, etc.). L'aspect particulier de la mobilité impose une complexité technologique et méthodologique qui s'ajoute en générale aux problèmes rencontrés par les robots manipulateurs. La résolution de ces problèmes passe par l'emploi de toutes les ressources disponibles tant au niveau technologique (capteurs, motricité, énergie) qu'à celui du traitement des informations par l'utilisation des techniques d'intelligence artificielle.

L'autonomie du robot mobile est une faculté qui lui permet de s'adapter ou de prendre des décisions dans le but de réaliser une tâche malgré un manque d'informations préliminaires ou éventuellement erronées. Dans d'autres cas d'utilisation, comme celui des véhicules d'exploration spatiale, comme le robot Curiosity rover [1] qui est en phase d'exploration de la planète Mars actuellement. L'autonomie est un point fondamental puisque la télécommande est dans ce cas-là impossible par le fait de la durée de temps de transmission des informations.

Afin d'être autonome, un robot mobile doit posséder de nombreuses capacités. Premièrement, il doit être capable de percevoir son environnement et de se localiser dans celui-ci. Pour ce faire, un robot possède une panoplie de capteurs en tous genres, comme des sonars et des dispositifs à balayage laser qui servent à mesurer des distances entre le robot et les obstacles à proximité.

Une fois localisé dans son environnement, le robot doit être capable de se déplacer d'un point à un autre en trouvant des chemins efficaces et sécurisés afin d'éviter d'éventuelles collisions avec des obstacles. De plus, un robot est souvent appelé à communiquer avec les agents situés à proximité. Cela peut être fait de différentes façons, comme par communication vocal ou depuis une interface graphique.

II.1.1 Présentation du robot mobile :

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 est capable de s'adapter à certaines variations de ses conditions de fonctionnement. Elle est dotée de fonction de perception, de décision et d'action, lui permettant ainsi d'effectuer des tâches diverses et variées, et ce de plusieurs manières, même en cas de situations inattendues.

Le robot mobile est doté de moyens qui lui permettent de se déplacer dans son espace de travail. Suivant son degré d'autonomie ou d'intelligence, il peut être doté de moyens de perception et de raisonnement. Certains sont capables, sous contrôle humain réduit, de modéliser leur espace de travail et de planifier un chemin dans un environnement qu'ils ne connaissent pas forcément d'avance. Actuellement, les robots mobiles les plus sophistiqués sont essentiellement orientés vers des applications dans des environnements variables ou incertains, souvent infestés d'obstacles, nécessitant une adaptabilité à la tâche.

II.2 Classification des robots mobiles :

La classification des robots se fait selon plusieurs critères, trois principalement [1] :

- Le degré d'autonomie.
- Le domaine d'application.
- Le système de locomotion.

Il est important de noter que l'expression « robots mobiles » bien que désignant l'ensemble des robots à base mobile (par opposition aux robots manipulateurs), est généralement employé afin de désigner les robots mobiles terrestres.

II.2.1 Classification selon le degré d'autonomie :

Un robot autonome est un système doté de capacités décisionnelles et de moyens d'acquisition et de traitements de données lui permettant d'accomplir sous contrôle humain réduit voire même absent un certain nombre de tâches, dans un environnement connu ou inconnu. Selon ce critère on peut classer les robots mobiles de la manière suivante :

II.2.1.1 Robot télécommandé :

Ce sont des robots commandé par un opérateur (machines ou être humain), qui leurs dicte chaque tâche élémentaire à accomplir (avancer, reculer, tourner, etc.).

II.2.1.2 Robot semi-autonome :

Ces types de robots effectuent un certain nombre d'opérations par eux même d'une façon complètement autonome mais il peut être interrompu à tous moment pour recevoir de nouvelles commandes dictées par un opérateur.

II.2.1.3 Robot autonome :

On considère qu'un robot est complètement autonome s'il est capable d'adapter son comportement à l'environnement dans lequel il évolue.

L'autonomie est la capacité propre d'un système sans équipage, à capter, percevoir, analyser, communiquer, planifier, et prendre des décisions et agir afin d'atteindre les buts qui lui ont été assignés lors de sa conception.

II.2.2 Classification selon le domaine d'application :

L'un des plus grands avantages des robots mobiles est le fait que leur domaine d'application est illimité, c'est pour cela que nous présentons ici quelques domaines d'application :

II.2.2.1 Les robots industriels et de service :

La robotique industrielle est officiellement définie par l'ISO comme un contrôle automatique, reprogrammable, polyvalent manipulateur programmable dans trois ou plusieurs axes.

Les applications typiques incluent des robots de soudage, de peinture et d'assemblage. La robotique industrielle inspecte les produits, rapidement et précisément.

Les robots industriels sont très utilisés en automobile, leur conception nécessite une bonne connaissance et un très haut niveau dans le domaine de l'ingénierie.

Quant aux robots de service, ils sont destinés à aider des handicapés moteurs dans leur vie quotidienne, à guider les aveugles ainsi qu'à piloter des voitures automatiques.

II.2.2.2 Les robots militaires :

Un robot militaire, aussi appelé arme autonome, est un robot autonome contrôlé à distance, conçu pour des applications militaires. Les drones sont une sous-classe des robots militaires.

Les applications militaires de la robotique mobile sont nombreuses. Ce champ d'application présente l'intérêt de fournir des spécifications accrues telles la vitesse des véhicules, leurs capacités de franchissement d'obstacles (la robustesse de ce type de robots est essentielle), ainsi que leur rapidité de réaction en font des robots de très hautes performances.

II.2.2.3 Les robots de laboratoire :

Afin de valider des travaux théoriques sur la perception et la planification de mouvements, de nombreux laboratoires travaillent dans le domaine de la robotique.

La robotique de laboratoire est l'utilisation de robots dans les laboratoires de biologie ou de chimie. Par exemple, les sociétés pharmaceutiques utilisent des robots pour la synthèse de nouvelles entités chimiques ou pharmaceutiques afin de tester la valeur des matières chimiques.

II.2.3 Classification selon le type de locomotion :

II.2.3.1 Les robots mobiles à roues :

La mobilité par roues est la structure mécanique la plus utilisée en robotique mobile. Ce type de robot assure un déplacement aisé, mais nécessite un sol relativement plat. On distingue plusieurs classes de robots à roues, déterminées principalement par la position et le nombre de roues utilisées. Nous citerons ici les classes principales de robots à roues.

II.2.3.2 Robots unicycle :

Un robot de type unicycle est actionné soit par une seule roue ou par deux roues indépendantes, on utilise des capteurs d'attitude (gyroscope) et de déplacement (accéléromètre) pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. Le schéma suivant illustre un robot de type unicycle.

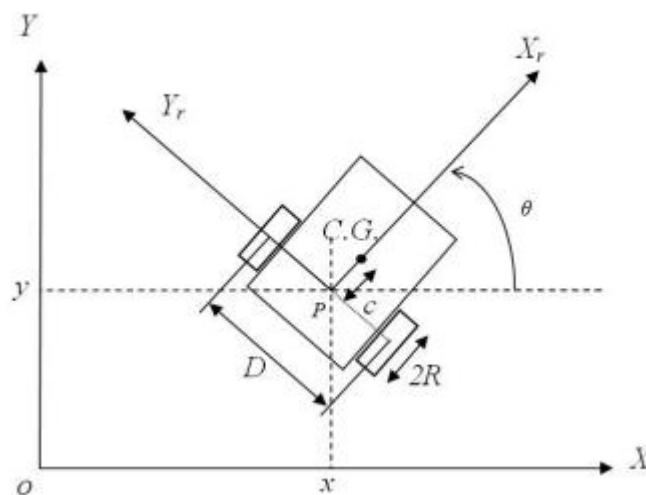


Figure 3 : schéma d'un robot de type unicycle

Un robot unicycle est défini comme étant un robot qui bouge dans un plan 2D ayant une certaine vitesse de déplacement vers l'avant, mais aucun mouvement latéral instantané, car les robots unicycles sont un système non-holonyme. Ils sont donc incapables d'avoir un déplacement dans une direction perpendiculaire aux roues de locomotion.

II.2.3.3 Robots tricycle :

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue libre (folle) centrée, orientable sur l'axe longitudinal du robot. Le mouvement du robot est alors donné par la vitesse des roues fixes et son orientation est assurée par la roue libre. Comme l'illustre la figure suivante :

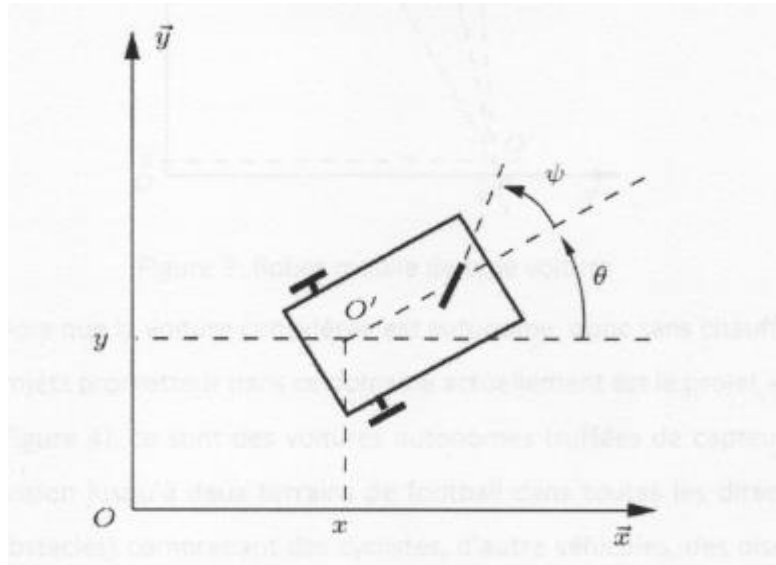


Figure 4 : schéma d'un robot de type tricycle

C'est là aussi un robot non-holonyme. En effet, il lui est impossible de se déplacer dans une direction perpendiculaire aux roues fixes.

II.2.3.4 Robots voiture :

Un robot de type voiture est semblable au robot tricycle, sauf qu'il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe. (Voir la figure 5). 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.

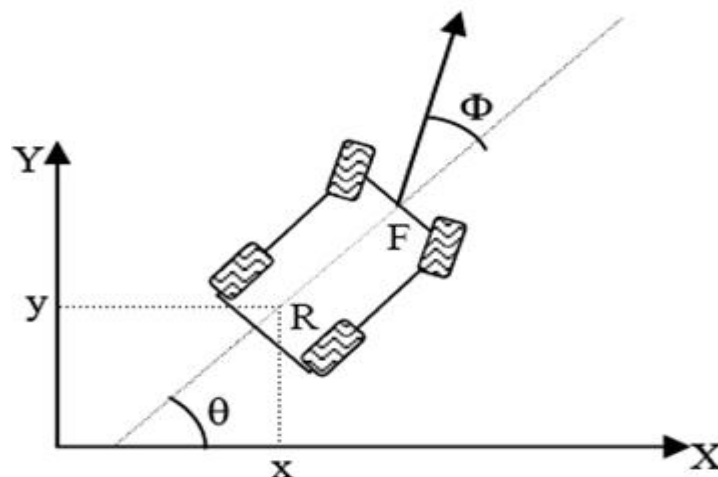


Figure 5 : schéma d'un robot de type voiture

L'un des plus grands projets dans ce domaine est le projet développé par « Google » le « Google Self-Driving Car Project », (voir la figure 6), ce sont des voitures complètement autonomes truffées de capteurs leur permettant d'avoir un champ de vision très étendu qui peut aller jusqu'à 800 m à la ronde. Ainsi elles peuvent détecter toute sorte d'objets

(obstacles) que ce soit des véhicules, des piétons, voire même des sachets en plastique. Ces informations aident grandement à une navigation sécurisée et pérenne et cela sans aucune intervention humaine.

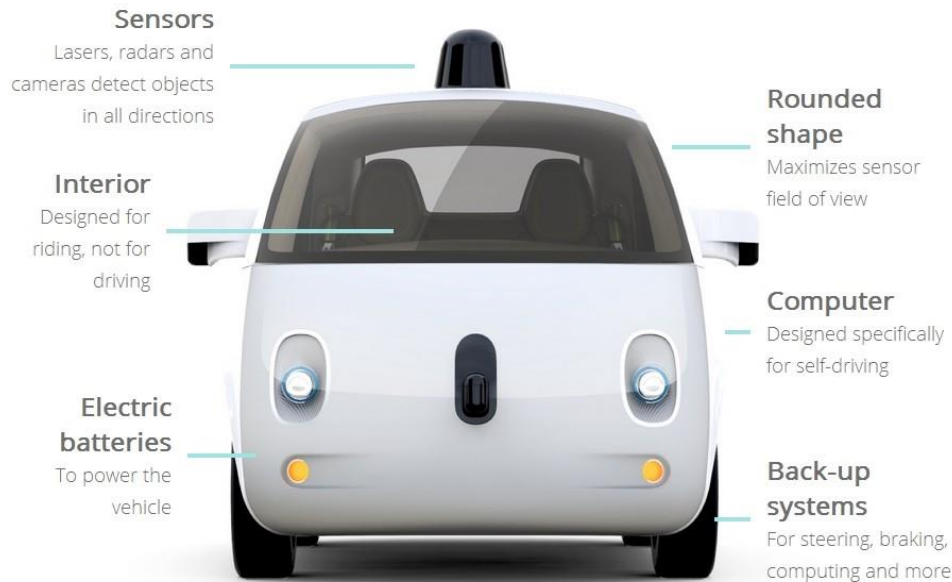


Figure 6: Google Self-Driving Car Project

II.2.3.5 Robots mobiles omnidirectionnels:

Un robot mobile est dit omnidirectionnel si l'on peut agir indépendamment sur la vitesse de translation selon les axes x et y et la vitesse de rotation autour de z (voir la figure 7). Il peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables qui sont placées de manière à former un triangle équilatéral.

L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer absolument dans toutes les directions. Mais ceci ne se fait qu'au dépend d'une complexité mécanique bien plus importante par rapport aux autres types de robots mobiles [1].

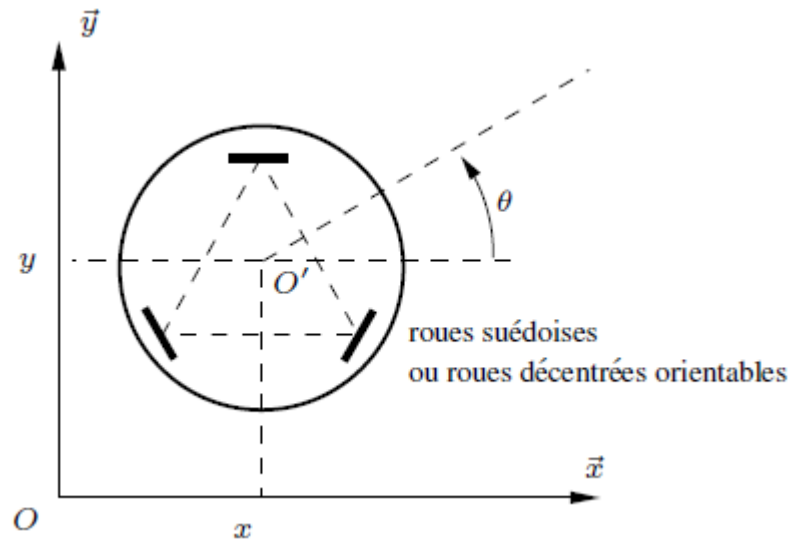


Figure 7 : schéma d'un robot de type omnidirectionnel.

II.2.3.6 Les avantages et les inconvénients des différents types de robots à roues :

Type du robot	Avantage	Inconvénient
Unicycle	* Stable * Rotation sur soi-même * Complexité mécanique faible	* Non-holonyme
Tricycle	* Complexité mécanique modérée	* Non-holonyme * Peu stable * Pas de rotation sur soi-même
Voiture	* Stable * Complexité mécanique modérée	* Non-holonyme * Pas de rotation sur soi-même
Omnidirectionnel	* Holonyme * Stable * Rotation sur soi-même	* Complexité mécanique importante

Tableau 1 : Différents types de robot à roues.

II.2.3.7 Les robots mobiles à chenilles :

Les robots mobiles à chenilles présentent l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation est orientée vers l'emploi sur sol accidenté

ou de mauvaise qualité au niveau de l'adhérence (présence de boue, herbe...). Généralement ils sont utilisés à des fins militaires, principalement comme robots de surveillance ou de déminage (voir figure 8).



Figure 8 : robot mobile à chenilles

II.2.3.8 Les robots humanoïdes :

Les robots humanoïdes (marcheurs) sont destinés à réaliser des tâches variées dont l'accès au site est difficile, dangereux ou impossible à un humain. Leur anatomie a de nombreux degrés de liberté, ce qui permet un rapprochement avec les robots manipulateurs. La locomotion est commandée via des articulations. Les méthodes de commande des articulations définissent le concept d'allure qui assure le déplacement de l'ensemble. Les différentes techniques étudiées se rapprochent de la marche des êtres humains. (Robot de type humanoïde voir figure 9). C'est des robots bipèdes qui se déplacent de façon complètement autonome.



Figure 9 : Robot ASIMO 2000 conçu par HONDA

Ainsi que même des animaux, dans cette catégorie nous mentionnons le robot Cheetah de Boston Dynamics qui a marqué l'histoire le 29 mai 2015, car pour la première fois un robot a réussi une succession de tests (sauts d'obstacles, course, etc.). Et cela de façon complètement autonome. (Voir figure 10).



Figure 10: robot Cheetah de Boston Dynamics

On trouve aussi des robots insectes (à quatre pattes, six pattes ou même plus) exemple le robot Octopodes T8 de Robugtix (voir figure 11). L'adaptation au support est un problème spécifique aux robots marcheurs, car la surface de contact entre le robot et le sol est très faible d'où la grande instabilité de ce type de robot. C'est pour remédier à cela que ces robots sont truffés de capteurs en tous genres (capteur de proximité, de contact, de vision, etc.).



Figure 11 : robot Octopodes T8 de Robugtix

II.2.3.9 Les robots mobiles rampants :

La reptation est une solution de locomotion pour un environnement de type « tunnel » ou « galerie » c'est ce qui a conduit à la réalisation des robots à structures filiformes (voir figure 12). Le système est composé d'un ensemble de module ayant chacun plusieurs mobilités. Ici les techniques utilisées découlent des méthodes de locomotion des serpents.

- Le type scolopendre constitue une structure inextensible articulée selon deux axes orthogonaux.
- Le type lombric comprend trois articulations, deux rotations orthogonales et une translation dans le sens du mouvement principal.
- Le type péristaltique consiste à réaliser un déplacement relatif d'un module par rapport aux autres modules voisins.

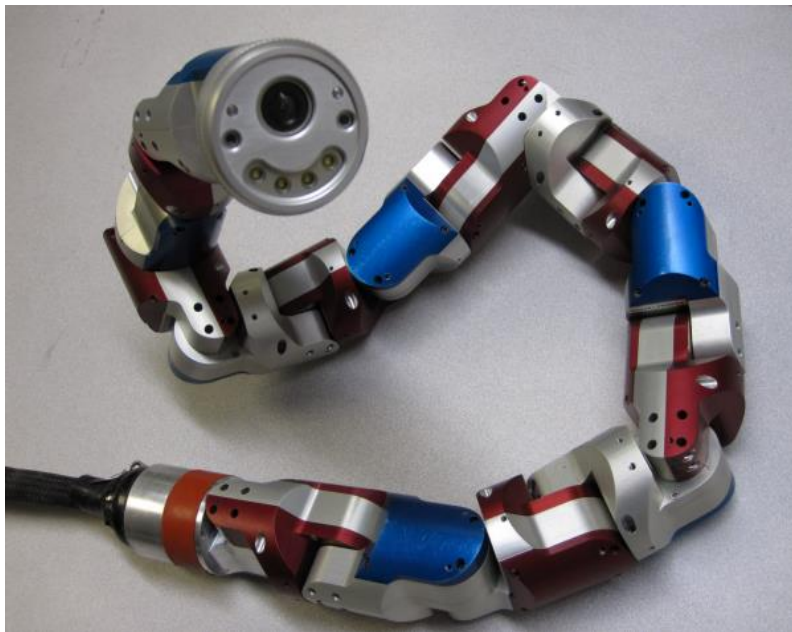


Figure 12 : robot rampant développé par l'université de Carnegie Mellon

II.3 Notion de Perception :

La notion de perception en robotique mobile est relative à la capacité du système à recueillir, traiter et mettre en formes des informations utiles au robot afin d'agir et réagir dans le monde qui l'entoure. Alors que pour des tâches de manipulation on peut considérer que l'environnement du robot est relativement structuré, ce n'est plus le cas lorsqu'il s'agit de naviguer de manière autonome dans des lieux partiellement connus. Aussi, pour extraire les informations utiles à l'accomplissement de sa tâche, il est nécessaire que le robot dispose de nombreux capteurs mesurant aussi bien son état interne que l'environnement dans lequel il évolue. Le choix des capteurs dépend bien évidemment de l'application envisagée.

II.3.1 Les capteurs en robotique mobile :

La robotisation des tâches s'oriente d'une manière générale vers une plus grande prise d'initiatives de la part du robot par rapport à un système automatisé. Le robot choisit ainsi lui-même sa stratégie d'action en fonction de la tâche à réaliser et en fonction de son environnement. Pour accomplir cette opération, le robot doit être doté de capteurs lui permettant d'appréhender en temps réel son environnement [3].

Le capteur est un instrument qui permet de détecter l'information contenue dans un objet ou issue d'un phénomène. Ce dispositif est soumis à l'action d'une mesurande non électrique, et fournit un signal électrique à sa sortie (voir figure 13). Il va assurer la mesure permanente des grandeurs réelles en question.



Figure 13 : principe de fonctionnement d'un capteur

Ainsi on obtient la grandeur électrique de sortie en fonction de la grandeur physique d'entrée comme représenté sur la figure suivante :

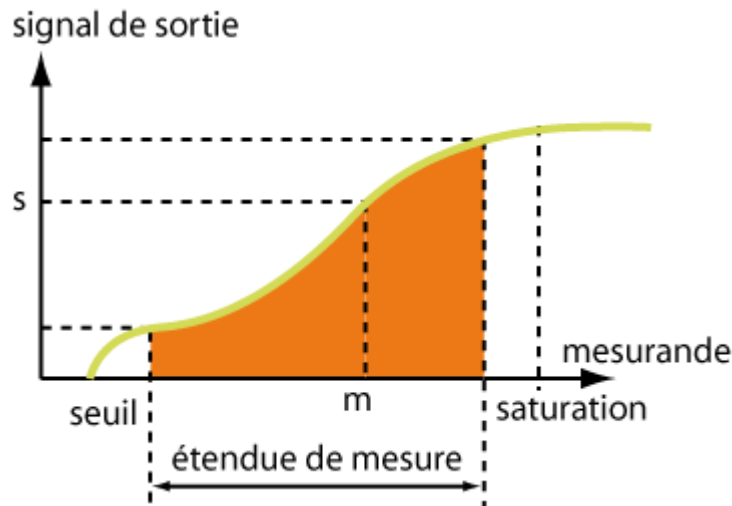


Figure 14 : courbe d'étalonnage d'un capteur

L'exploitation numérique de la grandeur de sortie du capteur en fonction de la mesurande s'obtient comme suit :

$$S = f(m) \quad (1.1)$$

La mesurande : c'est l'objet de la mesure ou plus simplement la grandeur à mesurer.

Le mesurage : c'est l'ensemble des opérations pour déterminer la valeur de la mesurande.

La mesure : c'est le résultat du mesurage. Autrement dit c'est la valeur de la mesurande.

Les capteurs typiquement installés sur un robot sont des capteurs ultrasonores, capteurs laser de proximité, des encodeurs de roues (odomètres), des caméras optiques et des microphones.

Les types d'informations perçues ainsi que leur précision varient beaucoup d'un capteur à l'autre. Par exemple, un capteur laser de proximité permet de mieux percevoir les contours de l'environnement que les sonars puisque le capteur offre une meilleure résolution angulaire et une meilleure précision sur la distance.

La classification des capteurs est réalisée suivant les phénomènes qu'ils mesurent : force, énergie, distance, champ magnétique, etc. Par conséquent, la classification peut être élaborée suivant que les capteurs utilisent un phénomène physique générateur de tension ou de courant, on parle alors de capteurs actifs, ou qu'ils nécessitent une source d'alimentation pour pouvoir traduire la variation du phénomène en tension, ce sont alors des capteurs passifs. Ils peuvent se classer aussi selon le type d'information qu'ils délivrent soit analogique ou numérique.

En robotique mobile, on classe traditionnellement les capteurs en deux catégories selon qu'ils mesurent l'état du robot lui-même ou l'état de l'environnement dans lequel il évolue. Dans le premier cas, on parle de proprioception et donc de capteurs proprioceptifs. On trouve dans cette catégorie des capteurs de position, de vitesse des roues, ainsi que des capteurs de charge de la batterie.

Dans un second cas, les capteurs renseignant sur l'état de l'environnement, donc tous ce qui est extérieur au robot, sont appelés capteurs extéroceptifs. Il s'agit de capteurs donnant la distance du robot par rapport à un obstacle, la température, le contact du robot avec l'environnement, etc. La figure 15 illustre ces deux types de capteurs :

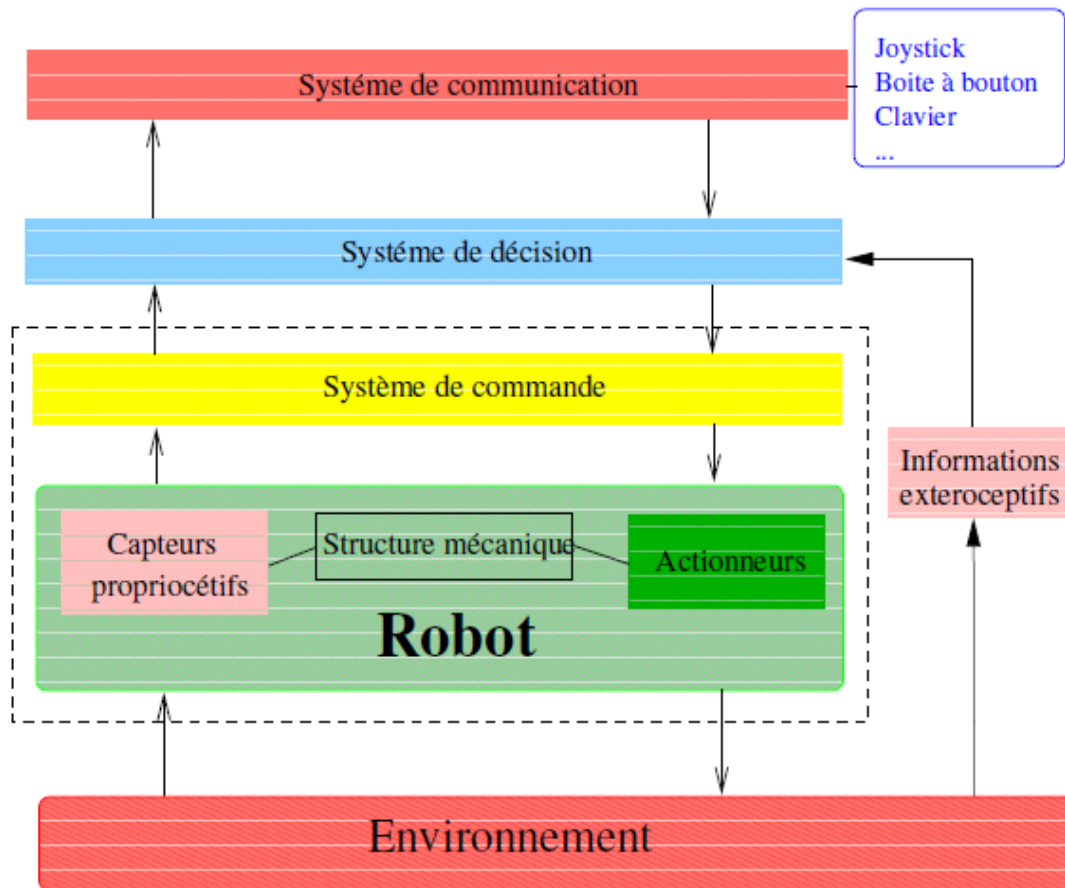


Figure 15 : Architecture d'un robot.

Il existe une classe de capteurs non inclus dans les boucles d'asservissement d'un robot, dont l'intérêt est de veiller sur l'état du robot lui-même. Ce sont des capteurs de sécurité qui se déclenchent en cas de dysfonctionnement ou de panne dans les boucles internes ou externes. Ce type de capteur génère un ordre d'arrêt du robot lorsque le dépassement d'un certain seuil a lieu et ce afin d'éviter que les composants du robot ne soient détruits.

II.3.1.1 Capteurs proprioceptifs :

Ce type de capteurs fournit, des informations élémentaires sur les paramètres cinématiques du robot, permettant ainsi un contrôle permanent et la bonne exécution du mouvement.

Les informations sensorielles gérées sont généralement des vitesses, des accélérations, des angles de rotation ou des angles d'attitude. On peut regrouper ces capteurs internes en deux familles :

- Les capteurs de déplacement qui comprennent les odomètres, les accéléromètres, les radars Doppler et les mesureurs optiques. Cette catégorie permet de mesurer des déplacements élémentaires, des variations de vitesse ou d'accélération sur des trajectoires rectilignes ou curvilignes.
- Les capteurs d'attitude qui mesurent deux types de données : les angles de cap et les angles de roulis et de tangage. Ils sont principalement constitués par des gyroscopes et des gyromètres, les capteurs inertiels composites, les inclinomètres et les magnétomètres. Ces capteurs sont en majorité de type inertiel, en d'autres termes ces capteurs sont utilisés afin de déterminer le mouvement absolu d'un mobile.

II.3.1.1.1 Les capteurs de déplacement :

II.3.1.1.1.1 Les odomètres :

Les odomètres permettent de fournir une quantification des déplacements curvilignes du robot en mesurant la rotation de ses roues. Le calcul de la position relative du robot est réalisé par intégration des rotations élémentaires des roues.

Les odomètres sont généralement composés de codeurs incrémentaux qui permettent de mesurer les angles de rotation avec précision qui dépend de la résolution du codeur. Des méthodes fondamentales sont utilisées pour la mesure de position. Dans la première méthode, le capteur est lié à la partie mobile du robot, et fournit un signal dont la variation traduit la position. Il existe des codeurs digitaux absolus où il est déterminé par lecture optique d'une graduation codée en binaire portée par un disque solidaire. Dans la seconde méthode le capteur délivre une impulsion à chaque déplacement élémentaire : la position est déterminée par comptage des impulsions émises, ou décomptages selon le sens du déplacement. Il s'agit de capteur incrémental. L'information de déplacement nécessitera la connaissance du diamètre des roues, de l'entraxe des roues, de la structure mécanique et cinématiques du véhicule.

II.3.1.1.1.2 Les accéléromètres :

Un accéléromètre est un capteur qui nous permet de mesurer l'accélération linéaire du mobile sur lequel il est fixé, il fournit, en sortie, une tension proportionnelle à l'accélération linéaire dans une direction fixée par sa géométrie. Le corps d'épreuve est le même dans tous les capteurs d'accélération et on le nomme une masse sismique. Lorsque le capteur est soumis à une accélération, l'inertie de la masse sismique résiste à son mouvement avec une force $F = ma$. Il faut donc mesurer l'amplitude de la force F pour déduire l'accélération. Cela sera fait par l'élément de transduction.

La mesure de l'accélération utilise trois technologies de capteurs différentes :

- Accéléromètres piézoélectriques.
- Accéléromètres piézorésistifs.
- Accéléromètres asservis.

II.3.1.1.1.3 Les capteurs d'attitude :

Les capteurs d'attitude permettent d'estimer les paramètres intrinsèques du robot que sont les angles de cap, de roulis et de tangage. Ces capteurs sont principalement de type inertiel.

II.3.1.1.1.4 Les gyroscopes :

Les gyroscopes permettent de mesurer une variation angulaire. Ils sont intéressants en robotique mobile parce qu'ils peuvent compenser les défauts des odomètres. Une erreur d'orientation odométrique peut entraîner une erreur de position cumulative qui peut être diminuée voire compensée par l'utilisation conjointe de gyroscopes. Les gyroscopes à fibre optique, connus pour leur grande précision, sont devenus une solution attractive pour la navigation en robotique mobile.

II.3.1.1.1.5 Les gyromètres :

Le gyromètre est un capteur de mouvement. Il fournit une information de vitesse angulaire par rapport à un référentiel inertiel fixe. On distingue le gyromètre et le gyroscope qui est un capteur de position angulaire. La distinction est parfois subtile car un même appareil peut fonctionner en gyroscope ou gyromètre. Deux technologies sont actuellement disponibles, permettant un rapport coût/performance compatible avec des applications robotiques :

- Les gyromètres piézoélectriques
- Les gyromètres à fibres optiques

Il est utilisé principalement dans les avions, les bateaux, les sous-marins pour déterminer sa vitesse et sa position.

II.3.1.1.1.6 Les gyrocompas :

Le gyrocompas est un capteur qui permet de mesurer le cap. Il est composé d'un gyroscope et d'un compas magnétique. Le gyrocompas conserve le nord magnétique durant tout le déplacement du véhicule, après l'avoir initialement déterminé de façon autonome.

II.3.1.1.1.7 Le magnétomètre :

Le magnétomètre, qui est aussi appelé compas magnétique, mesure la direction du champ magnétique terrestre afin de déduire l'orientation du robot. Parmi toutes les technologies adoptées pour ce type de capteur, la mieux adaptée pour la robotique mobile est celle dite à vanne de flux. Elle a l'avantage de consommer peu d'énergie, de n'avoir aucune pièce mobile, d'être résistante aux chocs et vibrations.

II.3.1.2 Capteurs extéroceptifs :

Les capteurs extéroceptifs permettent de percevoir le milieu d'évolution du robot. Ils sont généralement le complément indispensable aux capteurs présentés précédemment. Deux familles de capteurs extéroceptifs embarqués peuvent être identifiées, les capteurs télémétriques et les systèmes de vision. Ces capteurs servent pour la vérification et l'amélioration de la trajectoire suivie par le robot, la mesure des interactions entre le robot et l'environnement et la perception de l'environnement et voire même pour la prévention et la sécurité du robot [3].

Parmi les capteurs externes qui seront étudiés dans la partie suivante, on retiendra pour notre part le capteur ultrasonore. Ce dernier fait partie des capteurs de distance à portée moyenne. On le retrouve pratiquement sur tous les robots mobiles. Souvent plusieurs capteurs sont montés autour du robot formant une sorte de ceinture de sécurité dont le rôle principal est la détection des obstacles.

II.3.1.2.1 Les capteurs télémétriques :

Le fonctionnement des capteurs utilisés classiquement en robotique mobile sont ceux que nous allons présenter maintenant :

- Les capteurs lasers
- Les capteurs infrarouges
- Les capteurs à ultrasons

Chacun de ces capteurs est caractérisé par son principe de fonctionnement, ainsi que par sa capacité de perception.

II.3.1.2.2 Les capteurs lasers :

Le capteur laser est basé sur l'émission d'ondes électromagnétiques, un faisceau d'ondes très concentré est émis du capteur vers l'objet dont on souhaite mesurer l'éloignement, puis en mesurant le déplacement de l'onde réfléchie de l'objet en déduit la distance. Une étude très détaillée de ce type de capteur est préposée dans [2]. Les systèmes laser possèdent de nombreux avantages qui en font des capteurs souvent utilisés dans les applications de robotique mobile [3], [4] :

- La résolution angulaire : des faisceaux angulairement très fins peuvent être obtenus avec des lentilles d'émission de petite taille, du fait de la courte longueur d'onde émise.
- La réponse optique des cibles : les longueurs d'ondes courtes permettent d'obtenir un écho pour des angles d'incidence du faisceau sur la cible allant jusqu'à 75°, et améliorent de ce fait la probabilité de détection.
- La précision sur la mesure de distance par rapport à la portée : pour les télémètres, une précision de l'ordre du dixième du millimètre peut être obtenue sur des distances de l'ordre de 30 mètres.

L'inconvénient majeur du laser se situe au niveau de l'électronique de mesure qui doit être capable de mesurer des intervalles de l'ordre de la picoseconde pour des précisions de

distance de l'ordre du millimètre. Ceci implique un matériel coûteux et complexe. En robotique les systèmes laser sont utilisés principalement de trois manières différentes :

- En télémètre : ils sont utilisés, tout comme les capteurs à ultrasons, pour des mesures de distances [5].
- En goniomètre : ils sont utilisés pour des mesures d'angles [6].
- En source de lumière structurée : ils sont généralement utilisés pour la modélisation 3D de l'environnement. La source laser sera dans ce cas toujours associée à une caméra et fournira des informations sur la profondeur [7].

II.3.1.2.3 Les capteurs infrarouges :

Les capteurs infrarouges sont constitués d'un ensemble d'émetteur/récepteur fonctionnant avec des radiations non visibles, dont la longueur d'onde est juste inférieure à celle du rouge visible. La mesure des radiations infrarouges étant limitée et, en tout état de cause, la qualité est très dégradée au-delà d'un mètre, ces dispositifs ne servent que rarement de télémètres. On les rencontrera le plus souvent comme détecteurs de proximité. Il faut noter que ce type de détection est sensible aux conditions extérieures, notamment à la lumière ambiante, à la sécularité des surfaces sur lesquelles se réfléchissent les infrarouges, à la température et même à la pression ambiante. Ces capteurs ne sont pas complètement directionnels et leur caractéristique présente une zone de détection conique à l'origine d'incertitudes. Enfin, l'alternance de phase d'émission et de réception impose une distance de détection minimale.

Ce genre de capteurs sont des télémètres de faible portée, ils mesurent des distances absolues ou relatives par détection infrarouge et disposent, selon les modèles, de sorties numériques ou analogiques. La mesure est établie à partir de l'inclinaison du rayon réfléchi ; le rayon est en effet dévié à l'aide d'un système optique sur une matrice de photodiodes (PSD), comme présenté sur la figure 16 :

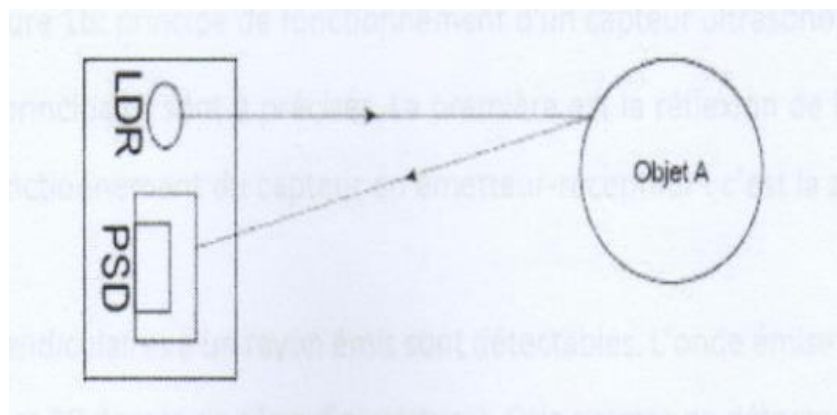


Figure 16 : Télémètres infrarouges

II.3.1.2.4 Les capteurs ultrasonores :

Les capteurs ultrasonores utilisent des vibrations sonores dont les fréquences ne sont pas audibles par l'oreille humaine. Les sons audibles par l'homme ont des fréquences comprises approximativement entre 20Hz et 20kHz. Les ultrasons, correspondent aux sons qui ont une fréquence supérieure à 20kHz (les sons ayant une fréquence inférieure à 20 kHz sont appelés infrasons.), les fréquences couramment utilisées dans ce type de technologie vont de 20 kHz à 200kHz. Les ultrasons émis se propagent dans l'air et sont réfléchis partiellement lorsqu'ils heurtent un corps solide (voir la figure 17), en fonction de son impédance acoustique. La distance entre la source et la cible peut être déterminée en mesurant le temps que prennent les ondes ultrasons pour faire un aller-retour entre le capteur et le corps distant.

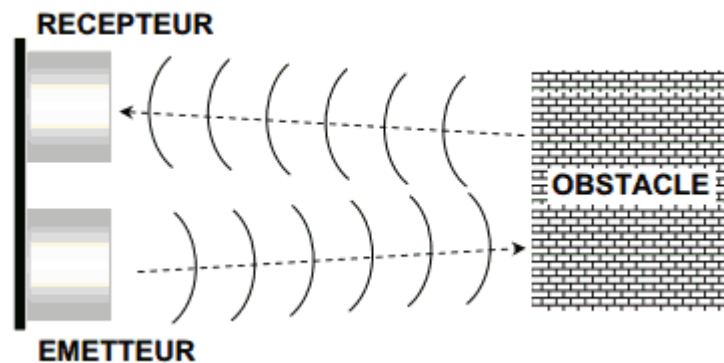


Figure 17 : principe de fonctionnement d'un capteur ultrasonore.

Deux caractéristiques principales sont à préciser. La première est la réflexion de l'onde sur un objet. La seconde découle du fonctionnement du capteur en émetteur-récepteur : c'est la zone aveugle ou morte.

Seuls les objets perpendiculaires à un rayon émis sont détectables. L'onde émise n'est pas très directive (typiquement entre 20 et 40 degrés de cône d'ouverture). Cela permet de détecter plus d'obstacles, mais on ne dispose alors plus d'information précise sur l'angle de mesure (voir les figures 18, 19, 20). On sait seulement que l'onde mesurée est réfléchi perpendiculairement à l'obstacle rencontré.

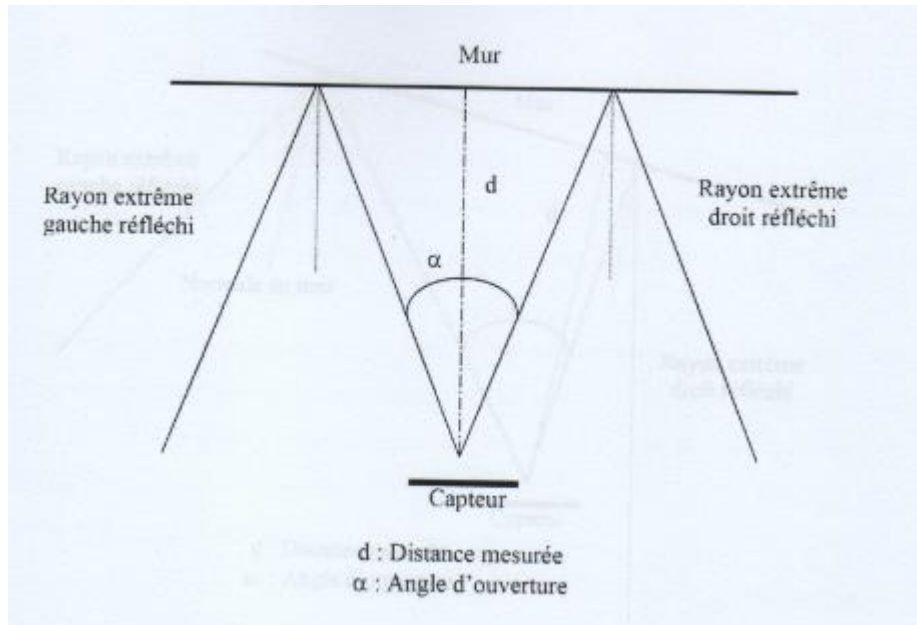


Figure 18 : obstacle perpendiculaire à l'axe du capteur
 (L'obstacle dans ce cas sera détecté par le capteur).

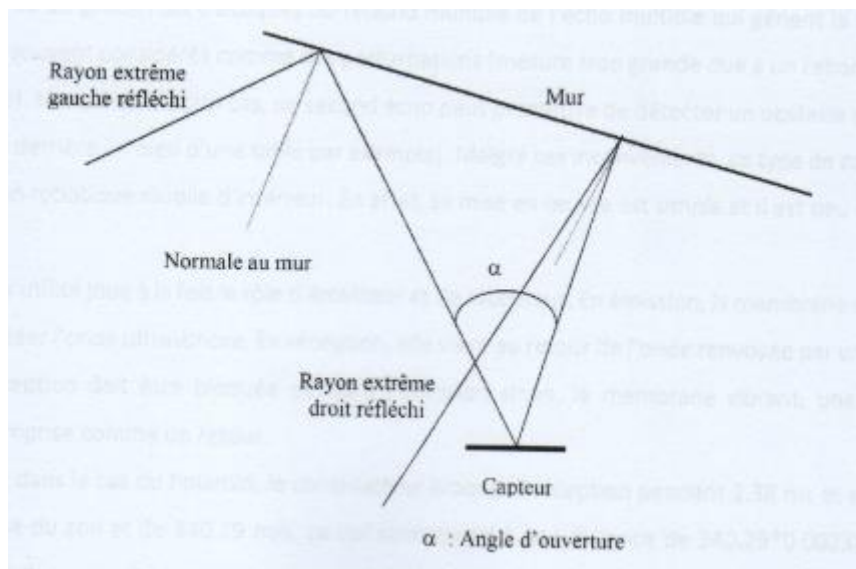


Figure 19 : obstacle en biais il ne sera pas détecté par le capteur.

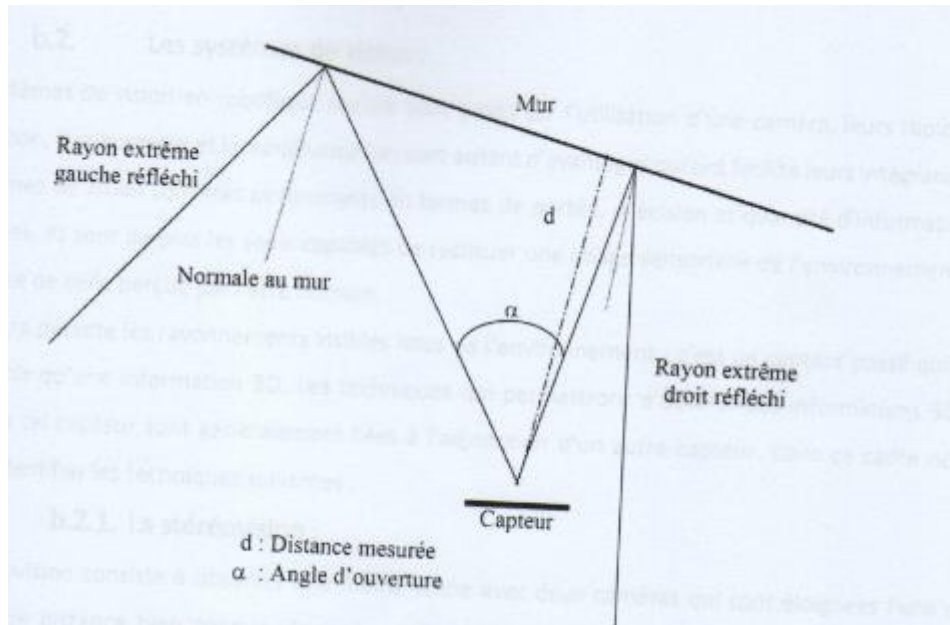


Figure 20 : obstacle en biais détecté par le capteur.

En plus de ces problèmes, on rencontre des problèmes classiques dû aux rebonds multiples de l'écho qui gênent la mesure. Ils sont le plus souvent considérés comme des perturbations (mesure trop grande due à un rebond multiple par exemple). Mais, dans certain cas, un second écho peut permettre de détecter un obstacle derrière un autre. Malgré ces inconvénients, ce type de capteurs est très utilisé en robotique mobile d'intérieur. En effet, sa mise en œuvre est simple et il est peu coûteux.

Ce capteur joue à la fois le rôle d'émetteur et de récepteur. En émission, la membrane du capteur vibre pour créer l'onde ultrasonore. En réception, elle vibre au retour de l'onde renvoyée par l'obstacle.

Ainsi, la réception doit être bloquée pendant l'émission sinon, la membrane vibrant, une onde émise sera détectée comme une onde de retour.

Par exemple dans le cas du Polaroid, le constructeur bloque la réception pendant 2.38 ms et en sachant que la vitesse du son est de 340.29 m/s, ce qui correspond à une distance de 0.81 m dans ce cas (l'onde fait un aller-retour). La zone aveugle est donc de $0.81/2 = 0.405$. Dans les faits, elle est légèrement plus importante vu qu'elle vaut environ 0.44m.

II.3.2 Les systèmes de vision :

Les systèmes de vision en robotique mobile sont basés sur l'utilisation de caméras, leurs rapidités d'acquisition, la robustesse et la miniaturisation sont autant d'avantages qui ont facilité leurs intégrations, les systèmes de vision sont très performants en termes de portée, de précision et de quantité d'informations exploitables. Ils sont de plus les seuls capables de restituer une image sensorielle de l'environnement la plus proche de celle perçue par l'être humain.

Une caméra détecte les rayonnements visibles issus de l'environnement : c'est un capteur passif qui ne peut fournir qu'une information 2D. Les techniques qui permettront d'obtenir des informations 3D à partir de tel capteur sont généralement liées à l'adjonction d'un autre capteur. Dans ce cadre nous pouvons identifier les techniques suivantes :

II.3.2.1 La stéréovision :

La stéréovision consiste à observer une même scène avec deux caméras qui sont éloignées l'une de l'autre d'une distance bien connue. Connaissant la géométrie exacte du système stéréoscopique, la première étape de reconstruction 3D consiste à mettre en correspondance les deux images. C'est le même principe de reconstruction de la profondeur que la vision humaine. La figure 21 illustre précisément le principe de la stéréovision. Tout point M visible depuis les deux caméras se projette en P_1 sur le plan image de la première caméra et en P_2 sur le plan de la deuxième caméra. Inversement, connaissant deux points en correspondance P_1 et P_2 , ainsi que la distance D séparant les centres optiques O_1 et O_2 des deux caméras, les coordonnées du point P sont données par l'intersection de (O_1P_1) et (O_2P_2) . Cette même technologie est utilisée dans les téléviseurs 3D [12].

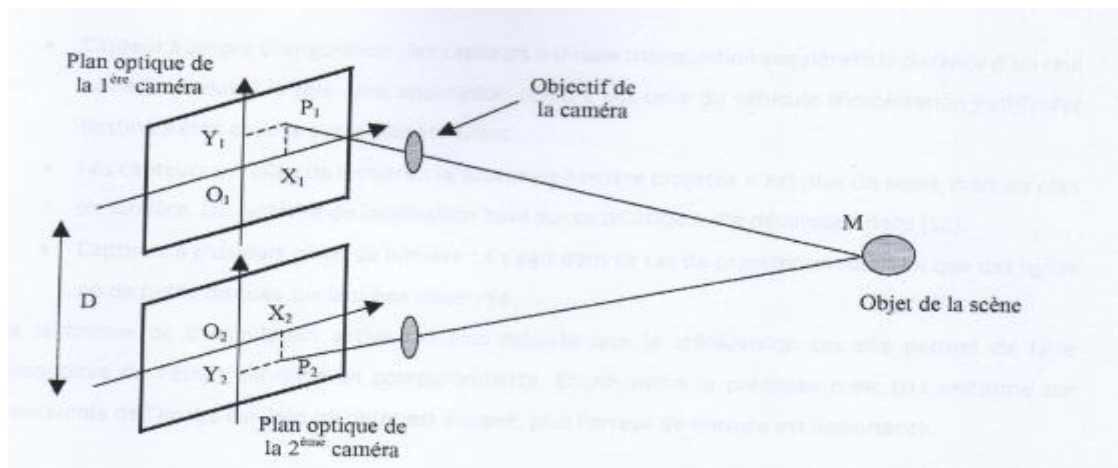


Figure 21 : principe de la stéréovision

La stéréovision apparaît aujourd'hui comme un des moyens de perception les plus performants en robotique mobile que ce soit pour la modélisation des scènes d'intérieurs [8] ou pour établir une cartographie de l'environnement [9].

II.3.2.2 Les capteurs à triangulation active :

Les capteurs à triangulation active ou lumière structurée sont utilisés pour l'obtention d'information 3D comme pour la stéréovision, mais on remplace une caméra par un laser qui sera chargé de projeter un motif de lumière structurée. Le principe est d'éclairer l'environnement avec une lumière cohérente et de l'observer avec une caméra sous une certaine parallaxe. Connaissant la position de la source lumineuse et celle de la caméra, l'information peut être obtenue par triangulation [10]. Ces capteurs sont utilisés pour obtenir

des images de profondeur, mais également pour l'évitement d'obstacles [11], pour la numérisation d'objet 3D.

La technique de triangulation active est plus simple que la stéréovision car elle permet de faire l'économie de l'étape de mise en correspondance. En revanche la précision n'est pas uniforme sur l'ensemble de l'image car plus un objet est éloigné, plus l'erreur de mesure est importante.

II.3.2.3 La vision omnidirectionnelle :

Ces systèmes de perception associent une caméra et un élément permettant d'obtenir une vue sur 360° de l'environnement. Suivant les éléments ajoutés à la caméra, on pourra distinguer quatre techniques pour obtenir une image omnidirectionnelle.

- Génération d'images par utilisation de plusieurs caméras. Cette technique consiste donc à utiliser plusieurs caméras afin de couvrir l'intégralité du champ de vision possible [13].
- Génération d'images multiples par rotation d'une caméra. Cette technique consiste à faire pivoter une caméra autour d'un axe. Cette méthode permet d'avoir une prise de l'environnement avec une seule caméra, mais il faut que celle-ci soit capable de faire des rotations à 360° afin de couvrir tous l'environnement [14].
- Utilisation de lentille spéciale. Cette méthode consiste en l'ajout d'un ensemble optique sur la caméra CCD (Charge Coupled Device) afin de dévier les rayons lumineux, comme par exemple l'œil de poisson qui utilise le même principe.
- La dernière méthode est l'utilisation de miroirs convexes.

II.4 La locomotion :

II.4.1 Les actionneurs :

Pour bouger à l'intérieur de son environnement et interagir avec celui-ci, les robots sont équipés de ce qu'on appelle des « Actionneurs ». Un actionneur est un dispositif qui transforme l'énergie délivrée par l'interface de puissance, en énergie utilisable par les effecteurs de processus.

Les actionneurs sont aux robots ce que les muscles sont à l'être humain. Il existe plusieurs types d'actionneurs qui sont utilisés en robotique mobile, ils sont classifiés selon trois catégories :

- Actionneurs hydrauliques : ils exploitent la propriété d'incompressibilité des fluides.
- Actionneurs pneumatiques : ils exploitent les propriétés de compression et de dilatation de l'air.
- Actionneurs électromécaniques: ils exploitent les propriétés telle que l'effet magnétique, l'effet piézoélectrique,...etc.

Dans le domaine de la robotique mobile, les actionneurs les plus utilisés sont les actionneurs électriques, ils assurent une conversion d'énergie et d'information de la forme électrique à la forme mécanique. Un actionneur électromécanique est aussi appelé moteur.

Les moteurs les plus souvent employés en robotique mobile sont les moteurs à courant continu, les moteurs pas à pas, les servomoteurs, et les motoréducteurs.

II.4.1.1 Les moteurs à courant continu:

Une machine à courant continu est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique, selon la source d'énergie.

- En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique, dans ce cas elle est aussi appelée dynamo.

Cependant, la machine à courant continu étant réversible et susceptible de se comporter soit en « moteur » soit en « générateur » dans les quatre quadrants du plan couple-vitesse, la distinction moteur/générateur se fait « communément » par rapport à l'usage final de la machine.

Inventée par Zénobe Gramme et présentée à l'Académie des Sciences, à Paris, en 1871, c'était au départ un simple générateur de courant continu (pour applications galvanoplastiques).

Un moteur à courant continu se compose généralement de trois éléments essentiels :

- Une partie fixe : l'inducteur (stator) constitué d'un aimant permanent.
- Une partie mobile : l'induit (rotor) comportant une ou plusieurs bobines tournant dans le champ magnétique créé par l'inducteur.
- Le collecteur, associé aux balais, permet de relier les enroulements du rotor à l'extérieur du moteur.

II.4.1.2 Les moteurs pas à pas :

Les moteurs pas à pas sont des moteurs spéciaux utilisés pour commander avec une grande précision le déplacement et la position d'un objet. Comme leur nom l'indique, ces moteurs tournent par incréments discrets. Chaque incrément de rotation est provoqué par une impulsion de courant fournie à l'un des enroulements du stator.

Selon sa construction, un moteur pas à pas peut avancer de 90°, 45°, 18°, ou d'une fraction de degré seulement par impulsion. En faisant varier la fréquence des impulsions, on peut faire tourner le moteur très lentement, d'un pas à la fois, ou rapidement à des vitesses aussi élevées que 4000 tr/min.

On trouve trois types de moteurs pas à pas :

- Le moteur à aimants permanents.
- Le moteur hybride.
- Le moteur à réluctance variable.

II.4.1.2.1 Le moteur à aimants permanents :

C'est le modèle dont le fonctionnement est le plus simple. Le rotor est constitué d'un aimant permanent, et le stator comporte deux paires de bobines. En agissant sur les bobines alimentées et le sens des courants, on fait varier le champ créé par le stator. A chaque pas, la direction du champ induit par le stator tourne de 90° . L'aimant permanent suit le déplacement du champ magnétique créé par les bobines et s'oriente selon une de ses quatre positions stables. Comme le rotor est aimanté, lorsque le moteur n'est pas alimenté, le flux magnétique dû à l'aimant permanent crée un couple résiduel en se plaçant dans l'axe de l'une des bobines.

Pour augmenter le nombre de positions stables et donc de pas du moteur à aimant permanent, on peut alimenter successivement une puis deux paires de bobines : c'est le mode "demi-pas"

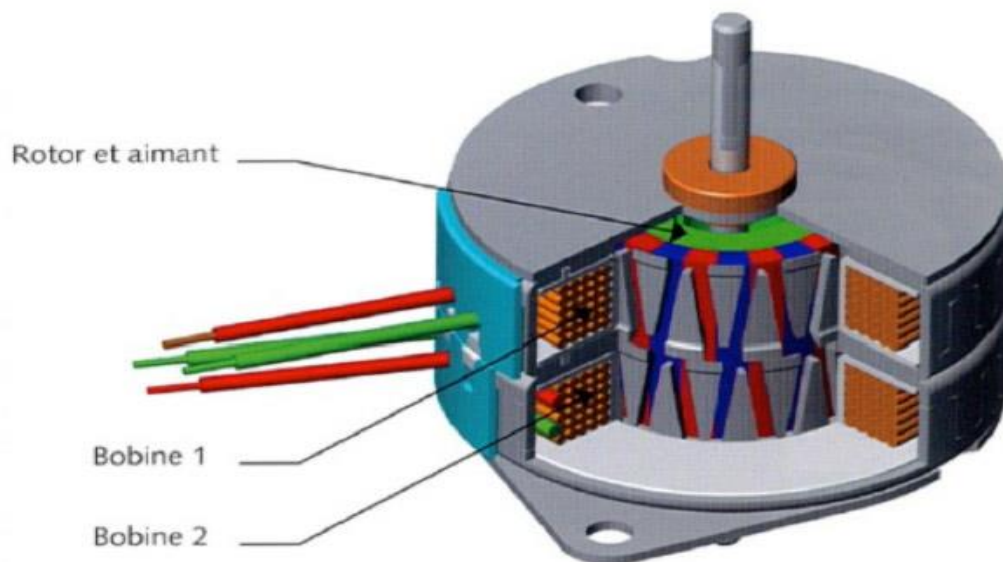


Figure 22: Moteur à aimants permanent.

II.4.1.2.2 Le moteur à réluctance variable :

Ce type de moteur pas à pas est composé d'un barreau de fer doux et d'un certain nombre de bobines. Lorsqu'on alimente une bobine, elle devient un électroaimant et le barreau de fer cherche naturellement à s'orienter suivant le champ magnétique. Dans cet exemple on alimente la phase 1, puis la phase 2, puis la phase 3... Si nous souhaitons changer le sens du moteur, il suffit de changer l'ordre d'alimentation des bobines

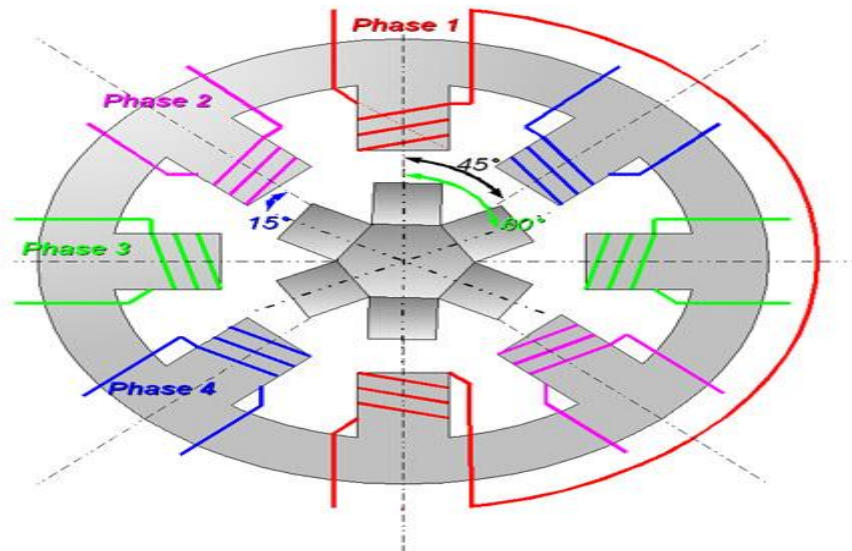


Figure 23: Moteur à réluctance variable

II.4.1.2.3 Le moteur hybride :

Le moteur pas à pas « hybride » allie le principe du moteur à réluctance variable à celui du moteur à aimant permanent. Le rotor présente plusieurs dents comme pour un moteur pas à pas à réluctance variable, mais chaque dent est polarisée comme pour un moteur pas à pas à aimants permanents. Physiquement le rotor est composé de deux éléments identiques à un rotor de moteur à réluctance variable, reliés ensemble par un aimant permanent, avec un déphasage d'une demi-dent. De ce fait ces deux éléments ont une polarisation différente (nord et sud) et vont réagir à la polarisation de chacune des dents du stator. C'est cette polarisation qui permet de n'utiliser que 2 bobines en même temps.

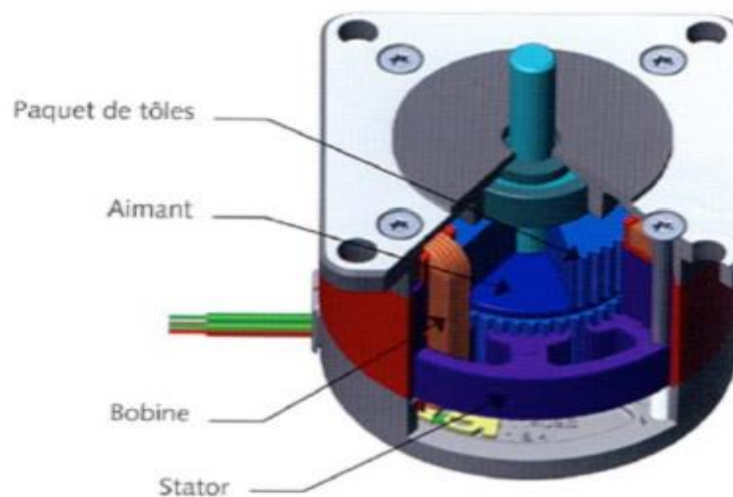


Figure 24: Moteur hybride.

II.4.1.2.4 Comparaison des différents types de moteurs pas à pas :

	Aimant permanent	Hybride	Reluctance variable
Cout	Bas	Haut	Moyen/haut
Vitesse	Bas	Très haut	Haut
Couple résiduel	Haut	Moyen	Minimum
Amortissement	Bon	Moyen/bon	Mauvais
Inertie rotor	Haut	Bas	Bas
Rendement	Moyen	Très haut	Moyen
Angle de pas	7.5°/15°/18°	0.9°/1.8°	1.8°
Nbrs de pas/tour	48/24/20	400/200	200
Précision du pas	Bas	haut	Moyen

Tableau 2 : Comparatifs des moteurs pas à pas.

II.4.1.3 Les servomoteurs :

Un servomoteur (souvent abrégé en « servo », provenant du latin *servus* qui signifie « esclave », est un moteur avec quelques spécificités en plus, contrairement à un moteur classique qui est utilisé pour tourner avec une vitesse proportionnelle à une tension, pour obtenir une position. Un servomoteur est capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure, cela grâce à un système de correction qu'il embarque, un servomoteur est donc un système asservi grâce à un correcteur généralement de type PID.

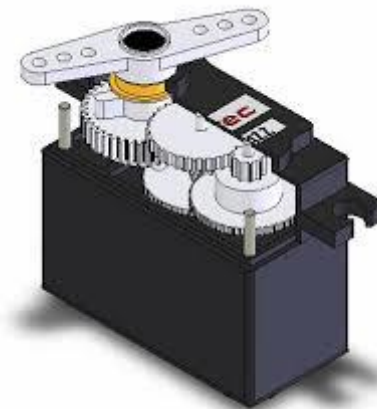


Figure 25: Un servomoteur

II.5 Les Architectures décisionnelles d'un robot mobile :

Un robot mobile autonome est un système doté de capacités décisionnelles lui permettant d'accomplir un certain nombre de tâches, dans un environnement plus ou moins connu d'avance, et cela avec une intervention humaine réduite voire inexistante.

Un robot est composé d'un ensemble de modules, chacun étant responsable d'une ou plusieurs capacités. L'un des premiers défis à résoudre est de déterminer comment relier efficacement les différents modules. Pour ce faire, il faut élaborer une architecture décisionnelle qui dicte les responsabilités de chacun des modules et comment l'information circule entre ces derniers. Depuis les débuts de la robotique, beaucoup d'architectures ont été proposées. Elles sont ainsi classées en trois grandes catégories [15] :

- Délibérative.
- Comportementale.
- Hybride.

II.5.1 Architecture délibérative :

Les architectures délibératives sont les premières à avoir été proposées. Les architectures de ce type sont basées sur des processus complètement planifiés. Si par exemple on veut exécuter un déplacement, un robot basé sur ce type d'architecture calcule un plan complet, lui disant d'avancer de x mètres, ensuite de tourner avec un angle ϕ , et ainsi de suite. Lorsqu'il y a un changement dans l'environnement, l'exécution est suspendue et un nouveau plan est généré.

Ce type d'architecture souffre de plusieurs lacunes importantes. Premièrement, puisque les capteurs sont imprécis et que l'environnement est dynamique et partiellement observable, il est très difficile de tout prévoir à l'avance. Pour ces raisons, il n'est pas d'une très grande utilité de tout planifier à l'avance, puisque les plans seront constamment à refaire. Un autre problème avec cette architecture est que la génération de plans précis demande beaucoup de ressources (temps de calcul et mémoire) [17].

II.5.2 Architecture comportementale :

L'architecture comportementale, proposée par Brooks [17] est inspirée du comportement des insectes. L'idée générale est de développer plusieurs petits modules simples et indépendants les uns des autres, une fois regroupés ces modules auront un comportement plus intelligent qui émergera sans que celui-ci ne soit préalablement programmé. Ce type d'architecture est complètement à l'opposé des architectures délibératives et ne fait aucune place à des processus raisonnés.

Les comportements sont tous indépendants les uns des autres. Par exemple, on peut avoir des comportements pour l'évitement d'obstacles, suivi de chemin ou manipulation d'objets. Les comportements sont exécutés parallèlement à une certaine fréquence. Lors d'une itération, chaque comportement calcule une ou plusieurs commandes motrices qui sont envoyées à un

module d'arbitration. Ce dernier fusionne l'ensemble des commandes reçues et calcule les commandes finales devant être envoyées à chacun des actionneurs du robot.

Ce type d'architecture a la difficulté à réaliser des tâches structurées, puisqu'elle ne contient aucun processus délibératif. En effet, les tâches complexes requièrent la capacité du robot à prédire les conséquences futures de ses actions afin de sélectionner celles qui conviennent le mieux pour la réalisation de ses activités. En d'autres termes, ces tâches complexes ne peuvent se faire qu'avec planification [16].

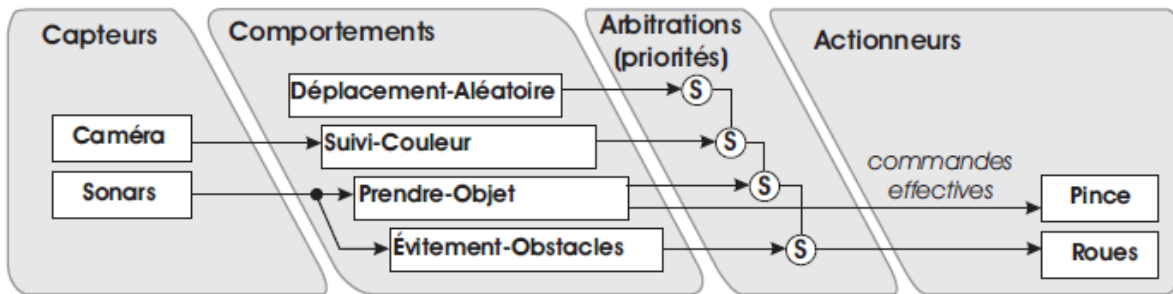


Figure 26 : Architecture comportementale.

II.5.3 Architecture hybride :

Les limitations des deux types d'architectures précédentes justifient l'émergence récente des architectures hybrides [16], tentant de combiner les avantages des architectures délibératives et comportementales. Elles sont généralement décomposées en plusieurs niveaux. Dans la partie supérieure, on place les modules de type délibératif. Dans la partie inférieure, on retrouve les modules de type comportemental. La figure 27 illustre un exemple d'architecture hybride où un séquenceur s'occupe des capacités délibératives.

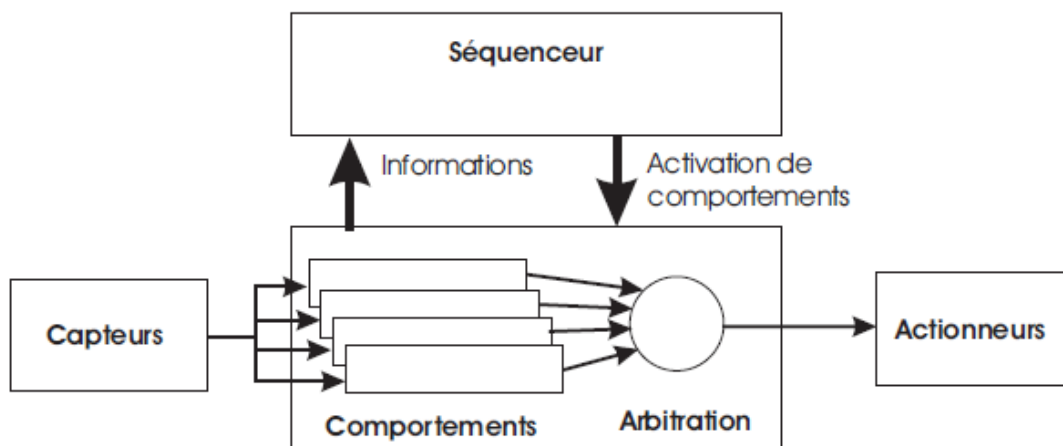


Figure 27 : Architecture hybride.

II.6 La navigation :

On désigne par navigation dans le domaine de la robotique mobile, la capacité d'aller d'une position initiale à une position finale et ce de façon sécuritaire et efficace [17].

Ce déplacer d'un point A de coordonnées (x_1, y_1) à un point B de coordonnées (x_2, y_2) , nécessite la mise en œuvre de fonctionnalités potentiellement complexes de perception, décision, et action (voir la figure 28). Le rôle de ces fonctionnalités se fait comme suit [13] :

- **Perception** : il s'agit essentiellement de détecter des obstacles et d'effectuer éventuellement une modélisation de l'environnement pour fournir et mettre en forme les informations nécessaires à la décision sur le déplacement à réaliser.
- **Décision** : l'environnement étant perçu et éventuellement modélisé, il faut décider du type de mouvement à exécuter en générant des consignes de positions appropriées à envoyer au robot ou en choisissant une trajectoire à exécuter.
- **Action** : il s'agit alors de veiller à réaliser le mouvement ou la trajectoire décidé, suivre les consignes de vitesses reçues en appliquant le type de commande adapté.

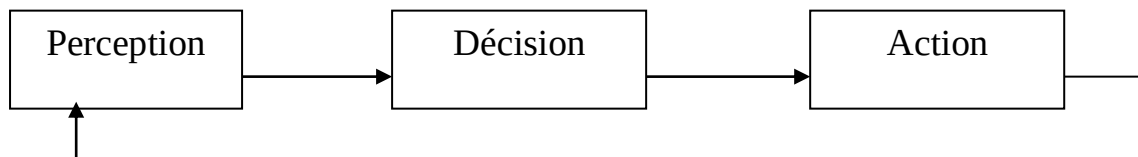


Figure 28 : le cycle perception/décision/action que doit réaliser un robot.

II.7 Conclusion :

Dans ce chapitre on s'est intéressé à résumer toutes les notions de base nécessaires à la compréhension de la robotique puis nous nous sommes intéressé plus particulièrement au robot mobile. En présentant un panorama de techniques appliquées dans la planification et la navigation d'un robot mobile autonome. Dans ce chapitre aussi nous avons abordé les différentes architectures d'un robot mobile, ainsi que les différents composants matériels le constituant.

On a étudié les différentes classes des robots mobiles qui sont catégorisés selon différents critères, que ce soit leur degré d'autonomie, ou le domaine d'application, ou bien encore selon le type de locomotion utilisé.

Par la suite nous avons également démontré comment un robot arrive à percevoir son environnement grâce aux informations recueillies par différents capteurs, tout en développant les notions qu'utilisent ces derniers, et comment ces informations sont utilisées par le robot afin de se mouvoir dans son environnement, et ce en suivant une architecture décisionnelle propre à chaque robot, cette dernière lui permet d'évoluer dans son environnement en utilisant des actionneurs de différents types.

Dans la partie qui va suivre nous allons aborder la conception d'un robot mobile explorateur, cela en mettant en évidence tous les éléments utiliser pour concevoir ce genre de robot.

CHAPITRE II : CONCEPTION D'UN ROBOT MOBILE

I. Introduction :

Après avoir présenté un état de l'art sur la robotique, et nous être intéressés particulièrement à la robotique mobile, nous allons maintenant passer à la conception de notre robot mobile. Pour cela on a besoin tout d'abord de choisir la structure du robot, ainsi que les différents éléments qui le composeront.

Tout d'abord, pour pouvoir se localiser dans un environnement, un robot mobile a besoin d'un moyen de perception (capteurs) qui l'informe sur sa position dans l'espace dans lequel il évolue.

Dans un second temps, un robot a besoin d'actionneurs (moteur, servomoteur) pour pouvoir se déplacer et interagir avec son environnement.

Enfin pour terminer, le robot a besoin d'un élément essentiel qui représente son cerveau. Ce composant est sans équivoque le microcontrôleur ATmega 328 de l'Arduino Uno, qui sert à orchestrer les différents éléments constituant le robot. Partant de l'acquisition des données qui parviennent des capteurs à la prise de décision, ensuite l'exécution des différentes tâches de déplacements qui permettent au robot de naviguer dans son environnement de manière plus ou moins autonome.

I.1 Structure de notre robot :

Dans notre travail nous avons choisi de concevoir un robot mobile autonome à roues, de type voiture, ce type de structure a pour avantage d'être plus stable mécaniquement et d'une complexité mécanique modérée ainsi qu'un déplacement aisé dans son environnement.

Notre robot utilise des motoréducteurs comme moyen de locomotion, ceux-ci sont en fait des moteurs à courant continu rattachés à un réducteur afin d'avoir un meilleur compromis entre la vitesse de rotation et le couple moteur. Ceci est utilisé dans le but de mieux contrôler le mouvement du robot et d'avoir une meilleure précision de ce dernier.

La commande de ces moteurs exige l'utilisation d'un pont en H qui est un jeu de quatre transistors montés selon une certaine configuration lui permettant de gérer le sens de circulation du courant traversant les moteurs et ainsi contrôler leur sens de rotation.

Nous utiliserons aussi un capteur HC-SR04 monté sur un servomoteur SG90, cet ensemble formera le système de perception de notre robot. Afin de pouvoir faire interagir tous ces éléments entre eux, nous faisons appel à la carte Arduino Uno qui est une carte de commande à base d'un microcontrôleur ATmega.

I.2 La conception matérielle de notre robot :

I.2.1 Le capteur à ultrason HC-SR04 :

Le capteur à ultrason HC-SR04 est un moyen précis pour la mesure de la distance (2 cm à 4 m avec une précision de 3 mm). Il est très utilisé en robotique pour son efficacité et sa facilité d'interaction avec toute une sorte de microcontrôleurs (Arduino, Microchip PIC, basic stamp, Propeller chip, etc.), il est aussi plus efficace qu'un capteur infrarouge qui lui peut être influencé par la lumière.

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.



Figure 29 : Capteur à ultrason HC-SR04.

I.2.1.1 Caractéristiques et spécifications :

Le HC-SR04 présente les Caractéristiques et spécifications suivantes :

- Dimensions : 45 mm x 20 mm x 15 mm.
- Portée : de 2 cm à 4 m.
- Résolution de la mesure : 0.3 cm.
- Angle de mesure efficace : 15 °.
- Largeur d'impulsion sur l'entrée de déclenchement : 10 μ s.

I.2.1.2 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.

I.2.1.3 Spécifications et limites :

Paramètre	Min	Type	Max	Unité
Tension d'alimentation	4.5	5.0	5.5	V
Courant de repos	1.5	2.0	2.5	mA
Courant de fonctionnement	10	15	20	mA
Fréquence des ultrasons	-	40	-	kHz

Tableau 3: Spécifications et limites du HC-SR04.

I.2.1.4 Le branchement :

Le branchement du capteur à la carte Arduino est fort simple comme le montre la figure suivante:

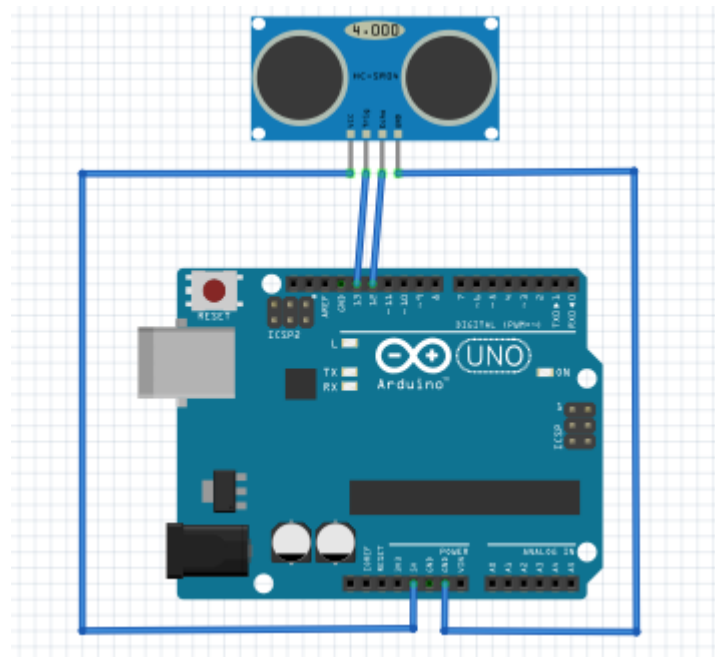


Figure 30 : Branchement du capteur à ultrason.

I.2.1.5 Principe de fonctionnement :

Pour déclencher une mesure, il faut présenter une impulsion "High" (5V) d'au moins 10 μ s sur l'entrée "Trig". Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz, puis le "Trig" se désactive et la broche "écho" s'active simultanément. Ensuite la broche de réception se désactive lorsque le signal d'émission est réfléchi. On peut donc conclure que la distance entre le capteur et la cible est proportionnelle à la durée d'activation de "l'écho" (voir figure 31 et 32) [27].

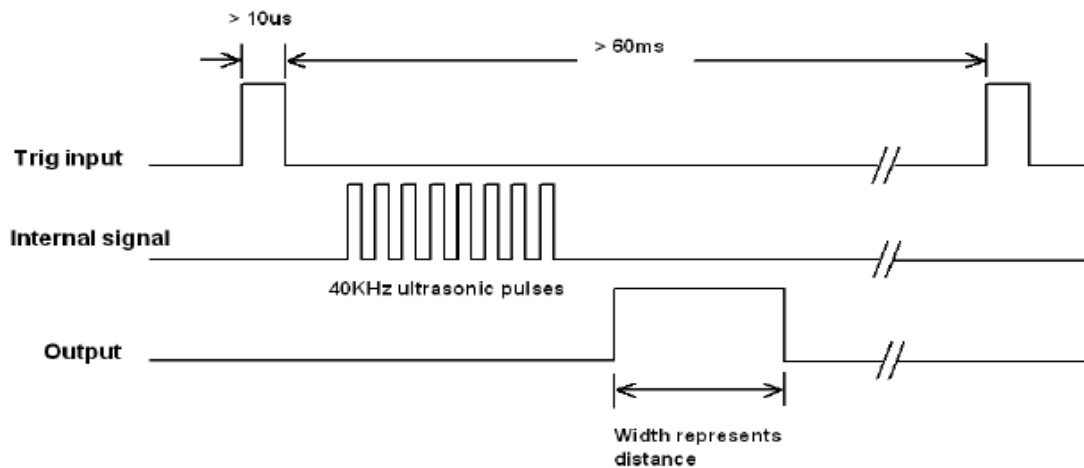


Figure 31 : Train d'impulsions émis par le capteur.

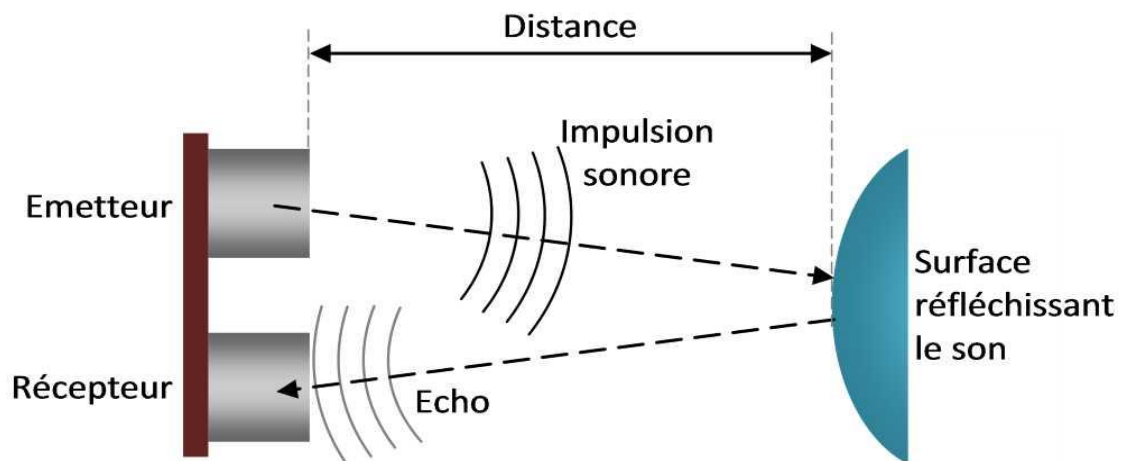


Figure 32 : principe de fonctionnement d'un capteur ultrasonique

I.2.1.6 Considérations pratique :

En pratique, pour assurer un bon fonctionnement du capteur il faut respecter les conditions suivantes :

- L'éloignement des objets ne doit pas dépasser 4m.
- Il faut prendre aussi en considération que le capteur ne peut pas détecter les objets qui sont à des angles raides.
- Si un objet est très petit au point de ne pas refléter les ondes incidentes, alors il est indétectable par le capteur.
- A grande distance, la surface de l'objet à détecter doit mesurer au moins 0.5 m^2 .
- Un objet ayant une surface molle ou irrégulière peut dans certains cas ne pas être détecté (à noter que la surface de l'eau est détectable donc faire attention à ne pas utiliser ce type de capteurs dans des environnements humides).
- La température a un effet sur la vitesse du son dans l'air, si on connaît la température on peut obtenir la vitesse du son avec la formule suivante :

$$C_{\text{air}} = 331.5 + (0.6 * T_c) \text{ [m/s]} \quad (2.1)$$

Tel que :

- C_{air} : est la vitesse du son dans l'air.
- T_c : est la température de l'air.
- Seuls les objets perpendiculaires à un rayon émis sont détectables; l'onde émise est typiquement de 30° de cône d'ouverture.

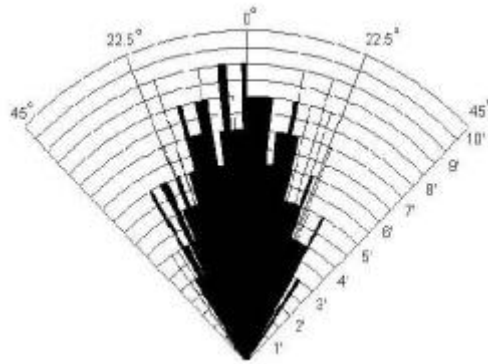


Figure 33 : Angle d'ouverture du capteur HC-SR04.

I.2.1.7 Distance de l'objet :

La distance parcourue par un son se calcule en multipliant la vitesse du son, environ 350 m/s, (à une température de 30°) ou 35'000 cm/1'000'000 μ s par le temps de propagation, soit : $d = v \cdot t$ (distance = vitesse * temps). On sait aussi que le son fait un aller-retour, donc la distance vaut la moitié.

$$d = (35'000 \text{ cm}/1'000'000 \mu\text{s}) * (\text{valeur} / 2) \text{ en simplifiant } d = 175'00 / 1'000'000 \text{ cm} * \text{valeur}$$

$$\text{Finalement} \quad d = (175/10\,000 \text{ cm}) * \text{valeur} \quad (2.2)$$

La formule $d = \text{durée}/58 \text{ cm}$ figure aussi dans le manuel d'utilisation du HC-SR04 car la fraction 175/10 000 est égale à 1/58. Elle donne cependant des résultats moins précis [27].

I.2.2 Les motoréducteurs :

Un motoréducteur est un moteur à courant continu attaché à un réducteur, nous aborderons ces deux notions dans les paragraphes ci-dessous. Les motoréducteurs sont les actionneurs les plus utilisés en robotique, car ils permettent d'obtenir un bon rapport entre vitesse et couple.

I.2.2.1 Les moteurs à courant continu :

Tous d'abord, un moteur est un composant de conversion d'énergie électrique en énergie mécanique. Dans notre cas, les moteurs à courant continu transforment l'énergie électrique en énergie mécanique sous forme de rotation. A noter que les moteurs à courant continu peuvent aussi servir de générateur d'électricité en convertissant une énergie mécanique (qui est la rotation du moteur) en énergie électrique. C'est un des points à ne pas négliger car si un

moteur est actionné à l'envers il va générer du courant, donc il faut bien protéger son circuit contre ces injections d'énergie non désirées qui proviennent de cette conversion.

Il existe deux types de moteurs à courant continu :

- Moteur à courant continu avec balais.
- Moteur à courant continu sans balais.

I.2.2.1.1 Les moteurs à courant continu avec balais :

Ce sont des moteurs de ce genre qui sont généralement utilisés dans les motoréducteurs que nous même nous utilisons dans notre projet, ils sont très rentables et sont très efficaces pour ce que nous voulons en faire, les moteurs à courant continu sans balais, eux, sont utilisés lorsque l'on a besoin d'une grande vitesse de rotation ; exemple pour faire tourner les hélices d'un drones.

Les moteurs à courant continu aussi appelé Machine à Courant Continu, sont composés principalement de deux parties :

- **Le rotor (induit)** : c'est la partie tournante constitué d'électro-aimants (bobines)
- **Le stator (inducteur)** : c'est la partie statique, il peut être composé d'aimant permanent.

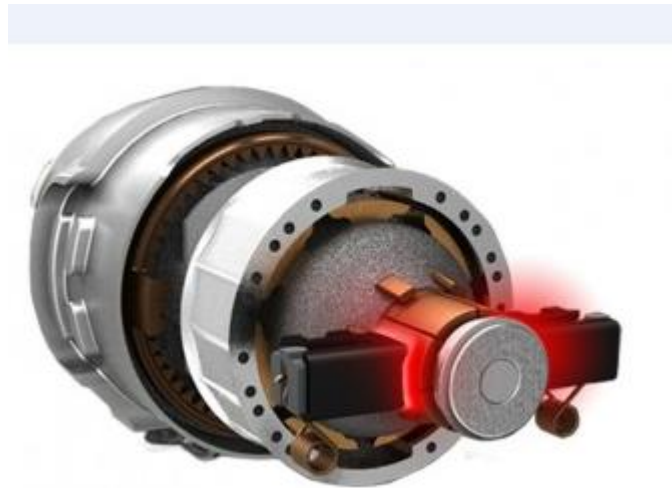


Figure 34 : Moteur à courant continu avec balais.

I.2.2.1.1.2 Principe de fonctionnement :

Le moteur à courant continu dispose d'un bobinage (induit) placé dans un champ magnétique produit par le stator (inducteur). Lorsque le courant circulant dans les spires de l'induit via un dispositif adapté (balais, collecteur) produit des forces électromagnétiques opposé au champ magnétique du stator, ceci génère une répulsion entre les deux dispositifs, donc d'une autre manière engendre le mouvement de rotation du moteur.

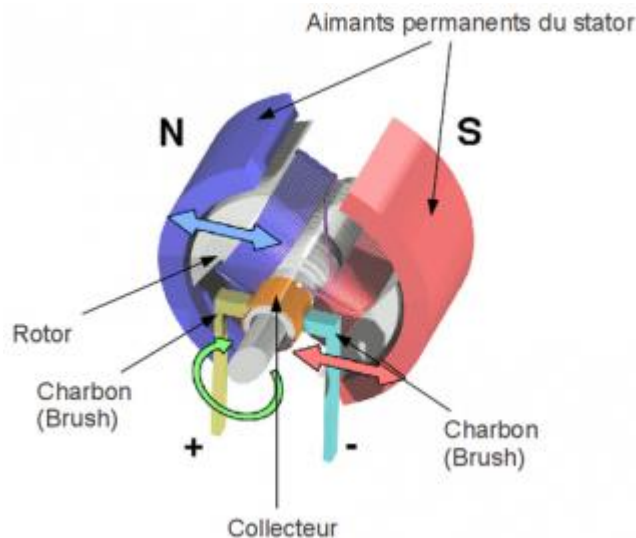


Figure 35 : Vue interne d'un moteur à courant continu.

A noter : la figure est juste un schéma simplifié, d'un moteur à courant continu avec deux bibines, car en pratique ce genre de moteur n'existe qu'avec trois bobines ou plus.

Par ailleurs, les balais qui appuient sur le collecteur grâce à un système à ressorts sont en graphite pour permettre le passage du courant. Le collecteur, lui, est en cuivre, il ne s'use pratiquement jamais contrairement aux balais. De ce fait, le moteur à courant continu avec balais collecteur nécessite donc une maintenance régulière. D'autre part, le passage du courant à travers le contact glissant provoque parfois des étincelles, ce qui a pour effet de limiter l'usage de ces moteurs dans certains environnements.

I.2.2.1.2 Les moteurs à courant continu sans balais :

L'application de ces moteurs (héritée de l'anglais DC brushless motors, voir figure 36) est trompeuse. Il ne s'agit pas en effet de moteurs à courant continu, mais de moteurs à courant alternatif ; ce sont des moteurs synchrones autopilotés. Ils furent développés et commercialisés à partir des années 60, notamment pour pallier aux défauts des moteurs à courant continu, principalement dus au frottement des balais sur le collecteur.

Les moteurs à courant continu sans balais ont ainsi un meilleur rendement, et de meilleures propriétés mécaniques.

I.2.2.1.2.1 Principe de fonctionnement :

Un moteur à courant continu sans balais dispose d'une configuration opposée à celle du moteur avec balais, en effet son bobinage est placé sur la partie fixe (stator), mais l'aimant permanent devient dans ce cas précis la partie mobile (rotor). Le champ magnétique produit par le bobinage du stator engendre une interaction avec celui de l'aimant, ce qui a pour conséquence de faire tourner le moteur.



Figure 36 : Moteur à courant continu sans balais.

Du fait qu'il n'y ait pas de frottements entre le rotor et le stator, leur confère une plus grande fiabilité et un temps de vie supérieur. Ils occasionnent moins de bruit de communication et aucune étincelle. L'absence de bobinage au rotor permet souvent un refroidissement par simple convection, sans circulation forcée d'air vers l'extérieur. Ceci peut donc permettre une meilleure étanchéité. La vitesse de rotation reste limitée mécaniquement par les liaisons pivot entre bâti et rotor, et par l'échauffement qui a un effet sur l'efficacité des aimants. Ces moteurs présentent toutefois certains défauts, tels des effets d'ondulation de couple aux basses vitesses (effet cogging) [25].

1.2.2.2 La mécanique liée au moteur :

Nous allons détailler quelques notions de mécanique liées aux moteurs.

1.2.2.2.1 Le couple :

Le couple est une notion un peu dure à comprendre. L'unité du couple est le Newton mètre (Nm), Cette unité nous informe de deux choses : le couple est à la fois lié à une distance (le mètre) mais aussi à une force (le Newton), le couple s'exprime par rapport à un axe. On peut en conclure que le couple est la capacité du moteur à faire tourner quelque chose sur son axe [25]. Plus le couple est élevé et plus le moteur sera capable de mettre en mouvement une charge plus importante. Ce qui nous permet d'introduire la formule suivante :

$$C = F * R \quad (2.3)$$

Avec :

C: le couple, en Newton mètre.

F: la force exercée, en Newton.

R : le rayon de l'action en mètre.

A noter : Plus le courant délivré au moteur est important, plus son couple sera élevée. D'une autre manière, l'intensité du champ magnétique devient d'autant plus importante sur une bobine à chaque fois que le courant l'est aussi.

I.2.2.2 La vitesse de rotation :

La vitesse de rotation est mesurée par rapport à l'axe de rotation du moteur. Imaginons que le moteur entraîne son axe lorsqu'il est alimenté par un courant, ce dernier va avoir une vitesse de rotation. Il peut tourner lentement ou rapidement. On mesure une vitesse de rotation en mesurant l'angle en radians parcourus par cet axe pendant une seconde. C'est à dire que le moteur est en fonctionnement, que son axe tourne et que l'on mesure jusqu'où va l'axe de rotation, à partir d'un point de départ fixe, en une seconde. On exprime la vitesse de rotation en radians par seconde rad/s.

A noter : Plus la tension délivrée au moteur est importante, plus sa vitesse sera élevée [25].

I.2.2.3 Les réducteurs :

Un moteur électrique est bien souvent très rapide en rotation. Hors nous avons besoin de faire un robot qui ne va pas trop vite, il va falloir faire en sorte de réduire sa vitesse de rotation. On peut très bien mettre un "frein" qui va empêcher le moteur de tourner vite, ou bien le piloter. Cela dit, même si on réduit sa vitesse de rotation, le moteur ne va pas pouvoir supporter des charges lourdes. Autrement dit, notre robot ne pourra même pas se supporter lui-même ! Nous avons donc besoin de couple. Et pour avoir du couple, tout en réduisant la vitesse de rotation, on va utiliser ce que l'on appelle un réducteur. Un réducteur est un ensemble composé d'engrenages qui permettent de réduire la vitesse de rotation de l'axe du moteur tout en augmentant le couple de sortie.

Les réducteurs consistent en un ensemble de roues engrenant les unes sur les autres, grâce au profil de leurs dentelures qui peuvent être droites ou hélicoïdales. Ces réducteurs offrent un rapport de réduction égal au rapport du nombre de dents des roues dentées. La plus petite des roues, nommée pignon, est montée sur l'arbre du moteur, tourne plus vite que la plus grande qui est nommée roue. Il est fréquent de préférer les engrenages à denture hélicoïdales aux engrenages à denture droite pour des raisons de bruit et de contraintes mécaniques.



Figure 37 : Réducteurs à dentures droite et hélicoïdale.

Bien que l'on puisse utiliser des roues possédant des diamètres très différentes pour augmenter le taux de réduction, il est généralement souhaitable de mettre plutôt plusieurs étages de réduction pour obtenir ce résultat de manière beaucoup plus compacte, on peut ainsi retrouver une sortie concentrique, ce qui est aussi un avantage.

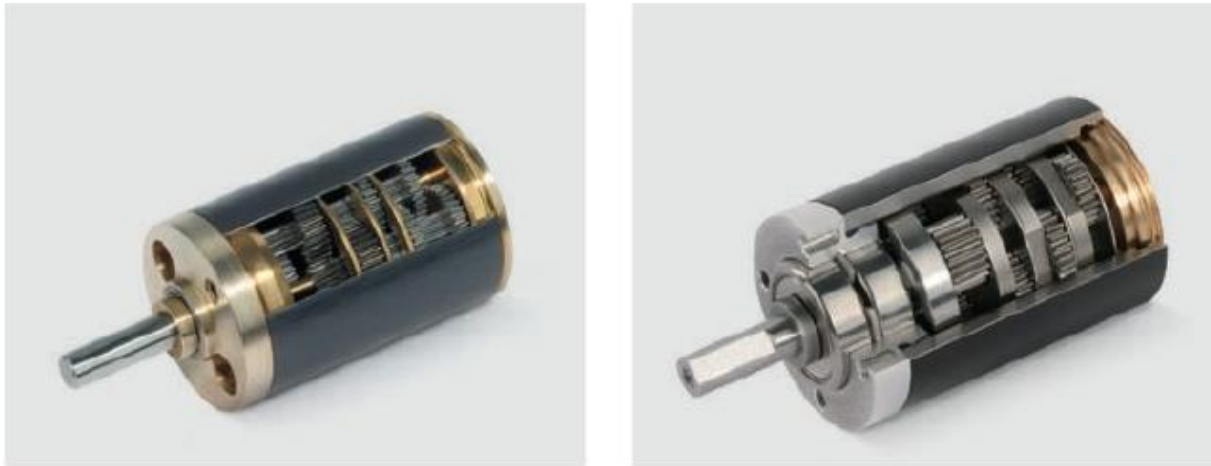


Figure 38 : Réducteurs Maxon compacts à dentures droite (gauche) ou à train épicycloïdal (droite).

Le but de rajouter un réducteur pour un moteur à courant continu est donc d'avoir un couple élevé, cette notion est très importante en robotique mobile car on a besoin d'avoir des moteurs qui peuvent supporter le poids du robot et d'avoir une certaine précision qui permet d'exécuter certaines instructions dans les délais afin d'éviter les obstacles.

Et le rapport de réduction est calculé avec la formule suivante :

$$R = \frac{V_e}{V_s} = \frac{C_s}{C_e} \quad (2.4)$$

Avec :

R : le rapport de réduction du réducteur.

V_e : la vitesse de rotation de l'axe en entrée du réducteur.

V_s : la vitesse de rotation de l'axe en sortie du réducteur.

C_e : couple exercé par l'axe en entrée du réducteur.

C_s : couple exercé par l'axe en sortie du réducteur.

Un réducteur donc est un système qui modifie deux grandeurs qui sont liées : le couple et la vitesse. On peut schématiser son fonctionnement de la manière suivante :

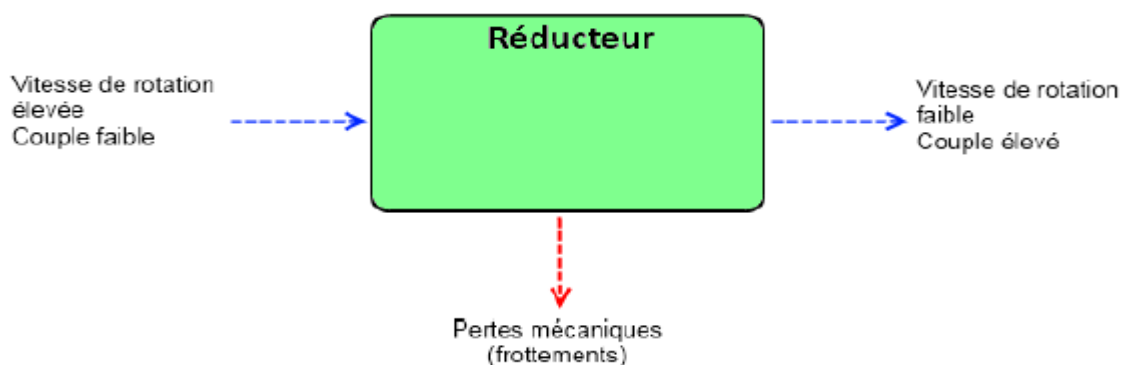


Figure 39 : schéma de fonctionnement d'un réducteur.

1.2.2.4 Commande des moteurs à courant continu :

Pour faire tourner un moteur à courant continu avec l'Arduino Uno :

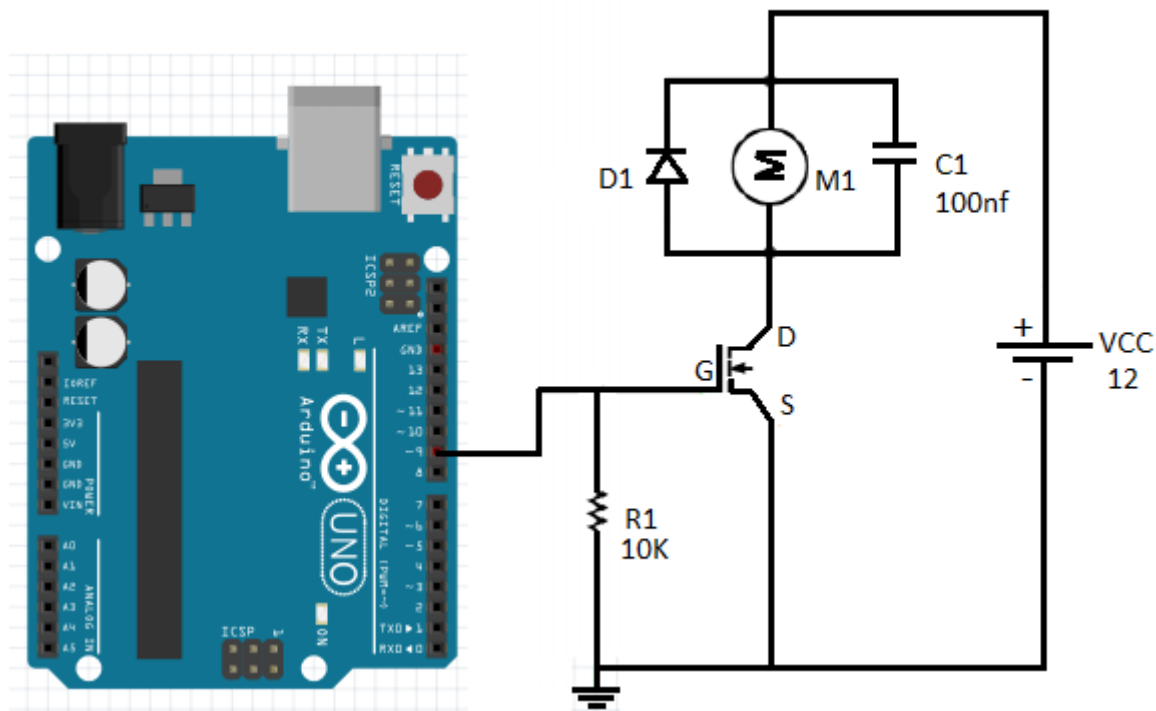


Figure 40 : montage de commande d'un moteur via la carte Arduino.

Comme nous pouvons le voir sur la figure 40, on ne peut pas brancher directement le moteur à la carte Arduino, et cela vaut également pour n'importe quel autre système à microcontrôleurs, il est en effet indispensable de réaliser ce montage car les moteurs à courant continu sont de véritables sources de parasites qui pourraient gravement endommager la carte Arduino. Quand les moteurs à courant continu tournent, ils génèrent tous un tas de parasites qui peuvent être des surtensions très grandes, de plus le courant qu'ils demandent est bien trop grand par rapport à ce que peut fournir une sortie numérique de l'Arduino.

Vue que les sorties numériques de l'Arduino ne peuvent pas fournir un courant suffisamment fort pour pouvoir commander le moteur, on se doit d'avoir recours à un étage de puissance. Par exemple dans le schéma précédent on a utilisé une source d'alimentation externe et un transistor de type MOSFET optimisé à fonctionner à des régimes de fort courant. Leur fonctionnement est fort simple, une différence de potentiel sur la broche « gate » et il commute (laisse passer le courant entre le « drain » (D) et la « source » (S) sinon il est bloqué, en plus on utilise une résistance R1 de tirage vers le bas (pull-down) pour bloquer le transistor si le signal n'est pas défini sur la broche de l'Arduino alors par défaut le moteur ne tournera pas. Pour pallier au problème de parasites l'on utilise un condensateur qui sera monté en parallèle avec le moteur ainsi qu'une diode de roue libre pour la dissipation.

Dans le cas de ce montage, nous pouvons faire tourner le moteur dans un seul sens de rotation, pour changer de sens on a besoin d'un moyen d'inversion de la polarisation du

moteur d'une manière automatique, et voire même être capable de freiner le moteur. Pour cela on utilise un composant très fréquent dans la robotique, c'est ce qu'on appelle le pont en H (H bridge en anglais).

Pour pouvoir aller dans les deux sens, il nous faut donc faire appel à ce dispositif qui est capable de faire passer le courant dans le moteur dans un sens ou dans l'autre, la figure 41 représente un pont en H, il est à noter qu'à la place des interrupteurs, nous aurons des transistors de puissance [25].

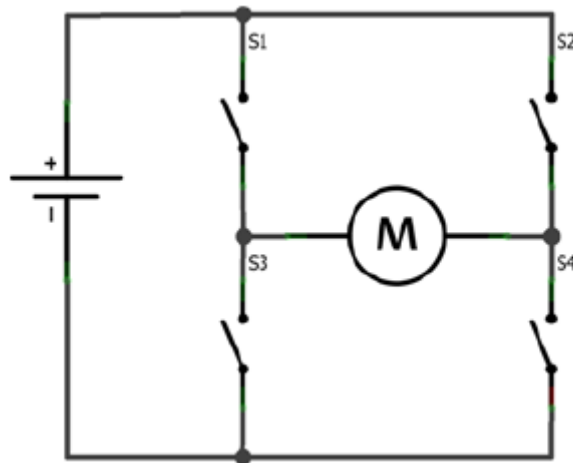


Figure 41 : Pont en H.

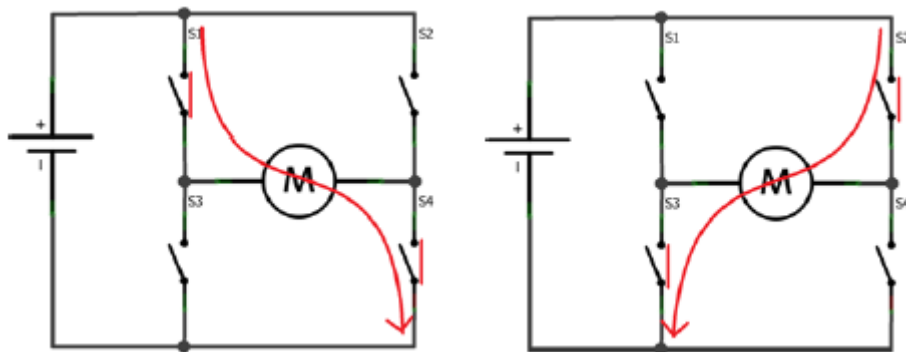


Figure 42 : illustre la manière dont il faut alimenter le pont en H pour faire tourner le moteur dans les deux sens.

Les transistors fonctionnent par paire de deux et jamais deux transistors d'un même côté ne doivent être fermés simultanément, cela ferait un court-circuit de l'alimentation et nos transistors risqueraient de griller immédiatement si l'alimentation est capable de fournir un courant plus fort que ce qu'ils ne peuvent admettre. Maintenant, que se passe-t-il lorsqu'on décide de fermer 1 et 2 en laissant 3 et 4 ouverts ? Cette action va créer ce que l'on appelle un frein magnétique. Comment cela fonctionne lorsque l'on met une diode de roue libre aux bornes du moteur ? Le moteur se retrouve alors court-circuité. En tournant à cause de son inertie, le courant généré va revenir dans le moteur et va le freiner. Attention cependant, c'est différent d'un phénomène de roue libre où le moteur est libre de tourner.

I.2.2.5 Les protections nécessaires :

I.2.2.5.1 Les diodes de roue libre :

Comme nous l'avons vu plus haut, pour protéger un transistor des parasites ou lors du freinage électronique du moteur, nous plaçons une diode. Dans le cas présent, cette diode devra être en parallèle aux bornes du transistor (voir la figure qui suit). Ici nous avons quatre transistors, nous utiliserons donc quatre diodes que nous placerons sur chaque transistor. Ainsi, le courant trouvera toujours un moyen de passer sans risquer de forcer le passage dans les transistors en les grillant. Les diodes de type Schottky sont recommandées pour leurs caractéristiques de tension de seuil faible et commutation rapide.

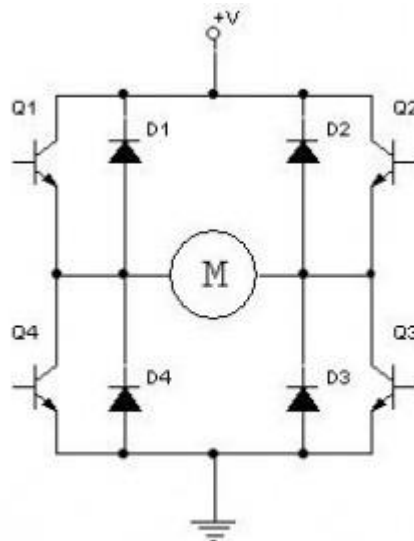


Figure 43 : montage pont en H diode de roue libre.

I.2.3 Les servomoteurs :

I.2.3.1 Définition et principe de fonctionnement :

Les servomoteurs sont des moteurs particuliers, car contrairement aux autres types de moteur, leur rôle n'est pas seulement de tourner mais aussi de maintenir une position donnée. Le mouvement de sortie est une rotation qui entraîne un changement de position de l'axe du servomoteur. Ils intègrent dans un même boîtier un moteur à courant continu jumelé avec un train d'engrenage démultipliant (réducteur), qui entraîne à son tour un axe avec une grande force de torsion mais avec une vitesse réduite (la vitesse est inversement proportionnelle au couple), ils embarquent aussi un capteur de retour de position (généralement un potentiomètre), ainsi qu'un dispositif électronique de commande qui permet la correction en temps réel de la position de l'axe pour qu'elle soit égale à la consigne émise par le microcontrôleur, ainsi qu'un hacheur permettent de réaliser la commande adéquate, les servomoteurs sont donc des systèmes asservis (les figures 44 et 45)représentent respectivement le schéma synoptique du fonctionnement de l'asservissement du servomoteur, et une vue intérieur d'un servomoteur) [25].

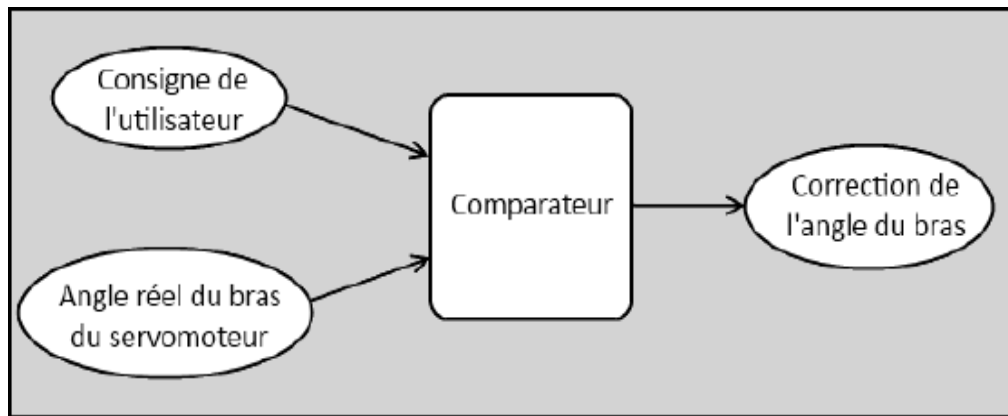


Figure 44 : Synoptique de fonctionnement de l'asservissement du servomoteur

En général les servomoteurs ont un angle de rotation limité à 180° et cela à cause d'un mécanisme interne de blocage (butée). Cela étant dit, on peut passer outre ce blocage soit en modifiant un servomoteur équipé de ce mécanisme, mais il existe aussi des servomoteurs à rotation continue ils sont appelés servotreuil.

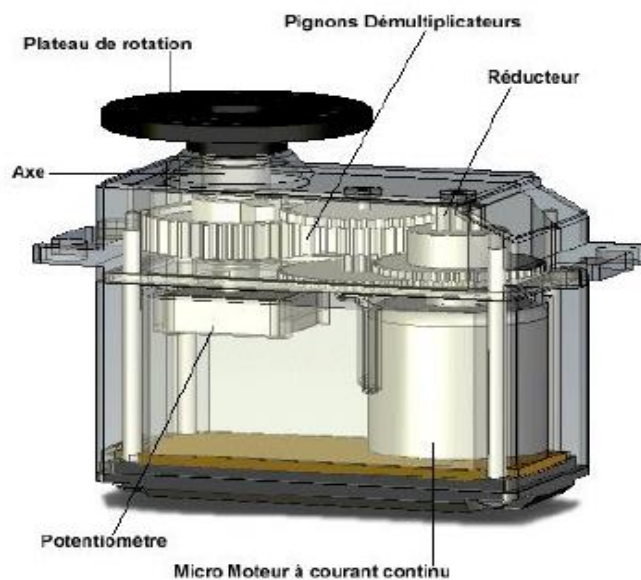


Figure 45 : Vue intérieur d'un servomoteur

I.2.3.2 La mécanique :

Comme on le voit sur la figure précédente, le servomoteur possède plusieurs pignons (engrenages) en sortie du petit moteur CC. Cet ensemble est ce qui constitue le réducteur. Ce réducteur fait deux choses : d'une part il réduit la vitesse de rotation en sortie de l'axe du servomoteur (et non du moteur CC), d'autre part il permet d'augmenter le couple en sortie du servomoteur (là encore non en sortie du moteur CC). Alors, à quoi ça sert de réduire la vitesse et d'augmenter le couple ? Les moteurs CC se débrouillent très bien pour tourner très vite mais lorsqu'ils font une si petite taille ils sont bien moins bons pour fournir du couple. On va donc

utiliser ce réducteur qui va réduire la vitesse, car nous n'avons pas besoin d'avoir une vitesse trop élevée, et augmenter le couple pour ainsi pouvoir déplacer une charge plus lourde.

Conformément à la formule :

$$R = \frac{\omega_{entree}}{\omega_{sortie}} = \frac{C_{sortie}}{C_{entree}} \quad (2.4)$$

Le rapport de réduction R du réducteur définit le couple et la vitesse de sortie (en sortie du réducteur). Selon la vitesse et le couple d'entrée (en sortie du moteur CC) [25].

Sur le datasheet de chaque servomoteur il est mentionné le couple qu'il délivre, qui est calculé par la formule :

$$C = F * R \quad (2.5)$$

C : est le couple du servomoteur (Kg.m).

F : est la force exercée sur le bras du servomoteur (Kilos).

R : la distance (en m).

Le couple (C) exercé sur un axe est égal à la force (F) appliquée au bout du levier accroché à ce même axe. À force identique, plus le levier est long et plus le couple exercé sur cet axe est important. En d'autres termes, si un servomoteur dispose d'un bras d'un mètre de long il aura beaucoup plus de difficultés à soulever une charge de 10g, que son homologue qui supporte la même charge avec un bras de 10 centimètres. Prenons l'exemple d'un servomoteur que nous utiliserons dans notre travail le SG90. Sa documentation nous indique que lorsqu'il est alimenté sous 4.8V il peut fournir un couple (*torque* en anglais) de 1.8 Kg.cm. C'est à dire, qu'au bout de son bras, s'il fait 1 centimètre, il pourra soulever une charge de 1.8kg. Si le bras fait 10 centimètres, l'on perdra 10 fois la capacité à soulever une masse, on se retrouve alors avec un poids de 180g, (voir figure 46).

Ainsi donc on peut calculer avec la formule suivante la charge maximale que peut supporter le servomoteur :

$$P_{max} = \frac{C}{d} \quad (2.6)$$

P_{max} : est le poids maximal de charge en Kg.

C : couple du servomoteur, en Kg .cm.

d : distance à laquelle le poids est placé en cm.

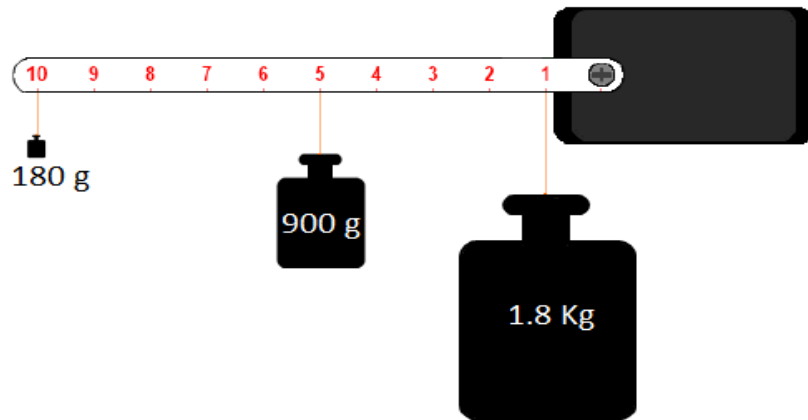


Figure 46 : La variation du poids que peut supporter un servomoteur SG90.

I.2.3.3 La connectique :

Le servomoteur a besoin de trois fils de connexion pour fonctionner. Deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande :

Rouge : pour l'alimentation positive (4.5V à 6V en général).

Noir ou **Marron** : pour la masse (0V).

Orange, Jaune, Blanc, ... : entrée du signal de commande.

I.2.3.4 Le signal de commande des servomoteurs :

La consigne envoyée au servomoteur n'est autre qu'un signal électronique de type PWM (modulation largeur d'impulsion). Il dispose cependant de deux caractéristiques indispensables pour que le servomoteur puisse comprendre ce qu'on lui demande. À savoir : une fréquence fixe de valeur 50Hz et une durée d'état HAUT fixée à une certaine limite comme l'illustre la figure 47.

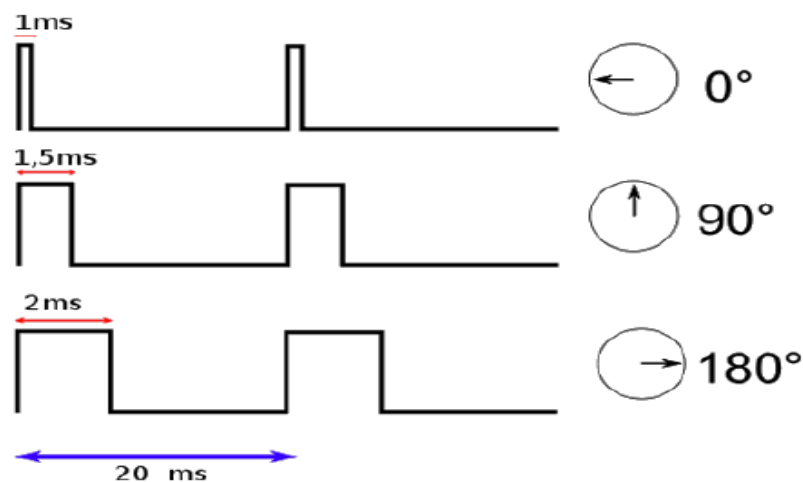


Figure 47 : commande de la position du servomoteur.

Nous pouvons d'abord noter que les impulsions PWM doivent se répéter à un rythme tel que l'on ne doit pas avoir plus de 10 à 20 ms d'écart entre deux impulsions successives.

- Une impulsion de 1ms de large place le servomoteur dans sa position de 0°.
- Une impulsion de 1.5 ms centre le servomoteur à 90° dite position de repos.
- Une impulsion de 2 ms de large permet de faire tourner le servomoteur à 180° qui est sa position maximal.
- Toute autre largeur d'impulsion, comprise entre 1 et 2 ms, permettent d'obtenir les innombrables positions intermédiaires.

La connexion du servomoteur à la carte Arduino est fort simple comme le montre la figure 48, la seule précaution à prendre se situe au niveau de l'alimentation du servo dont la masse doit être commune avec celle de l'Arduino. Par contre, la tension positive ne doit pas être prélevée sur le 5V destiné à l'Arduino car, d'une part la consommation du servo est trop importante pour le régulateur intégré se trouvant sur la carte Arduino, d'autre part chacun des mouvements provoque des appels de courant susceptibles de perturber la carte en causant une baisse anormale de sa tension d'alimentation.

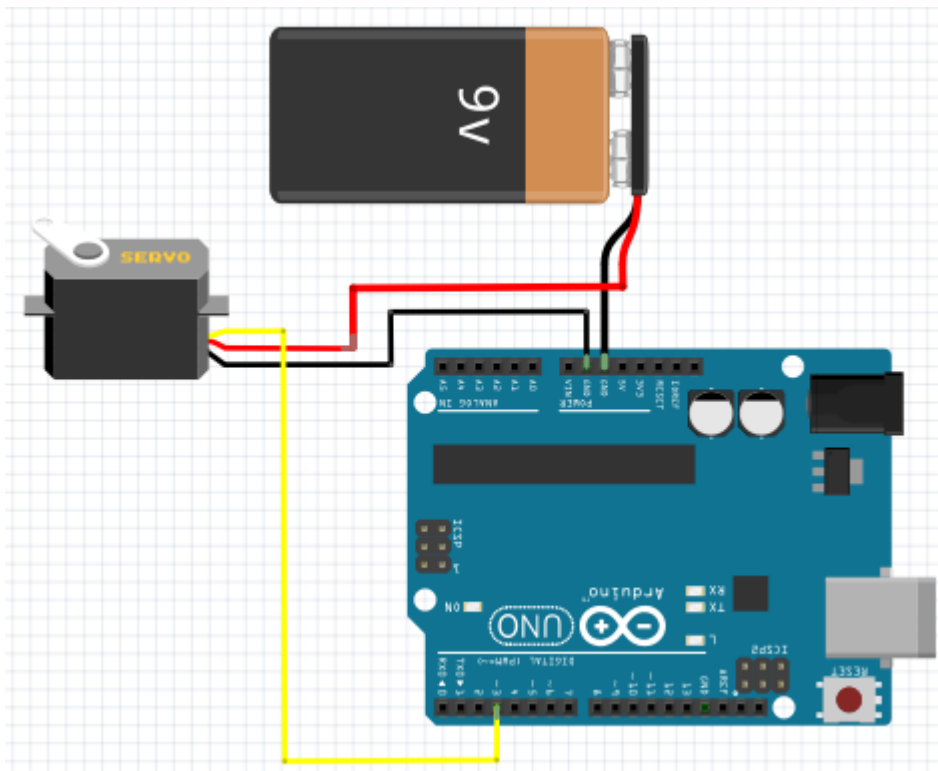


Figure 48 : Connexion du servomoteur à l'Arduino.

I.2.3.5 La librairie Servo :

Pour utiliser le servo avec Arduino, il va nous falloir générer le signal PWM vu précédemment. C'est à dire créer un signal d'une fréquence de 50Hz et modifier l'état haut d'une durée comprise entre 1 et 2ms. Pour sa programmation l'équipe d'Arduino a implémenter une classe portant le nom « Servo » l'objet servo va nous permettre d'utiliser les servomoteurs. Cette bibliothèque est tellement performante qu'elle nous permet en effet de commander jusqu'à 12 servos avec l'Arduino Uno [25].

I.2.4 La conception informatique de notre robot :

I.2.4.1 Arduino :

Une équipe de développeurs composée de *Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti* a imaginé un projet répondant au nom de Arduino et mettant en œuvre une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout un chacun dans le but de créer facilement des systèmes électroniques.

Arduino a été créé en Italie à l'interaction Design Institute Ivrea en 2005 comme un moyen de prototypage, destiné aux étudiants, dans un but précis pallier les manques des étudiants en matières d'électronique et de prototypage. Le nom « Arduino » lui vient du nom d'un bar où les fondateurs du projet Arduino ont l'habitude de se rencontrer, qui à son tour, a été nommé d'après Arduin d'Ivrée, roi d'Italie de 1003 à 1014 ap. J-C.

I.2.4.2 Une carte électronique

Une carte électronique est un support plan, flexible ou rigide, généralement composé d'époxy ou de fibre de verre. 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. Évidemment, tous les composants d'une carte électronique ne sont pas forcément reliés entre eux. Le câblage des composants suit un plan spécifique à chaque carte électronique, qui se nomme le schéma électronique.

I.2.4.3 La carte Arduino Uno :

Dans notre travail nous utilisons la carte Arduino UNO (voir figure 49) qui est un outil de développement à base d'un microcontrôleur ATmega328 d'ATmel.



Figure 49 : carte Arduino UNO Rev 3.

Dès son arrivée sur le marché, la carte Arduino atteint une grande communauté, car elle était dotée d'une flexibilité jamais vue alors. Avec le temps cette carte a commencé à changer et à s'adapter aux nouveaux besoins et défis auxquels elle fut confrontée, la rendant de plus en plus utilisée et prisée à travers les années, dans toutes sortes de projets, que ce soit pour des projets amateurs ou pour des projets scientifiques complexes.

Maintenant la question qui se pose est pourquoi nous avons choisi la carte Arduino et non une autre carte de développement, car il en existe plusieurs, telles les cartes de Microchip à base de microcontrôleurs PIC, ou encore les cartes Parallax Basic Stamp, Phidgets, Netmedia's BX-24, etc. Comme toutes ces cartes la carte Arduino Uno simplifie grandement l'utilisation de son microcontrôleur et elle offre d'autres avantages non négligeables :

- Multiplateforme : l'IDE Arduino qui sert à la programmation des cartes Arduino marche pratiquement sur tous les systèmes d'exploitation que ce soit Windows ou Macintosh OSX ou Linux, par contre les autres systèmes à microcontrôleurs ce limitent généralement à Windows seulement.
- Open source software : la nature open source permet le développement de bibliothèques par les utilisateurs expérimentés en programmation.
- Open source hardware : vu que tous les schémas de l'Arduino sont disponibles au grand public, cela permet aux utilisateurs de créer leur propre carte Arduino dans le but de mieux comprendre son fonctionnement, ainsi que de pouvoir créer ce qu'on appelle les shield (carte d'interface en français).
- L'environnement de développement d'Arduino est basé sur le projet Processing [18], qui permet d'avoir une compatibilité avec les langages C/C++ et Java, ce qui permet une adaptation rapide pour la majorité des utilisateurs (car les langages C/C++ et le Java sont des langages de programmation très populaires et même si on ne les maîtrise pas parfaitement ils sont très faciles à comprendre et à mettre en œuvre).
- On peut programmer la carte Arduino directement via le port USB inclus à la carte, sans programmeur externe, et l'effacer tout autant à contrario d'autres microcontrôleurs.
- La carte Arduino Uno que nous utilisons est équipée d'un microcontrôleur ATmega 328, 32 signifie que compte 32 Kilos mots, le 8 signifie que chaque mot est codé sur 8 bits donc ce microcontrôleur a une capacité de 32 Ko de programme, il utilise par défaut un Bootloader de 0.5 Ko, il nous reste 31.5 Ko ce qui est très largement suffisant dans la plupart des cas [24], [29].

I.2.4.3.1 Caractéristiques :

Les caractéristiques de la carte Arduino Uno :

- Microcontrôleur ATmega 328 de (32 Ko).
- 14 Pins digitaux en entrée/sortie, dont 6 sont utilisés comme PWM (Pulse Width Modulation ou en français Modulation de largeur d'impulsion).
- 6 Pins analogiques en entrée /sortie.
- Un quartz de fréquence 16 MHz.

- Un connecteur USB pour la communication avec un ordinateur.
- Un connecteur POWER JACK pour l'alimentation externe (7-12V).
- Un connecteur mâle à six broches ICSP pour la programmation série de l'Arduino (ils permettent de se passer du port USB).
- Un bouton reset.

Dans la figure 50 qui représente un schéma explicite de cette carte, nous pouvons y observer l'emplacement des différents éléments que nous venons de citer ainsi que d'autres éléments.

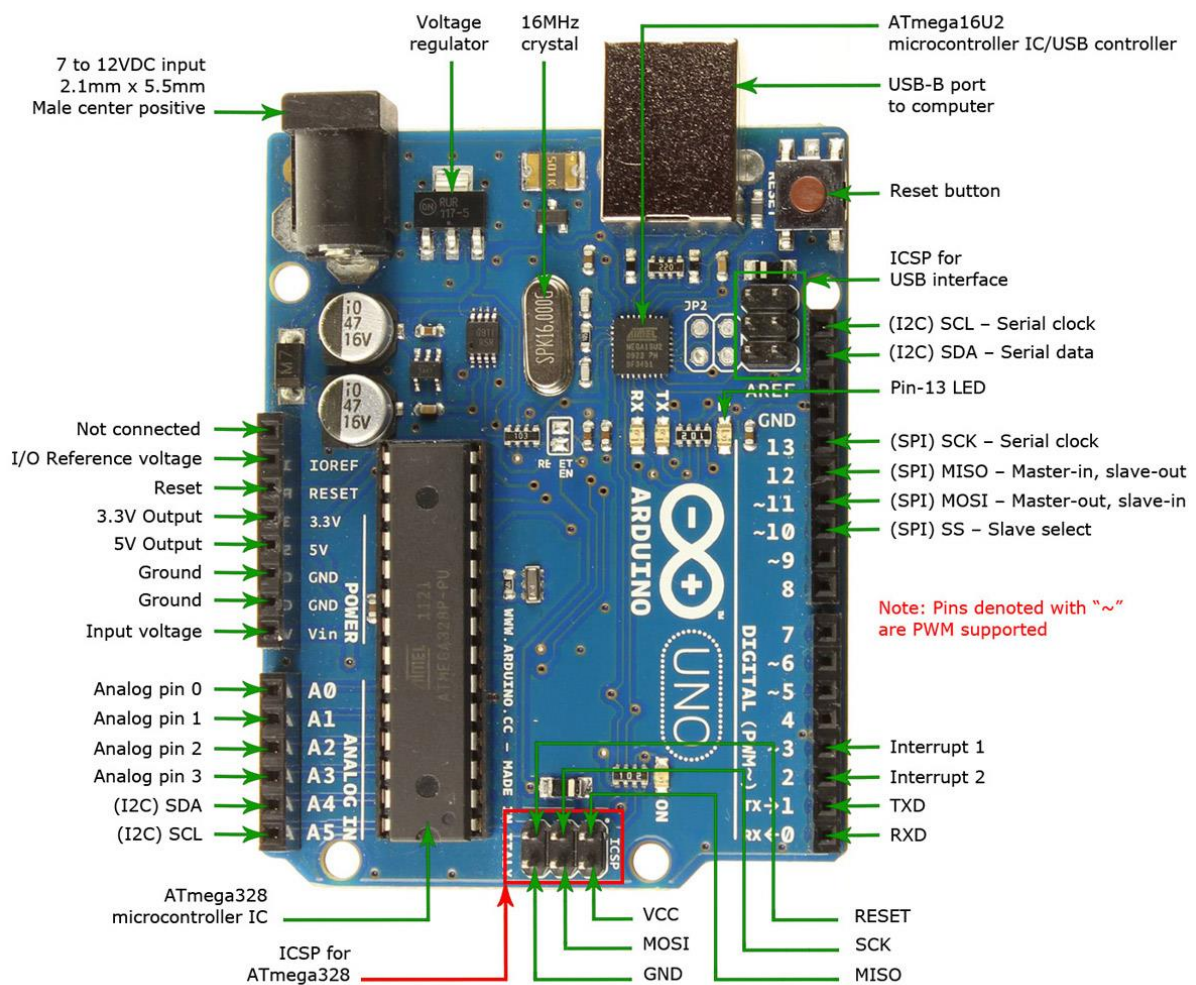


Figure 50 : schéma explicite de la carte Arduino Uno Rev 3.

Le tableau ci-dessus donne toutes les références hardware de l'Arduino Uno Rev3 :

Microcontrôleur	ATmega 328
Tension de fonctionnement	5V
Tension d'entrée (recommander)	7-12V
Tension d'entrée limite	6-20V
Entrée/sortie Digitale	14 (dont 6 fonctionne en PWM)
Entrée Analogique	6
Courant par broche I/O	40mA (Attention : 200mA cumulé pour l'ensemble des broches E/S)
Courant par broche en 3.3V	50Ma
Intensité maxi disponible de la sortie 5V	fonction de l'alimentation 500 mA max si port USB est utiliser seul
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le Bootloader
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Vitesse d'horloge	16 MHz
Longueur	68.6 mm
Largeur	53.4 mm
Poids	25 g

Tableau 4 : références hardware de l'Arduino Uno Rev3.

I.2.4.3.2 Alimentation de la carte Arduino Uno :

La carte Arduino Uno peut être alimentée soit via le port USB (A-B) ou par une source d'alimentation externe, la source d'alimentation est sélectionnée automatiquement.

En ce qui concerne la source d'alimentation externe il lui faut un adaptateur AD vers DC ou encore une batterie.

La carte opère avec une alimentation externe de 6 à 20 volts, mais la tension fournit est inférieur à 7V la broche de 5V peut ne pas fournir les 5V dans certains cas, ce qui aura comme répercussion une instabilité de la carte. Dans le cas où on utilise plus de 12V, le régulateur de tension va se chauffer rapidement ce qui va endommager la carte, c'est pour cela qu'il est recommandé d'opérer avec une tension entre 7 à 12V.

I.2.4.3.3 Les broche de l'Arduino :

La carte Arduino Uno dispose de 32 broches, nous allons détailler leurs fonctions dans ce qui va suivre :

I.2.4.3.3.1 Les broche d'alimentation :

- **VIN** : C'est la tension d'entrée de la carte lorsque l'on utilise une source d'alimentation externe. On peut fournir la tension d'entrée via cette broche. Si la tension est fournie via le power jack on peut l'obtenir grâce à cette broche.
- **5V** : Cette broche fournit une sortie de 5V stabilisée, qui est généralement utilisée pour alimenter de petits dispositifs tels (capteur, module Bluetooth,...etc.). Si on alimente la carte via cette broche, on contourne le régulateur de tension et cela peut endommager la carte sauf si on dispose d'une alimentation déjà stabilisée.
- **3.3V** : Cette broche fournit une tension de 3.3V comme son nom l'indique, mais au contraire de toutes les autres broches qui ne délivrent que 40mA celle-ci en délivre/épuise 50mA. Comme la broche 5V cette broche contourne également le régulateur de tension et cela peut donc aussi endommager la carte, donc faire très attention lors de l'utilisation de ces deux broches.
- **GND** : broche de masse.
- **IOREF** : Cette broche fournit la tension de référence avec laquelle le microcontrôleur de la carte opère. Les shield utilisent cette broche pour déterminer quelle source d'alimentation à utiliser.

I.2.4.3.3.2 Les broches d'entrées/sorties :

- Les broches numériques (digitales) :

Chacune des 14 broches digitales de la carte Arduino Uno peuvent être utilisées soit en entrée ou en sortie, elles opèrent avec une tension de 5V et utilisent 40mA et elles ont des résistances internes de (20-50K Ω) (pull-up resistors) qui sont déconnectées par défaut.

Certaines de ces broches ont des fonctionnalités spéciales :

- **Communication serial 0 (RX) et 1 (TX)** : utilisé pour recevoir la broche (RX) et pour envoyer la broche (TX) les données TTL série. Ces broches sont connectées aux broches correspondant sur la puce ATmega 8U2USB vers TTL.
- **Interruptions externes 2 et 3** : ces broches peuvent être configurées comme source d'interruptions soit sur un niveau logique bas ou haut, ou bien encore à des fronts montant ou descendant.
- **PWM 3, 5, 6, 9, 10 et 11 (modulation largeur d'impulsion)** : ces broches fournissent l'option de fonctionnement en PWM l'une des tâches les plus importantes supporter par l'Arduino. Ces broches sont utilisées lors de la commande de (moteur pas à pas, servomoteur, etc.)
- **SPI 10 (SS), 11(MOSI), 12(MISO), 13 (SCK)** : ces broches supportent la communication SPI. Une liaison SPI est un bus de données série synchrone. Les circuits communiquent selon un schéma maître-esclaves, où le maître s'occupe

totale de la communication. Plusieurs esclaves peuvent coexister sur un même bus. On remarque que les broches 10 et 11 peuvent être utilisées en PWM ou dans la transmission SPI.

Le bus SPI utilise quatre signaux logiques :

- **SCK** : Serial Clock, Horloge (généré par le maître).
- **MOSI** : Master Output, Slave Input (généré par le maître).
- **MISO** : Master Input, Slave Output (généré par l'esclave).
- **SS** : Slave Select, Actif à l'état bas (généré par le maître).

- Les broches analogiques :

La carte Arduino Uno dispose de 6 broches analogiques de A0 à A5, chacune d'elles possède une résolution de 10 bits soit (1024 valeurs possibles). Par défauts, elles fonctionnent entre la masse (GND) et 5V, mais il est possible de modifier cette marge en utilisant la broche AREF. Comme pour les broches digitales, certaines de ces broches ont des fonctionnalités spécialisées :

- **TWI A4 (SDA), A5 (SCL)** : ces broches utilisées pour la communication TWI (I2C). Est un bus série synchrone bidirectionnel half-duplex. Qui permet des échanges entre un seul maître et un (ou tous les) esclave(s), toujours à l'initiative du maître (jamais de maître à maître ou d'esclave à esclave).
La connexion est réalisée par l'intermédiaire de deux lignes :
 - **SDA (Serial Data Line)** : ligne de données bidirectionnelle,
 - **SCL (Serial Clock Line)** : ligne d'horloge de synchronisation bidirectionnelle.
- **AREF** : sert comme référence pour les broches analogiques.
- **Reset** : activée au niveau bas, cette broche permet la réinitialisation du microcontrôleur.

I.2.4.3.4 La communication avec l'extérieur :

L'Arduino dispose de plusieurs moyens de communication que ce soit avec une autre carte, ou avec un ordinateur, ou encore un autre microcontrôleur. Le ATmega 328 fournit un moyen de communication série UART TTL (5V) sur les broches 0 (RX) et 1 (TX). Le circuit intégré ATmega 16U2 permet d'avoir à travers le port USB un port virtuel COM pour le logiciel sur l'ordinateur. L'ATmega 16U2 utilise les pilotes USB COM standard, donc pas besoin de pilotes extérieurs. L'IDE d'Arduino contient un moniteur série qui permet d'envoyer et de recevoir des données sous forme de texte simple entre la carte Arduino et un ordinateur. Les LEDs RX et TX flash à chaque fois qu'il y a une transmission qui s'effectue.

Le microcontrôleur ATmega 328 permet aussi d'utiliser la communication I2C (TWI) plus sur les broches A4 qui représente la broche SDA en communication I2C, et A5 qui représente la broche SCL en communication I2C.

On peut aussi utiliser la communication SPI via les broches 10, 11, 12, 13 qui représentent respectivement SS, MOSI, MISO, SCK de la communication SPI.

Afin d'utiliser tous ces protocoles de communication il existe une librairie qui permet l'interfaçage avec l'IDE Arduino. Pour la communication série on utilise la librairie Software Serial [19], pour la communication I2C on utilise la librairie Wire [20], et pour la communication SPI on utilise la librairie SPI [21].

I.2.4.3.5 Gestion d'interruptions :

Une interruption, comme son nom l'indique, consiste à interrompre momentanément le programme que l'Arduino exécute pour qu'il effectue une autre tâche demandée [22].

L'Arduino possède 24 sources possibles d'interruption dont voici les plus utilisées :

- Interruptions liées aux entrées **INT0 (PD2) et INT1 (PD3)**.
- Interruptions sur changement d'état des broches **PCINT0 a PCINT23**.
- Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables)
- Interruptions liée au comparateur analogique.
- Interruptions de fin de conversion **ADC**.
- Interruptions du port série **USART**.
- Interruptions du bus **TWI (I2C)**.

I.2.4.3.6 Timers du microcontrôleur de l'Arduino :

Un Timer est un registre à l'intérieur du microcontrôleur qui s'incrémente (ou se décrémente) chaque fois qu'il reçoit une impulsion d'un signal d'horloge [25].

La carte Arduino Uno qui est construite autour du microcontrôleur AVR ATmega328P qui possède 3 Timers :

- Le **timer0**, sur 8 bits, utilisé par les fonctions delay (), millis () et micros (). Il commande également des PWM (Pulse Width Modulation ou Modulation par Largeur d'Impulsion) sur les broches 5 et 6.
- Le **timer1**, sur 16 bits, qui compte de 0 à 65535 (0 à FFFF en hexadécimal) et qui est utilisé par la bibliothèque Servo ou bien pour de la PWM sur les broches 9 et 10.
- Le **timer2**, sur 8 bits, qui est utilisé par la fonction Tone () ou bien pour de la PWM sur les broches 3 et 11.

I.2.4.3.7 Le microcontrôleur ATmega 328 :

Le microcontrôleur est la pièce essentielle de toute carte électronique, c'est donc le cas pour l'ATmega 328 qui est le cœur de la carte Arduino Uno, il est le cerveau de cette dernière, c'est lui qui décide de tout ce qui se passe sur la carte. C'est un microcontrôleur de la famille AVR 8bits, la signification de ATmega 328 est la suivante :

- **AT** : composant créé par ATMEL.
- **Mega** : il fait partie de la série des megaAVR.
- **328** : il possède 32 KB de mémoire flash, et une CPU à 8 bits.

Tous les ports et broches (voir figure 51) peuvent être configurés en entrée ou en sortie de manière individuelle. Certaines broches ont quelquefois une fonction secondaire, voir même plus dans certain cas. Afin de pouvoir utiliser les fonctions auxiliaires, il faudra toujours les configurer de manière logicielle, les fonctionnalités principales et auxiliaires des différents ports et broches sont représentées ci-dessous :

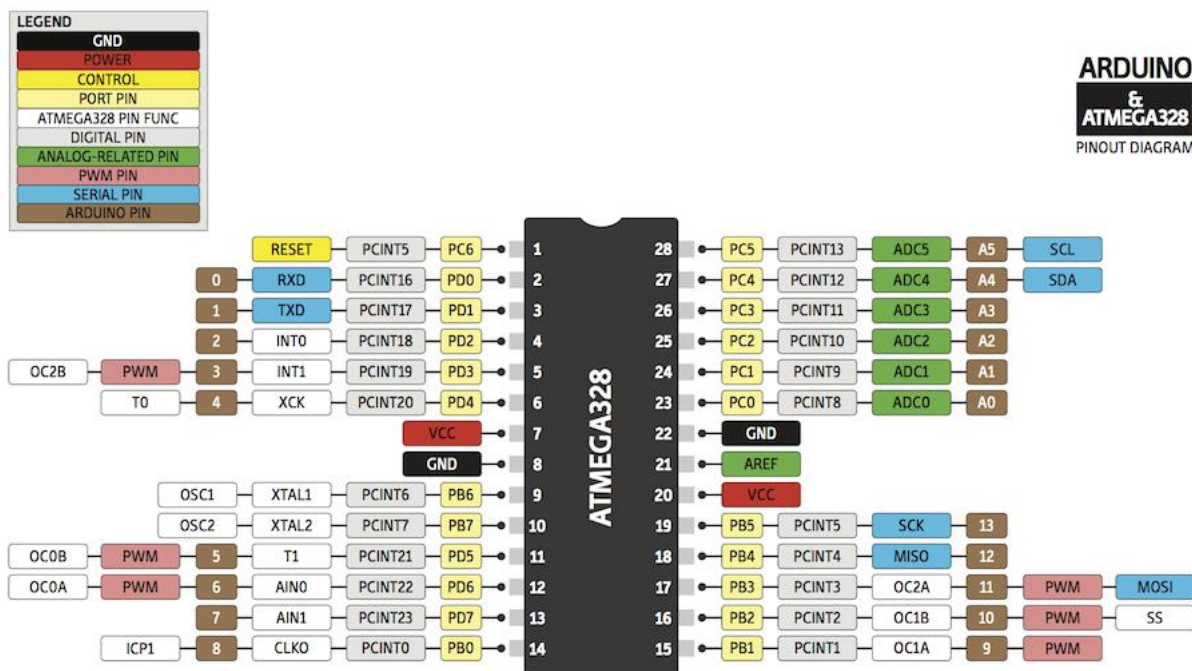


Figure 51: Brochage du microcontrôleur ATmega 328.

- **Port B (PB7...PB0)** le port B est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (rappelle) choisi pour chaque bit. Il sert aussi de comparateur analogique (sortie sur PB2, PB3), ou de SPI.
- **Port C (PC7...PC0)** le port C est aussi un port d'entrée-sortie de 8 bits bidirectionnel avec lui aussi des résistances de tirage pour la sécurité. Il sert aussi comme oscillateur pour le Timer/Compteur 2 et d'interface I2C.
- **Port D (PD7...PD0)** le port D est exactement comme les trois autres. Mais lui il est utilisé pour l'USART et en entrées pour les interruptions externes.
- **RESET** : il est déclenché par un front descendant maintenu pendant plus de 50 ns, il produit le reset du microcontrôleur, même si l'horloge ne court pas.
- **XTAL1** : entrée de l'oscillateur externe ou libre pour l'horloge interne.
- **XTAL2** : production de l'amplificateur d'oscillateur.
- **AVCC** : est une broche de tension d'alimentation pour le convertisseur A/D qui doit être connectée à VCC via un filtre passe-bas pour éviter les parasites.

- **AREF** : est l'entrée de référence analogique pour le convertisseur A/D avec une tension dans la gamme de 2V à AVCC avec un filtre passe bas.
- **AGND** : masse analogique. Si la masse analogique est séparée de la masse générale, brancher cette broche sur la masse analogique, sinon, connecter cette broche à la masse générale GND.
- **VCC** : broche d'alimentation du microcontrôleur (+3 à +5V).
- **GND** : masse de l'alimentation.

Le schéma de l'architecture interne de l'ATmega 328 (voir la figure 52) nous permet de remarquer les fonctionnalités suivantes :

- 2 Timer/Compteur 8 bits avec facteur de pré-division indépendant.
- 1 Timer/Compteur 16 bits avec facteur de pré-division indépendant.
- 1 Horloge à temps réel (RTC) avec un quartz externe.
- 4 canaux PWM.
- 1 Convertisseur Analogique/Numérique 8 canaux avec une résolution de 10 bits.
- 1 Comparateur Analogique à 2 entrées configurables.
- 1 Interface SPI à trois modes sélectionnables (série synchrone).
- 1 interface de communication Asynchrone USART.
- 1 Interface I2C (TWI) pour la gestion d'un bus à 2 fils.
- 1 chien de garde (Watchdog) programmable avec oscillateur interne.
- 1 Port JTAG pour le débogage en temps réel.

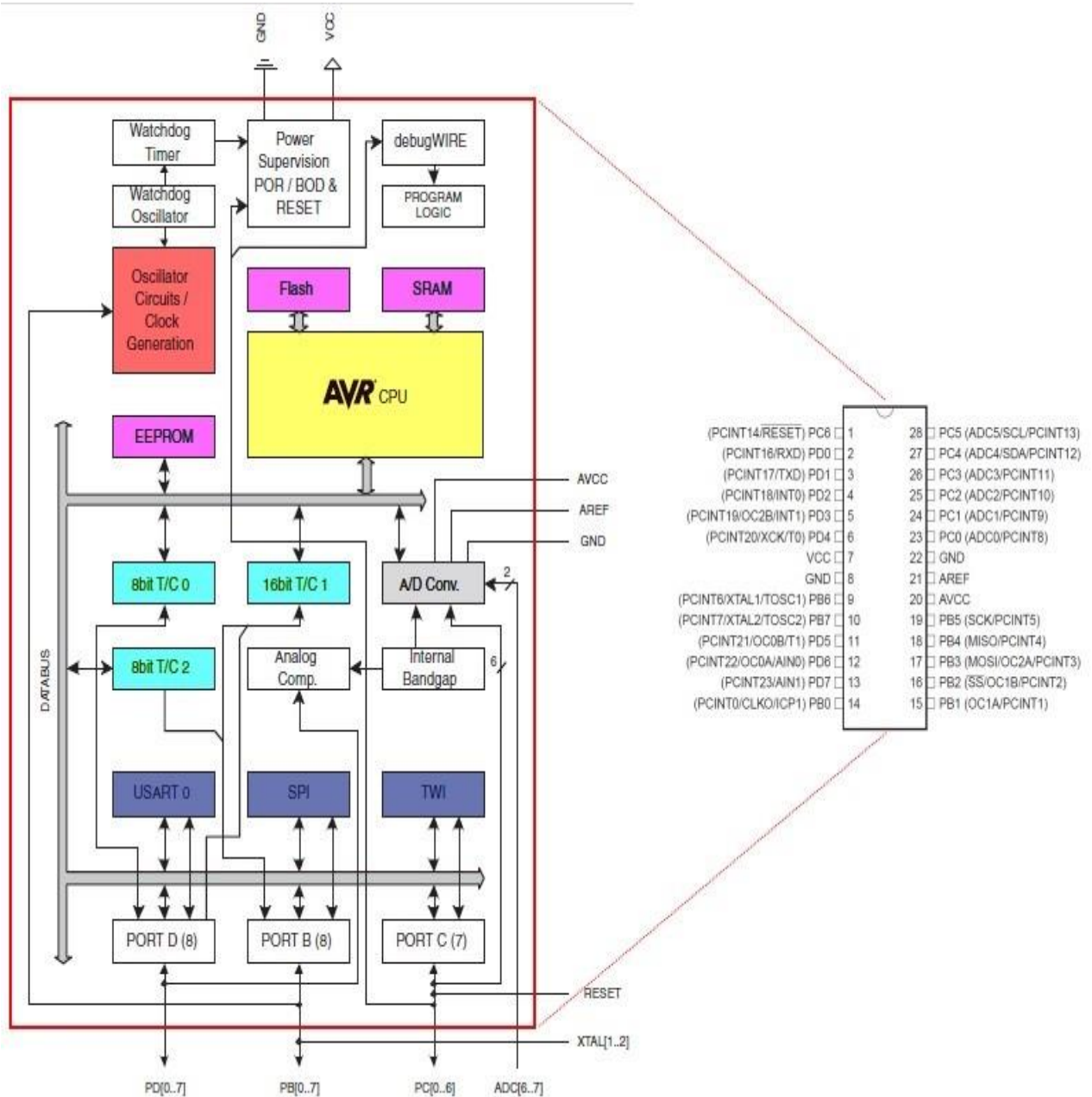


Figure 52 : schéma synoptique du microcontrôleur ATmega 328.

Comme on peut le voir, trois type de mémoires sont utilisées, la mémoire programme Flash, la mémoire de données SRAM, et la mémoire morte de type EEPROM.

- La mémoire programme Flash :** La mémoire programme permet de stocker et de faire fonctionner le microcontrôleur, le nombre d'écriture sur cette mémoire est limité à 10 000 lignes, mais c'est largement suffisant pour la majorité des applications. Sa capacité est de 32Ko, une petite zone de 0,5Ko à la fin de cette espace mémoire est réservée pour le Bootloader qui est le programme qui s'exécute avant le programme écrit par l'utilisateur afin d'effectuer diverses opérations.

- **La mémoire de données SRAM** : La mémoire de données contient les 32 registres de travail, les 64 registres de commande (voir la figure 53), et mémorise aussi toutes les différentes informations temporaire lors de l'exécution. Sa capacité est de 2Ko.
- **La mémoire EEPROM** : La mémoire morte de type EEPROM, l'accès est complexe, elle contiendra la configuration du programme et les données importantes qui seront sauvegardées en absence de courant.

Pour référence le cœur du microcontrôleur ATmega 328 est un processeur AVR dont l'architecture interne est représentée dans la figure suivante :

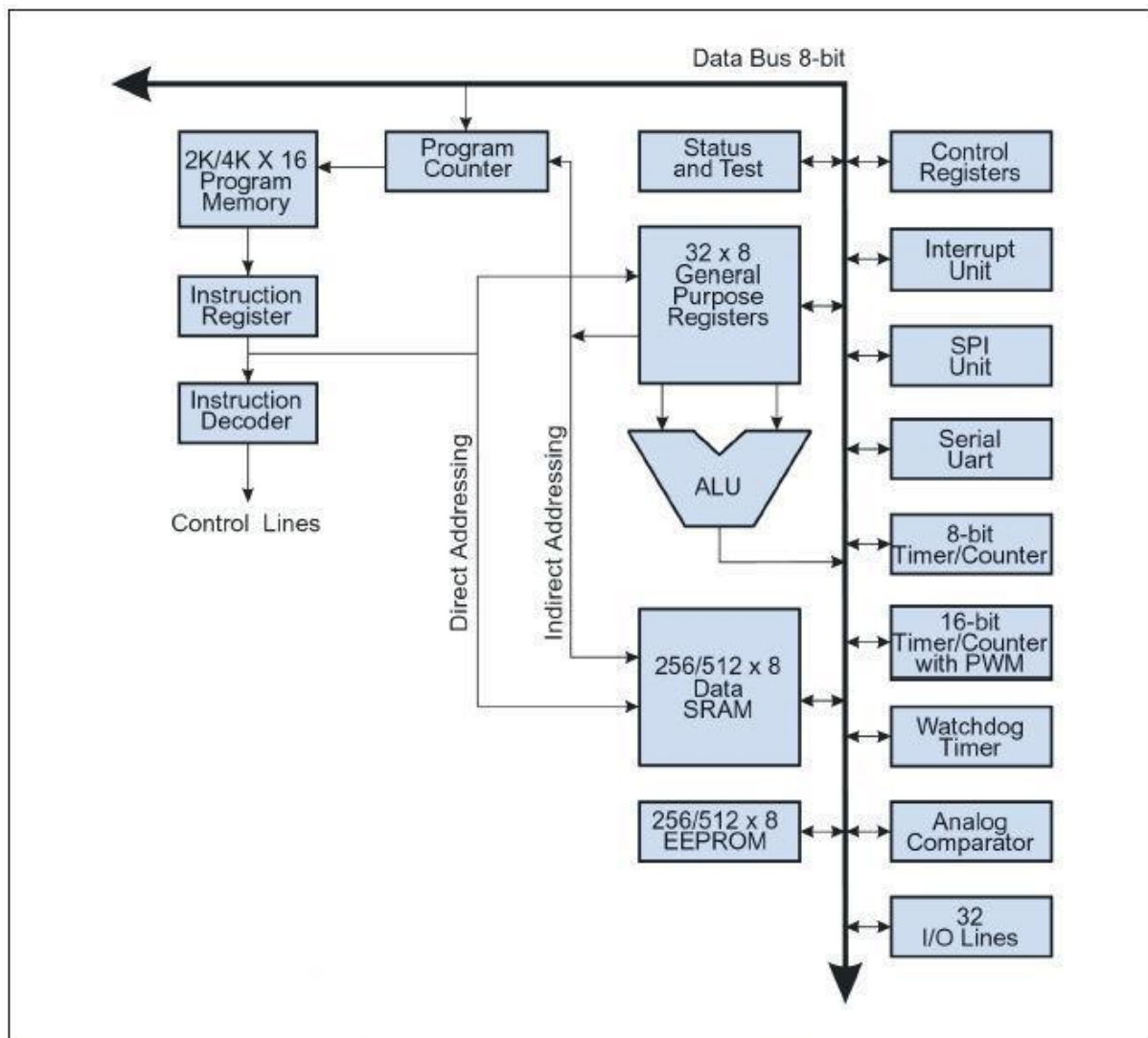


Figure 53 : schéma synoptique de l'architecture AVR.

1.2.5 Les Shields Arduino :

La réalisation d'applications peut être très compliquée, mais afin de rendre c'est derniereplus accessible car tout le monde ne dispose pas du matériel nécessaire pour réaliser ses propres cartes électroniques, Arduino a développé et mis en productions des shields (carte d'extension) pour pouvoir utiliser facilement des composants complexes. Ces extensions possèdent ainsi tout ce qu'il faut pour mettre en œuvre nos applications.

L'un des intérêts majeurs de l'utilisation des cartes Arduino est donc le fait qu'il existe ces cartes d'extension, capable de couvrir la majorité des besoins d'une application, elles permettent entre autres le rajout des fonctionnalités à la carte Arduino, afin de réaliser des applications plus poussées et évoluées, que ce soit en terme de capacité de communication via Ethernet, Wifi, bluetooth, ou encore Xbee, ainsi que d'autres capacités comme la commande de moteurs, etc.

Nous présenterons dans cette partie le shield moteur qu'on utilisera dans notre projet.

1.2.5.1 Le Motor Shield :

C'est le shield que nous utiliserons plus tard, donc on le présentera plus en detail, il faut juste savoir pour l'instant que ce shield est très pratique pour les applications robotiques, car il permet de commander à la fois deux moteurs à courant continu ou un moteur pas à pas grâce à un double pont en H (L298),figure 54 représente un Motor Shield Arduino.



Figure 54: Motor shield Arduino.

1.2.6 La communication Wifi :

1.2.6.1Les protocoles de communication :

Ce sont des protocoles de transmission de données nous pouvons citer (le Wifi, le Bluetooth, ZigBee,...). Tous ces protocoles ont l'avantage d'être bidirectionnels.

Le tableau suivant décrit certains protocoles :

Catégorie	Nom	Distance de couverture	Débit maximal théorique	Etat
WPAN (Wireless Personal Area Network)	Bluetooth (IEEE 802.15)	10 m	700 Kbits/s	Très implanté sur les petits terminaux
	ZigBee (IEEE 802.15.4)	Ente 100 m et 1000 m selon la version	200 Kbits/s	La norme en réseaux de capteurs
	IrDA	20 m	1 Mbits/s	Rarement utilisé
WLAN (Wireless Local Area Network)	Wifi (IEEE 802.11)	100 m au Km (omnidirectionnelle)	54 Mbits/s	Très utilisé
	Hiper LAN	100 m	70 Mbits/s	En développement
WMAN (Wireless Metropolitan Area Network)	WiMax (IEEE 802.16)	Quelques km	100 Mbits/s	Disponible depuis 2006

Tableau 5 : les différents protocoles de communication.

I.2.6.2 Téléguidage Wifi :

Afin de pouvoir commander notre robot à distance (donc avoir un feedback), on doit disposer d'un moyen de communication qui soit bidirectionnel et sans fil, c'est pour cela que nous avons choisi dans notre projet de travailler avec la technologie wifi.

I.2.6.3 La technologie wifi :

Le Wifi fait partie d'un ensemble de protocoles de communication permettant l'échange bidirectionnelle de données sans fil, il est régit par les normes du groupe IEEE 802.11 (ISO/CEI 8802-11). Un réseau Wifi permet de relier par ondes radio plusieurs appareils informatiques (ordinateur, routeur, Smartphone, décodeur Internet, etc.) au sein d'un réseau informatique afin de permettre la transmission de données entre eux.

Grâce aux normes Wifi, il est possible de créer des réseaux locaux sans fil à haut débit. En pratique, le Wifi permet de relier des ordinateurs portables, des machines de bureau, des assistants personnels (PDA), des objets communicants ou même des périphériques à une liaison haut débit : de 11 Mbit/s théoriques ou 6 Mbit/s réels en 802.11b, jusqu'à 1,3 Gbit/s théoriques pour le 802.11ac normalisé depuis décembre 2013.

La portée atteint plusieurs dizaines de mètres. Ainsi, des fournisseurs d'accès à Internet peuvent établir un réseau Wifi connecté à Internet dans une zone à forte concentration d'utilisateurs (gare, aéroport, hôtel, train...). Ces zones ou points d'accès sont appelés bornes Wifi ou « hot spots ».

Le terme « Wifi » suggère la contraction de « *Wireless Fidelity* », par analogie au terme « Hifi » pour « *High Fidelity* » (apparu dans les années 1930). Cependant, bien que la Wifi Alliance ait elle-même employé fréquemment ce terme dans divers articles de presse (notamment dans le slogan « *The Standard for Wireless Fidelity* »), selon Phil Belanger, membre fondateur de la Wifi Alliance, le terme « Wifi » n'a jamais eu de réelle signification. Il s'agit bien néanmoins d'un jeu de mots avec « Hifi ».

Le terme « Wifi » a été utilisé pour la première fois de façon commerciale en 1999, et a été inventé par la société *Interbrand*, spécialisée dans la communication de marque, afin de proposer un terme plus attractif que la dénomination technique « IEEE 802.11b Direct Sequence ».

I.2.6.4 Les normes Wifi :

La norme IEEE 802.11 est en réalité la norme initiale publiée en 1997 qui offrait des débits de 1 ou 2 Mbit/s (Wifi est un nom commercial, et c'est par abus de langage que l'on parle de « normes » Wifi). Des révisions ont ensuite été apportées à la norme originale afin d'augmenter le débit (c'est le cas des normes 802.11a, 802.11b, 802.11g, 802.11n et 802.11ac).

Pour des raisons de conformité avec le module ESP8266-07 nous présenterons donc dans le tableau suivant les normes qu'utilise ce module :

Norme	Nom	Description
802.11b	Wifi	La norme 802.11b était la norme Wifi la plus répandue en base installée au début des années 2000. Elle propose un débit théorique crête de 11 Mbit/s (6 Mbit/s réels) avec une portée pouvant aller jusqu'à 300 mètres (en théorie) dans un environnement dégagé. La plage de fréquences utilisée est la bande des 2,4 GHz.
802.11g		La norme 802.11g offre un plus haut débit (54 Mbit/s théoriques, 25 Mbit/s réels) dans la bande de fréquences des 2,4 GHz. La norme 802.11g offre une compatibilité ascendante avec la norme 802.11b. Cette aptitude permet aux équipements de proposer le 802.11g tout en restant compatibles avec les réseaux existants 802.11b. Le principe est le même que celui de la norme 802.11a (bande des 5 GHz).
802.11n	WWiSE (<i>World-Wide Spectrum</i>)	La norme 802.11n est disponible depuis le 11 septembre 2009. Le débit théorique atteint les 450 Mbit/s le débit réel jusqu'à 200 Mbit/s dans un rayon de 100 mètres) grâce aux

	<i>Efficiency)</i>	technologies MIMO (Multiple-Input Multiple-Output) et OFDM (Orthogonal Frequency Division Multiplexing). Le 802.11n a été conçu pour pouvoir utiliser les bandes de fréquences de 2,4 GHz ou 5 GHz.
--	--------------------	---

Tableau 6 : Les normes wifi de l'ESP8266.

I.2.7 Le module ESP8266-07 :

Est un circuit intégré à microcontrôleur avec connexion Wifi développé par le fabricant chinois Espressif au cours de l'année 2014. Donc pour communiquer avec notre robot nous utiliserons ce dernier qui nous permet d'avoir une liaison avec le robot et ainsi lui envoyer et recevoir des données sous forme d'une communication sans fil. Il s'agit de l'ESP8266.

L'ESP8266 est un module Wifi qui peut être utilisé de deux manières :

- associé à un autre microcontrôleur (par exemple un Arduino), l'ESP8266 s'occupera juste de la partie "communication Wifi". L'Arduino enverra des commandes à l'ESP8266 (commandes du style "connecte-toi à tel réseau Wifi", "envoie tel message à tel serveur", etc.). En utilisant l'ESP8266 de cette manière, on est très proche de ce que l'on pouvait faire avec un Shield Wifi.
- L'ESP8266 peut aussi être utilisé de manière totalement autonome, en exécutant lui-même des applications que vous aurez programmées. Là, il devient vraiment puissant. L'ESP8266 ne va pas se contenter d'ajouter des fonctionnalités Wifi à votre Arduino, il va exécuter lui-même le programme Arduino, tout en prenant en charge la partie Wifi. Nous n'uploaderons plus le programme sur l'Arduino, mais directement sur l'ESP8266 [28].



Figure 55 : Module ESP8266-07.

I.2.7.1 Caractéristiques de l'ESP8266 :

- Tension de fonctionnement: 3.3v.
- Protocole Wifi 802.11 b/g/n.
- Fréquence CPU: 80 MHz.
- RAM: 64Ko pour le code, 96Ko pour les données.
- Wifi: b/g/n, WEP ou WPA/WPA2.
- Consommation: entre 60mA et 200mA en fonctionnement normal, quelques dizaines de μ A en veille.
- 11 entrées/sorties digitales.
- Antenne: intégrée.

Propriétés de l'ESP8266 :

Catégorie	Propriétés	Valeurs
Paramètre Wifi	Certificat	FCC/CE/SRRC
	Protocole Wifi	802.11 b/g/n
	Rang de Fréquence	2.4 GHz- 2.5GHz
	Amplification de Tx	802.11 b : +20 dBm
		802.11 g : +17dBm
		802.11 n : +14dBm
	Sensibilité de Rx	802.11 b : -91 dBm
		802.11 g : -75 dBm
		802.11 n : -72 dBm

Paramètres Hardware	Type d'antenne	PCB, connecteur IPEX
	Périphérique bus	UART /SDIO/SPI
		GPIO/PMW
	Tension de fonctionnement	3 à 3.6V
	Courant de fonctionnement	80mA
	Température de fonctionnement	De – 40° à 125°
Paramètres Software	Interface externe	N/A
	Mode Wifi	Station/SoftAP/Station + SoftAP
	Codage	WEP/TKIP/AES
	Protocoles réseau	IPv4, TCP/UDP/HTTP/FTP
	Firmware upgrade	UART Download / OTA

Tableau 7 : Propriétés de l'ESP8266.

I.2.7.2 Les broche de l'ESP8266 :

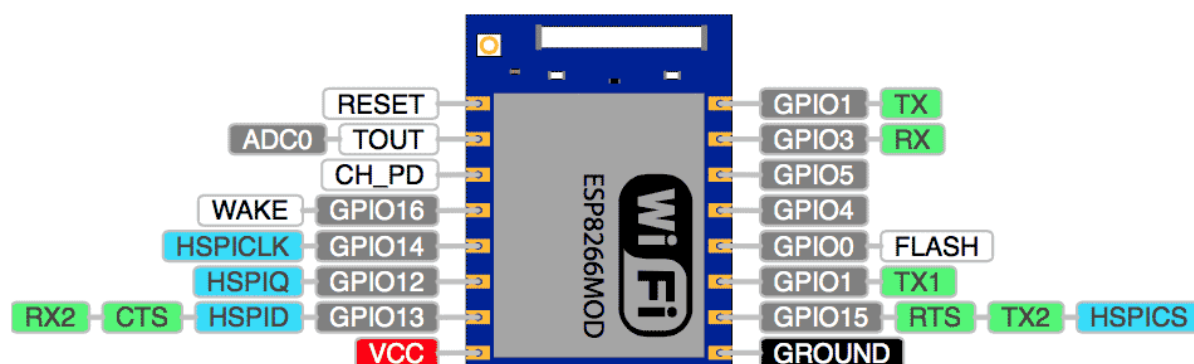


Figure 56 : Broches du module Wifi ESP8266.

Description de chaque Pin et de ses fonctions :

PIN	Nom	Autres fonctions	Utilisation
1	GND		Masse.
2	VCC		Alimentation.
3	UTXD	SPICS1, GPIO1, CLK_RTC	Généralement utilisé comme UART0 série TX.
4	URXD	I2SO_DATA, GPIO3, CLK_XTAL	Généralement utilisé comme UART0 série RX.

PIN	Nom	Autres fonctions	Utilisation
5	GPIO16	XPD_DCDC, RTC_GPIO0, EXT_WAKEUP, DEEPSLEEP	Connecté à la broche XPD_DCDC ESP, peut également être connecté à ESP_EXT_RSTB (reset) broche en fermant le cavalier près de la broche 8.
6	CH_PD		Power-bas: Activation du module wifi.
7	FOURMI		Antenne Wifi.
8	CCV		Entrée de 3.3V (la broche 8 est l'entrée de l'antenne et de la puce ESP)
9	GPIO14	MTMS, I2SI_WS, SP_CLK	Entrée/sortie digitale
10	GPIO12	MTDI, I2SI_DATA, MISO	Entrée/sortie digitale
11	GPIO13	MTCK, I2SI_BCK, MOSI	Entrée/sortie digitale
12	GPIO15	MTDO, I2SO_BCK, SP_CS	Utilisé pour les démarrages spéciaux.
13	GPIO2	I2SO_WS, U1TXD, U0TXD	Utilisé typiquement comme UART1 TX pour l'enregistrement de débogage.
14	GPIO0	SPICS2, CLK_OUT	Bootloader pour l'entrée en mode flash.

Tableau 8 : Description des Pins de l'ESP.

I.2.8 Conclusion :

A travers ce chapitre, nous avons étudié et mis en évidence, les caractéristiques générales de la configuration matérielle de notre robot puis nous nous sommes focalisés d'une manière approfondie sur le principe de fonctionnement de chaque constituant en dressant un tableau détaillé sur leurs spécificités techniques.

Les composants qui ont été l'objet de notre étude sont les suivants :

- **Le capteur HC-SR04** : représente l'élément de perception du robot.
- **Le servomoteur SG90** : constitue le support du capteur afin d'avoir un champ de balayage de 180°.
- **Les motoréducteurs** : est le moyen de locomotion le plus fiable pour avoir un meilleur rapport entre couple/vitesse.
- **Carte Arduino Uno** : c'est la pièce maitresse dans notre projet car elle est le servo et l'élément de commande du robot mobile.
- **Shield Moteur** : est une carte d'interface qui permet de contrôler le sens de rotation des moteurs.
- **L'ESP8266** : est un circuit intégré qui offre un moyen de communication via la création d'un réseau local sans fil WLAN (Wifi).

Dans la partie qui suit nous allons mettre en œuvre toutes les connaissances et compétences acquises durant cette étude afin d'élaborer la réalisation pratique fiable de notre robot mobile autonome.

CHAPITRE III : LA RÉALISATION PRATIQUE

I. Introduction :

Après avoir fait une étude théorique on passe maintenant à la réalisation pratique. En pratique il y a plusieurs défis à relever, on commence par choisir le bon châssis.

Dans le chapitre précédant on a établi une structure pour la conception de notre robot, maintenant il ne reste plus qu'à passer à la mise en œuvre pratique. Pour cela il nous faut tout d'abord établir une méthode à suivre pour la réalisation matérielle et logicielle, par la suite nous procéderons aux tests ainsi qu'à leurs interprétation.

I.1 Architecture du robot mobile :

Toute l'architecture de notre robot s'articule autour de la carte Arduino Uno qui assure l'interfaçage entre les différents organes du robot. L'acquisition des données du monde extérieur étant assurées par le capteur ultrasonore, ainsi que la prise de décision, la figure suivante représente les différentes interactions entre tous les constituants du robot :

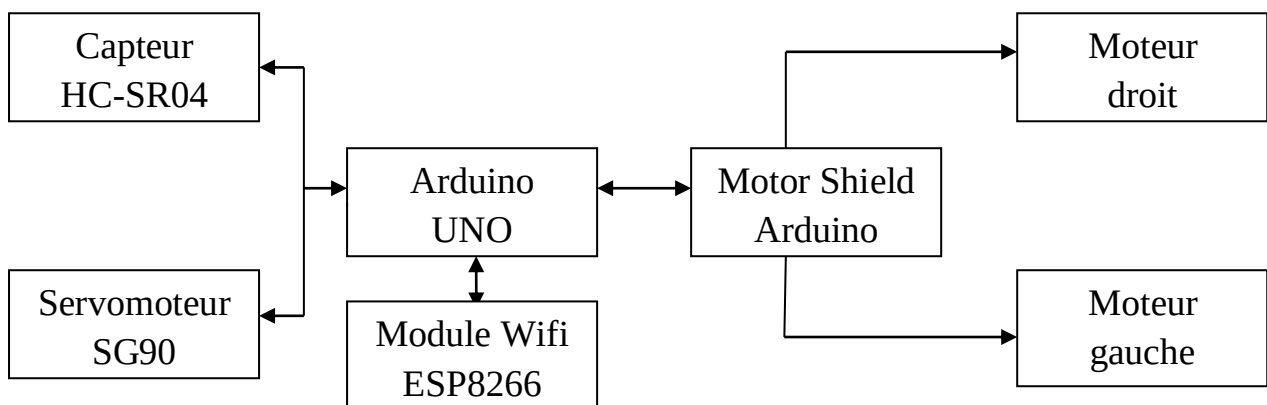


Figure 57 : Architecture de notre robot mobile.

I.2 La réalisation matérielle :

I.2.1 La perception :

La perception est sans doute la partie la plus importante d'un robot mobile autonome, car c'est cette partie de notre robot qui va lui permettre d'éviter les obstacles qu'il rencontrera dans son environnement.

Pour nous, la perception sera assurée grâce à un capteur ultrasonore HC-SR04 dont nous avons traité du principe de fonctionnement dans la partie conception. Il sera ainsi rattaché à un servomoteur (SG90) comme sur la figure 58, le capteur ultrasonore joue le rôle des yeux de notre robot lui permettant ainsi de visualiser son environnement (détecter les obstacles via des ondes ultrason), le servomoteur lui est le cou du robot, il permet au robot de regarder dans toutes les directions sans avoir à faire des rotations sur lui-même.

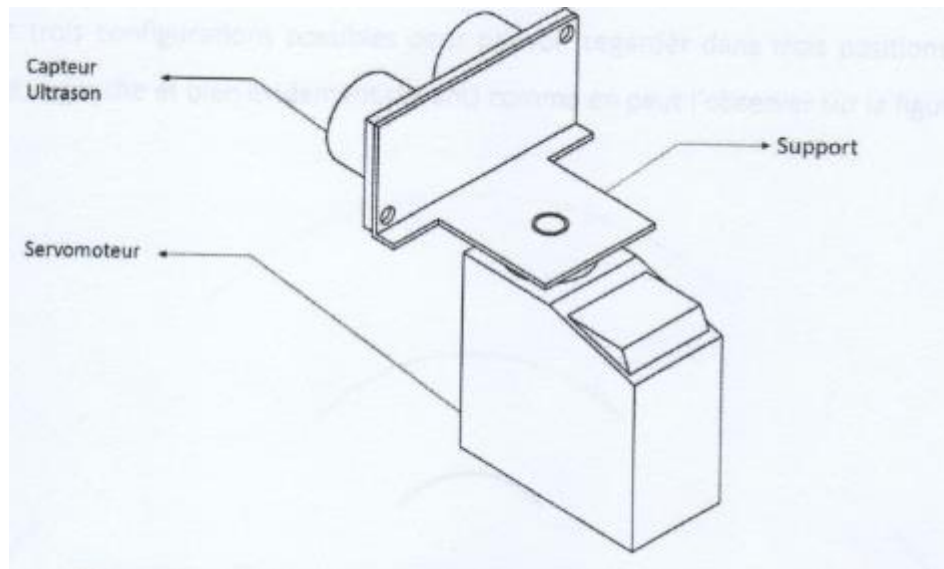


Figure 58 : la partie responsable de la perception du robot.

Le servomoteur SG 90 a les spécifications suivantes :

- Poids de 9g.
- Dimension de 22.2 x 11.8 x 31 mm.
- Vitesse de 0.1 s/60° à 4.8V.
- Couple de 1.8 kgf cm à 4.8V.
- Température de fonctionnement: 0 °C – 55 °C

Son connecteur est représenté dans la figure suivante :

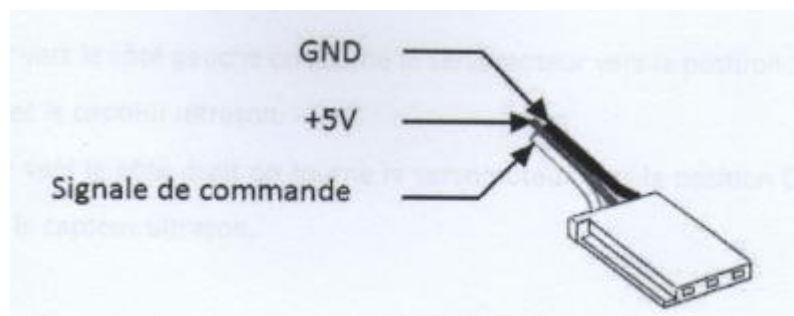


Figure 59 : Connecteur du servomoteur.

La figure 60 représente les trois positions possibles de configuration de notre servomoteur pour pouvoir regarder (vers l'avant, à droite, et à gauche).

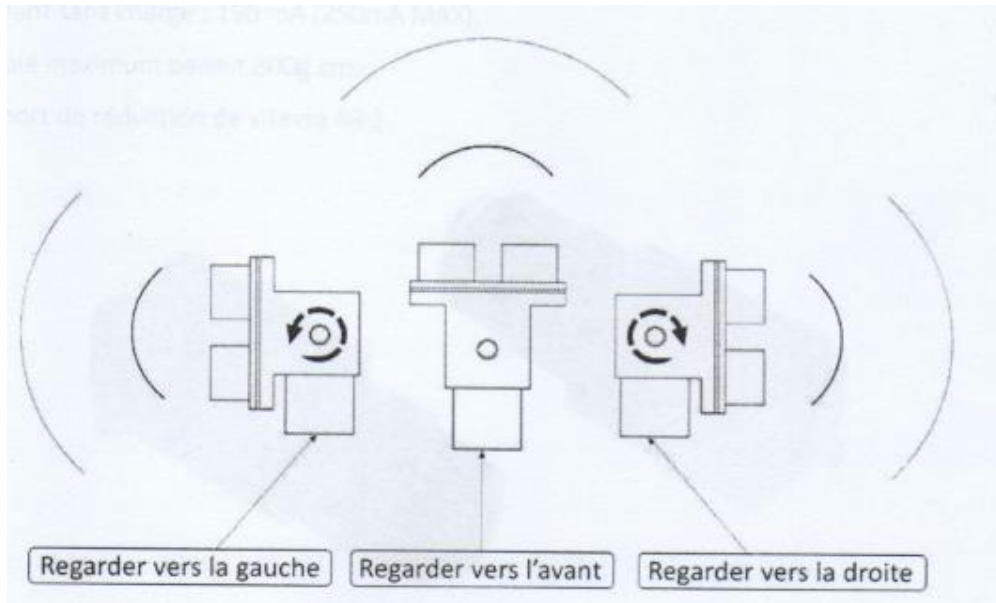


Figure 60 : Les différentes positions de la partie perception de notre robot.

Cette partie fonctionne comme suit :

- Pour regarder vers l'avant on tourne le servomoteur à sa position 90° puis on mesure la distance avec le capteur ultrason.
- Pour regarder vers la gauche on tourne le servomoteur à sa position 180° puis on prend aussi des mesure grâce au capteur ultrasonore, c'est la même chose pour le côté droit sauf qu'il faut tourner le servomoteur à la position 0° .

I.2.2 La locomotion :

I.2.2.1 Les motoréducteurs :

Nous avons déjà présenté dans le chapitre précédant les motoréducteurs ainsi que leur avantage. Pour notre part dans notre travail nous utilisons les motoréducteurs dans la référence est 25 GA 370 (voir figure 61). Ces moteurs ont les caractéristiques suivantes :

- Tension d'alimentation recommandée : 6V.
- Vitesse de rotation à vide : $250 \pm 10\%$ rpm.
- Courant sans charge : 150mA (250mA Max).
- Couple maximum : 1000g.cm.
- Rapport de réduction de vitesse : 35 :1.

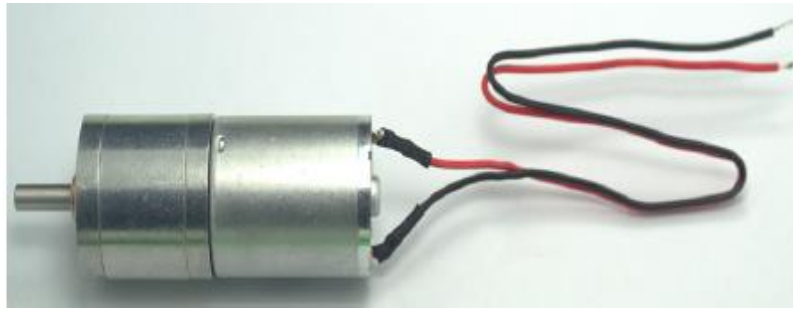


Figure 61 : Motoréducteur 25 GA 370.

I.2.2.2 La commande des motoréducteurs :

Afin de contrôler le déplacement du robot, on doit pouvoir contrôler le sens et la vitesse de rotation des deux moteurs, et cela de manière individuelle comme nous l'avons expliqué dans le chapitre précédent. Donc pour que le robot puisse avancer, les deux moteurs doivent tourner dans le même sens, s'ils tournent dans le sens inverse le robot va reculer. Pour tourner vers la gauche ou vers la droite les moteurs doivent tourner en sens opposés l'un par rapport à l'autre.

Pour contrôler le sens de rotation nous utiliserons un double Pont en H (Dual Full-Bridge driver), le L298 qu'en trouve quasiment dans la majorité de shields moteurs Arduino. C'est une solution très efficace afin d'éviter le recours à beaucoup de branchements qui sont très encombrants et pas toujours appropriés.

I.2.2.3 Le pont H double L298P :

Le L298P est un circuit intégré qui contient 15 broches, il est constitué de deux ponts en H (le pont A et le pont B), idéal dans notre application, il permet un pilotage assez aisé de deux moteurs en même temps. Il fonctionne à de fortes tensions et à de forts courants, il est ainsi capable de débiter jusqu'à 2A nominal par pont et jusqu'à 3A pendant un bref instant. Il propose aussi une fonction pouvant être intéressante qui est la mesure du courant passant au travers du pont (pour vérifier si notre moteur est rendu en butée par exemple). Et une tension de 46V (il peut même aller jusqu'à 50V) par pont. Pour avoir un fonctionnement optimal il est conseillé de placer des condensateurs de 100nf sur chaque ligne d'alimentation (en général on utilise des condensateurs de 100µf ou plus selon l'alimentation, pour mieux découpler cette dernière). Son zéro logique peut aller jusqu'à 1.5V (ce qui présente une grande immunité contre les bruits et les parasites générés par les moteurs à courant continu), sa tension d'alimentation logique est entre 4.5V à 7V.

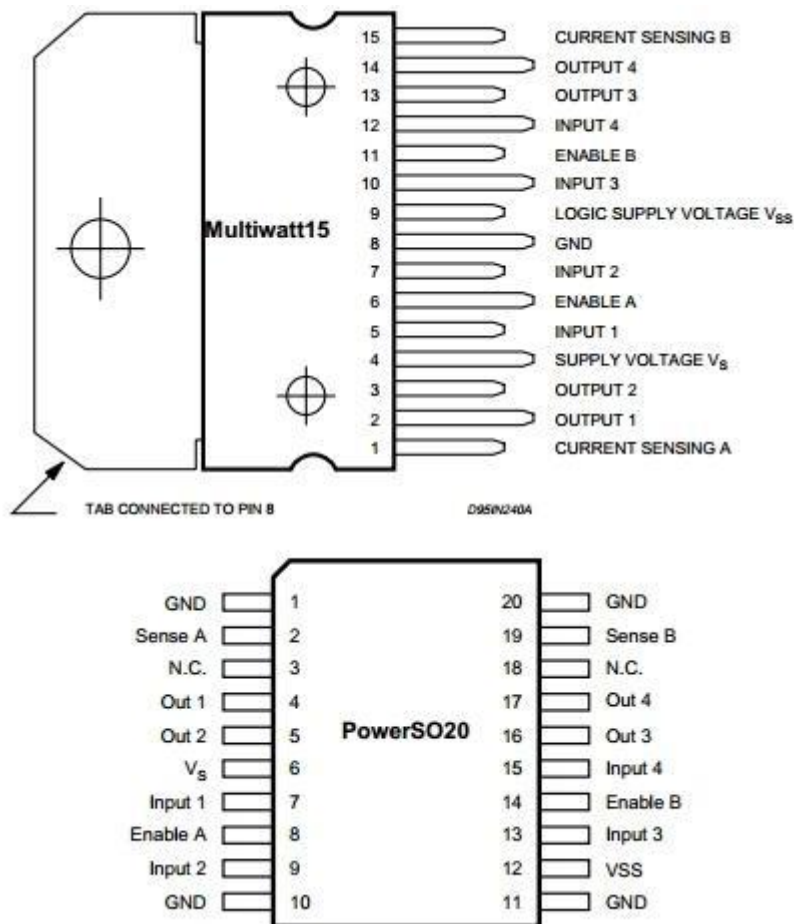


Figure 62 : brochages du circuit intégré L298P.

Dans le tableau ci-dessous nous représentons les différentes fonctions des broches du double pont L298P :

Broches sur le paquet Multiwatt15	Broches sur le paquet PowerSO20	Nom de la broche	Fonction de la broche
1 et 15	2 et 19	Sense A et Sense B	Entre ces broches et la GND on place une résistance pour contrôler le courant de la charge (moteurs dans notre cas).
2 et 3	4 et 15	OUT 1 et OUT 2	Les sorties du pont A : le courant qui traverse la charge qui est connectée entre eux est surveillé par la broche 1.
4	6	V_S	Tension d'alimentation de puissance entre 5V et 46V, un condensateur

			de 100nf doit être connecté entre broche et la GND
5 et 7	7 et 9	Input 1 et Input 2	Ce sont les entrées du pont A (compatible à la technologie TTL).
6 et 11	8 et 14	Enable A et Enable B	Permet d'activer et de désactiver les pont A et B respectivement, ils sont à l'état désactivé au niveau bas.
8	1, 10, 11, 20	GND	La masse.
9	12	VSS	Tension d'alimentation pour le block logique entre 4.5V et 7V. recommander d'utiliser une tension de 5V, et d'ajouter un condensateur de 100nf entre cette broche et la GND
10 et 12	13 et 15	Input 3 et Input 4	Ce sont les entrées du pont B (compatible avec la technologie TTL).
13 et 14	16 et 17	Out 3 et Out 4	Les sorties du pont B : le courant qui traverse la charge qui est connectés entre eux est surveillés par la broche 15
-	3 et 18	Non connecté	Non connecté

Tableau 9 : Les Pins du double pont L298P.

Pour contrôler la vitesse de rotation d'un moteur à courant continu, on envoie des signaux en PWM (Pulse Width Modulation) modulation de largeur d'impulsion, sur les broches Enable A et Enable B (voir la figure 63), ainsi nous pourrions contrôler la position de rotation sur les canaux A et B. Et cela en fonction des largeurs des PWM envoyées.

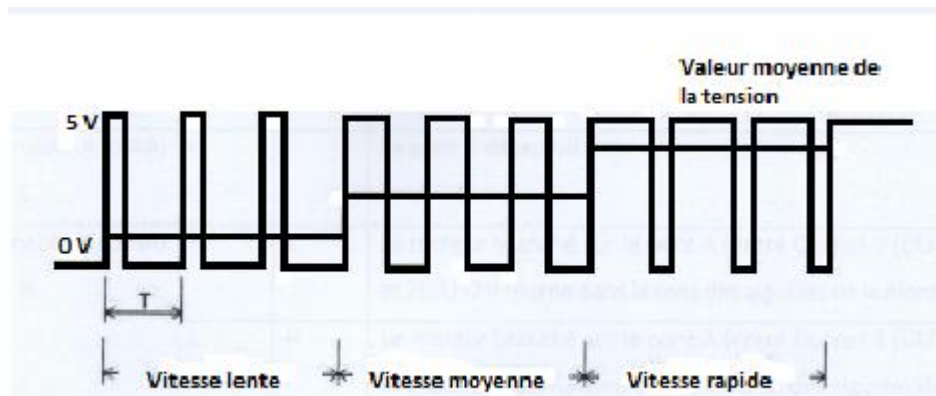


Figure 63 : Contrôle de la vitesse via la PWM.

I.2.2.4 Les PWM :

Nous avons dit qu'il est possible de modifier la vitesse de rotation d'un moteur, pour cela il nous faut utiliser un signal PWM.

Alors comment on s'en sert avec Arduino ? Commençons par dire où sont situées les broches utilisables avec la PWM. Elles sont au nombre de 6 et ont les numéros 3, 5, 6, 9, 10 et 11. Pour les utiliser, nous devons les configurer en sortie dans le `setup ( )` dans le croquis Arduino. Ensuite, on pourra agir sur le rapport cyclique du signal PWM (le ratio entre temps à l'état HAUT et temps à l'état BAS) en utilisant la fonction `analogWrite (broche, ratio)`. L'argument `broche` désigne... la broche à utiliser et l'argument `ratio` indique la portion de temps à l'état haut du signal.

Le rapport cyclique est défini par un nombre allant de 0 à 255. Cela signifie qu'à 0, le signal de sortie sera nul et à 255, le signal de sortie sera à l'état HAUT. Toutes les valeurs comprises entre ces deux extrêmes donneront un rapport cyclique plus ou moins grand. Dans notre cas, le moteur tourne plus ou moins vite selon que le rapport cyclique est grand ou petit. Pour savoir quel rapport cyclique correspond à quelle valeur, il faut faire une règle de trois :

Valeur argument	Rapport cyclique (%)
0	0
127	50
255	100

Tableau 10 : Variation de la PWM.

Le calcul donnant la valeur pour chaque portion est défini par cette relation :

$$argument = x * \frac{100}{255} \quad (3.1)$$

Le résultat de ce calcul donne la valeur de l'argument pour le rapport cyclique recherché. x est la valeur du rapport cyclique que l'on souhaite donner au signal.

Nous pouvons observer sur le schéma synoptique suivant, l'intérieur du double pont en H du circuit intégré L298P le pont A est à gauche, le pont B est à droite.

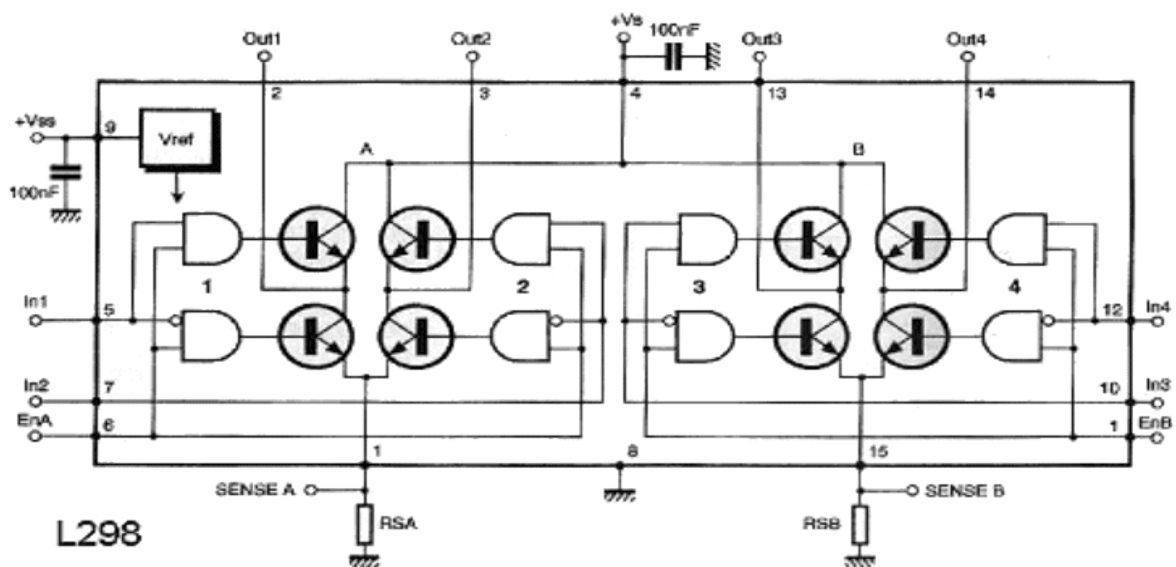


Figure 64 : schéma synoptique du circuit intégré L298P.

Les signaux de contrôle des ponts A et B :

Le pont A

	Input 1 (IN1)	Input 2 (IN2)	Fonction
Enable A (ENA) = L	-	-	Le pont A désactivé.
Enable A (ENA) = H	H	L	Le moteur branché sur le pont A (entre Output 1 (OUT 1) et 2 (OUT 2)) tourne dans le sens des aiguilles de la montre.
	L	H	Le moteur branché sur le pont A (entre Output 1 (OUT 1) et 2 (OUT 2)) tourne dans le sens inverse des aiguilles d'une montre.
	Input 1 = Input 2		Freinage rapide du moteur branché sur le pont A.

Tableau 11 : branchement du pont A.

Le pont B :

	Input 3 (IN3)	Input 4 (IN4)	Fonction
Enable B (ENB) = L	-	-	Le pont B désactivé.
Enable B (ENB) = H	H	L	Le moteur branché sur le pont B (entre Output 3 (OUT 3) et 4 (OUT 4)) tourne dans le sens des aiguilles d'une montre.
	L	H	Le moteur branché sur le pont B (entre Output 3 (OUT 3) et 4 (OUT 4)) tourne dans le sens inverse des aiguilles d'une montre.
	Input 3 = Input 4		Freinage rapide du moteur branché sur le pont B.

Tableau 12 : branchement du pont B.

Avec :

- L = LOW (niveau bas = 0 logique).
- H = HIGH (niveau haut = 1 logique).

Nous pouvons remarquer sur le schéma synoptique du circuit intégré L298P, qu'il n'y a pas de diodes de roue libre, c'est cela qu'il faut en rajouter pour protéger le circuit des courants de retour, de préférence il faut utiliser des diodes Schottky qui sont rapides, et peuvent fonctionner dans des régimes de hautes fréquences.

I.2.2.5 L'Arduino Motor Shield:

L'Arduino Motor Shield est un driver qui permet de commander aisément des moteurs à courant continu ou même des moteurs pas à pas dans certains cas, c'est pour cela que dans notre travail on va recourir au Motor Shield (voir la figure 65).

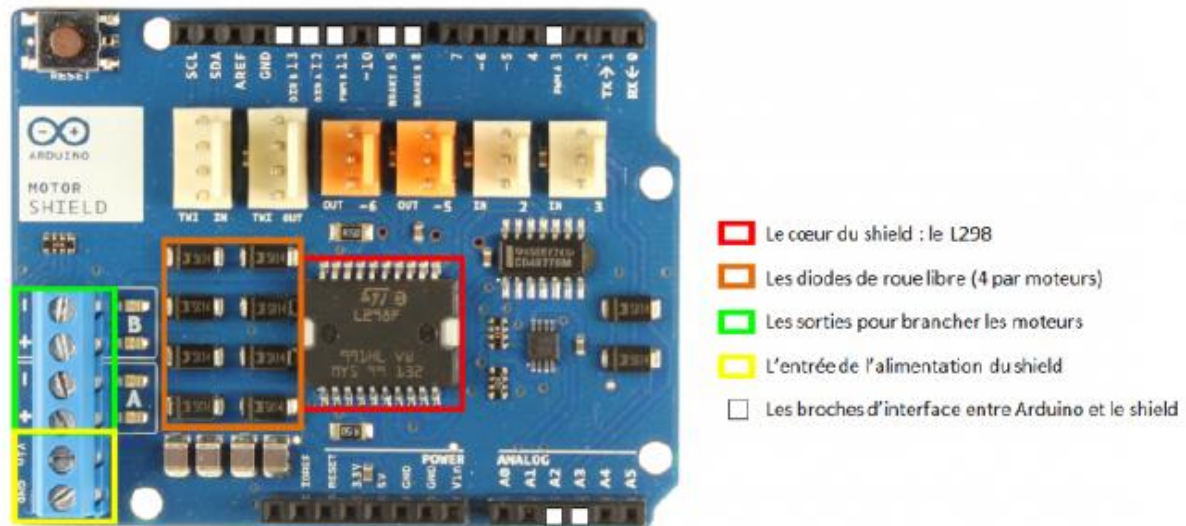


Figure 65 : Shield moteur Arduino.

Ce Shield est basé sur le circuit intégré L298P (qui est un double pont en H en paquet Power SO20 monté en surface). Il permet de contrôler la vitesse et la direction de chaque moteur individuellement, on peut aussi grâce à ce shield mesurer le courant absorbé par chaque moteur [24].

Les caractéristiques du shield moteur Arduino sont les suivantes :

- Tension d'alimentation : de 5V jusqu'à 12V.
- Détection de courant : 1,65V/A.
- Il peut contrôler deux moteurs à courant continu, ou un moteur pas à pas.
- Le courant maximum : 2A par canal ou 4A dans le cas d'une source d'alimentation externe.
- Il possède aussi deux modes de freinage :
 - La fonction d'arrêt immédiat (qui permet un arrêt immédiat du moteur).
 - La fonction arrêt de roulement libre (qui met un peu plus de temps).

Afin de protéger la carte Arduino sur laquelle le shield est monté, il est recommandé d'utiliser une source d'alimentation externe (entre 7V à 12V). Si on utilise des moteurs qui fonctionnent avec une tension d'alimentation de plus de 9V, il est recommandé de séparer les lignes d'alimentation du shield de celles de la carte Arduino, et cela en coupant le jumper « Vin Connect » sur la partie arrière du shield.

Les entrées/sorties du shield sont reliées à l'Arduino de la manière suivante :

Fonction	Broches moteur A	Broches moteur B
Direction	12	13
PWM	3	11
Frein	9	8
Mesure de courant	A0	A1

Tableau 13 : liaison de l'Arduino Uno au Shield moteur.

La mesure de courant se fait sur les broches A0 et A1. Mais dans le cas où on a besoin de ces broches pour une autre application, il est possible là encore de les désactiver en coupant le strap en dessous de la carte. Et c'est aussi la même chose pour la fonction de freinage, qui, elle aussi, peut être désactivée en coupant ses jumper [25].



Figure 66 : Shield moteur monté sur une carte Arduino Uno.

L'alimentation utilisée est un pack de 4 batteries Alcaline de taille AA dont la capacité est de 6V ainsi on respecte la consigne qui nous oblige à utiliser une source d'alimentation de 5V à 12V.

Le branchement des deux motoréducteurs au Shield moteur est représenté sur la figure suivante :

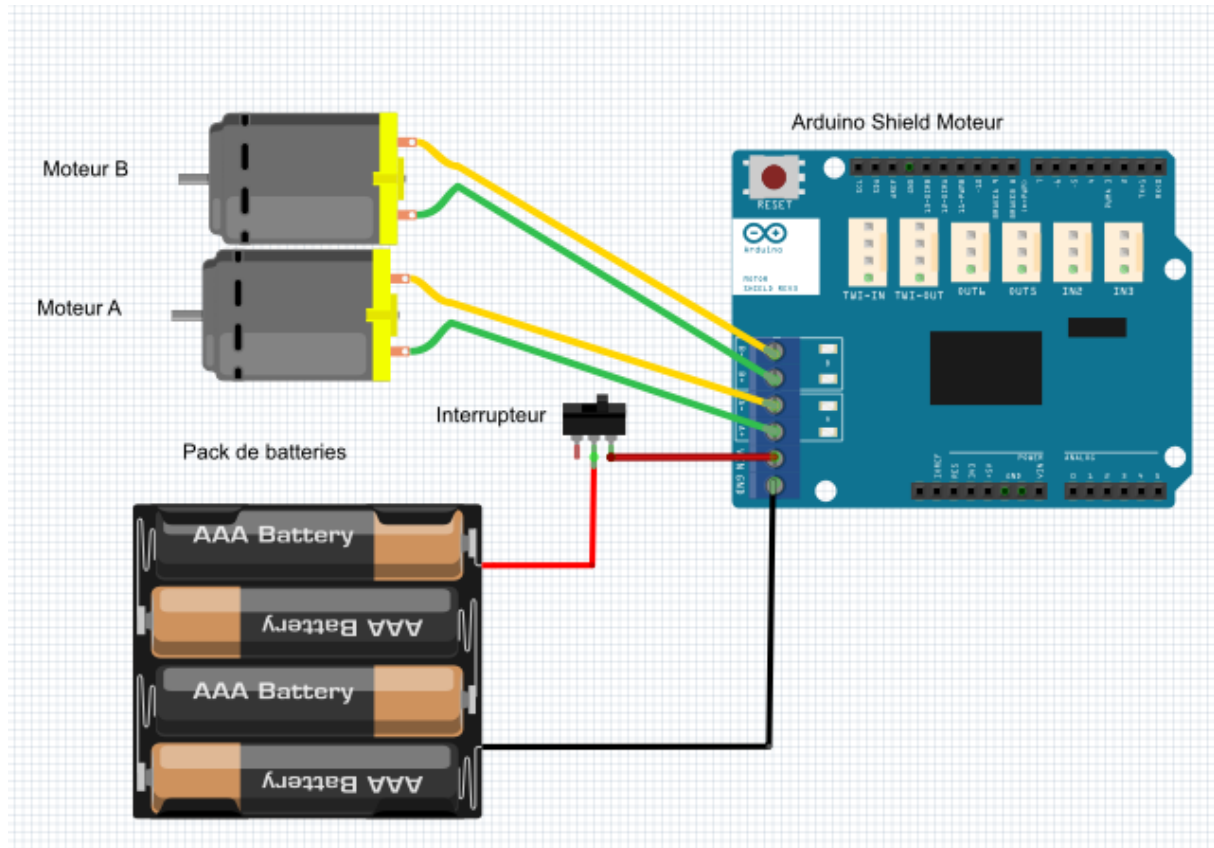


Figure 67 : Branchement des motoréducteurs vers le shield moteur.

I.2.3 La communication via Wifi :

Parfois l'intervention humaine est requise pour corriger la trajectoire du robot qui peut être erroné à cause des différentes défaillances que le robot peut rencontrer (dysfonctionnement d'un composant du robot, faille de logique, environnement non contrôlé, données erronées issues du capteur, etc.). Le contrôle via wifi a été choisi dans notre travail afin d'accomplir cette tâche bien précise mais d'autres moyens de communications (sans fils) peuvent être utilisés comme alternative (Bluetooth, XBee, ZigBee, etc.). L'avantage de la technologie Wifi est qu'elle offre un moyen efficace et simple pour contrôler le robot via différents appareils (ordinateur, téléphone, tablette), ce qui réduit considérablement le coût car nous n'avons pas besoin de réaliser l'émetteur.

La commande à distance du robot se fait en utilisant le module ESP8266-07 ce module comporte 16 pattes, mais nous ne ferons appel qu'à 5 d'entre elles qui seront les suivantes :

GND : Branché à la masse de l'Arduino.

VCC: Branché sur l'alimentation 3.3V.

TXD: Branché sur la broche RX de l'Arduino.

RXD: Branché sur la broche TX de l'Arduino.

CH-PD: Chip Select Power Down.

A noté :

- Lors du branchement de CH-PD du module wifi à l'Arduino il faut rajouter une résistance de 10 kOhms de manière à réduire le courant entrant dans la broche.
- L'alimentation du module se fait avec 3.3V même si l'ESP8266 possède un dispositif de sécurité jusqu'à 5.8V.

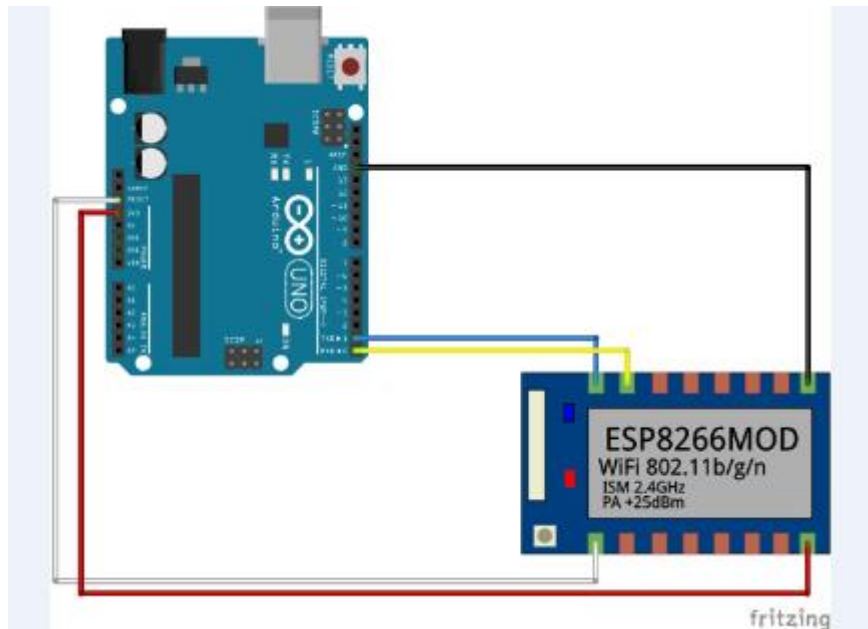


Figure 68 : Branchement de l'ESP8266 à l'Arduino.

I.2.3.1 Fonctionnement du module :

Le module ESP8266 peut fonctionner selon 3 modes :

- Mode Client : dans ce mode, il utilise le réseau wifi pour se connecter à un ordinateur ou un autre point d'accès.
- Mode Server : dans ce mode, il fonctionne comme un serveur, ou maître de plusieurs autres entités.
- Mode Client et Server : ce mode regroupe les deux cas précédents.

La sélection de ces modes de fonctionnement, ainsi que la configuration de ce module se fait grâce à des commandes spécifiques, qui sont les commandes AT [29].

I.3 La réalisation logiciel :

I.3.1 La programmation de l'Arduino :

Pour programmer l'Arduino, on doit la brancher à un ordinateur via un câble USB (fiche A vers B) (voir la figure 69), puis sur l'ordinateur on a besoin d'installer l'environnement de développement Arduino IDE (Integrate Development Environment) [22] qui va nous permettre de programmer la carte Arduino.



Figure 69 : Câble USB (fiche A vers B).

Les programmes écrits avec l'IDE Arduino sont appelés des « Croquis » ou (Sketch en anglais) leur extension est « .ino ».

L'IDE d'Arduino en fait, il s'agit d'un compilateur. Alors qu'est-ce que c'est exactement ?

I.3.2 Un compilateur :

En informatique, ce terme désigne un logiciel qui est capable de traduire un langage informatique, ou plutôt un programme utilisant un langage informatique, vers un langage plus approprié afin que la machine qui va le lire puisse le comprendre. Le compilateur va donc traduire les instructions du programme, écrites en langage texte, vers un langage dit "machine". Ce langage utilise uniquement des 0 et des 1. Cela peut se traduire de la façon suivante :



Figure 70 : Traduction d'un programme informatique en langage machine.

I.3.3 Présentation du logiciel :

La figure 71, représente l'interface de l'IDE d'Arduino.

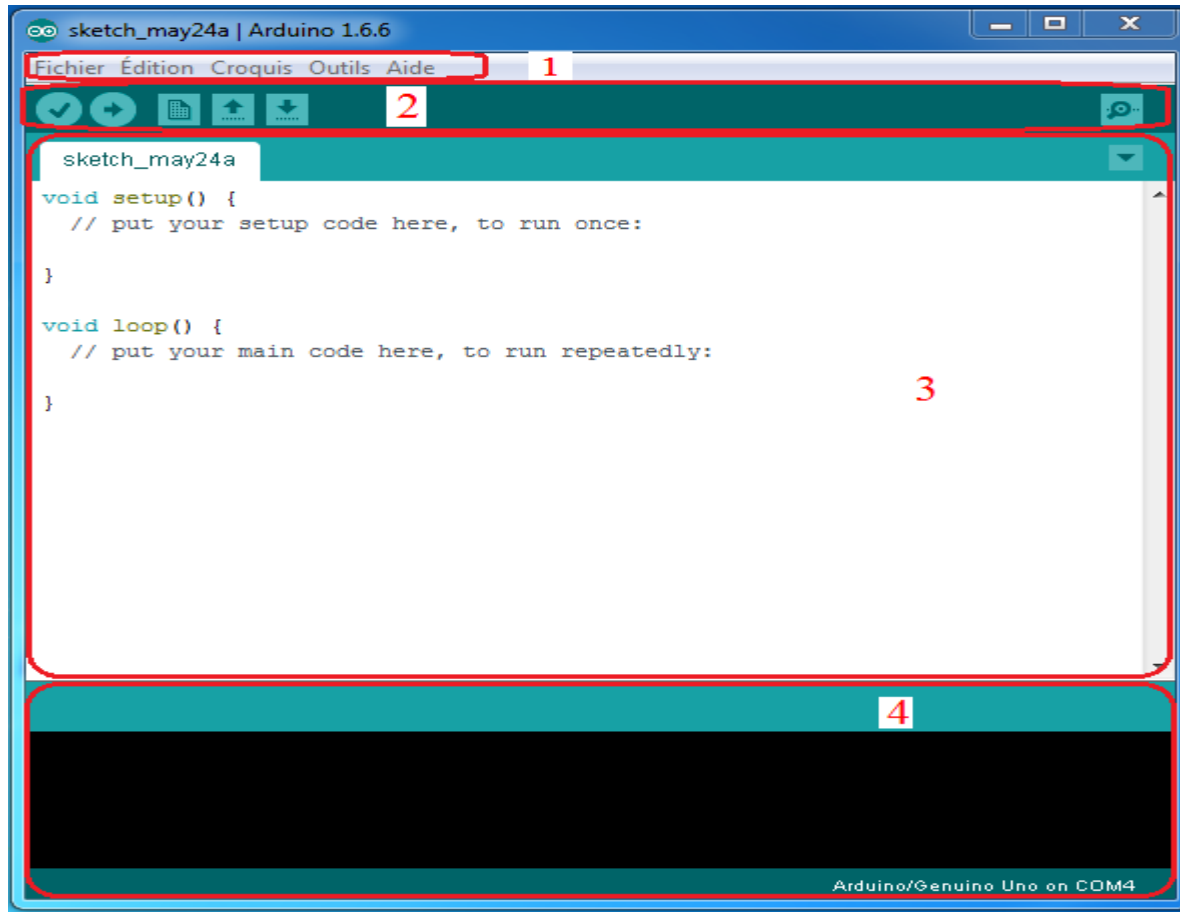


Figure 71 : Interface de l'environnement de développement Arduino IDE.

I.3.3.1 Correspondance :

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

Le cadre numéro 2 : il contient les 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 dans notre programme. C'est le débogueur.

I.3.3.2 Fonctions des icônes de la barre de boutons:

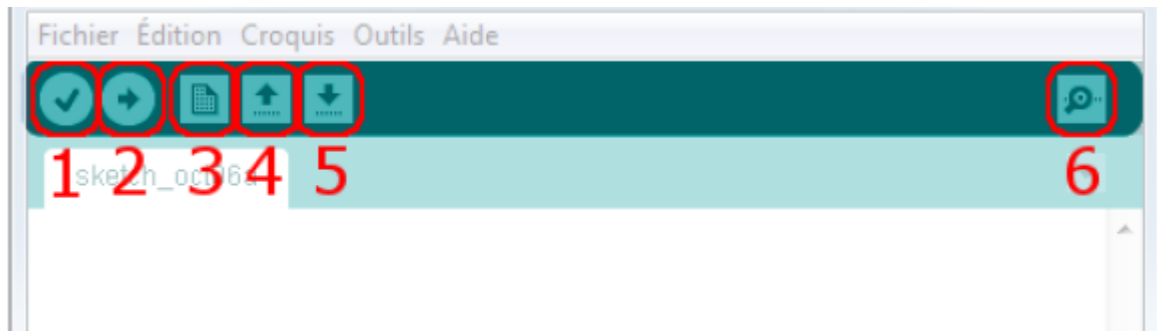


Figure 72 : les icônes de l'IDE.

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

Bouton 2 : Charge (Téléverser) le programme dans la carte Arduino.

Bouton 3 : Crée un nouveau croquis.

Bouton 4 : Ouvre un croquis déjà existant.

Bouton 5 : Enregistre le fichier courant.

Bouton 6 : Ouvre le moniteur série, qui permet la communication entre la carte Arduino et l'ordinateur à lequel on l'a branché.

Pour programmer n'importe quelle carte Arduino il faut suivre les étapes suivantes :

- Aller dans le menu **Outils**, ensuite cliquer sur **Type de carte** et puis sélectionner la carte dont on a besoin pour notre cas « Arduino/Genuino Uno » qui est en fait le même modèle mais juste fabriqué par un autre fabricant.
- Ensuite il faut s'assurer qu'on est connecté au bon port série (surtout si on utilise plusieurs cartes en même temps). Il nous faut donc aller dans le menu **outils** puis **Port** et enfin le port adéquate.
- Par la suite il ne reste plus qu'à programmer l'application que l'on souhaite réaliser, une fois le programme terminé, on clique sur le bouton « **Téléverser** » qui va se charger de transmettre le code source si celui-ci n'a pas d'erreurs bien sûr, voilà la carte Arduino a maintenant été programmée.

I.3.4 Programmation sous l'IDE d'Arduino :

Comme nous l'avons déjà dit dans la partie conception, le langage de programmation de l'Arduino étant dérivé du C et du C++, il respecte les règles de syntaxe propres à ces langages qui restent au demeurant, relativement abordables.

Le contenu du programme, donc le programme en lui-même, est ce qui va définir chaque action que va exécuter la carte Arduino. Mais ce n'est pas tout ! Dans le programme il y a plusieurs zones, que nous verrons plus en détail.

La première zone sert principalement à dire à la carte de garder en mémoire quelques informations qui peuvent être : l'emplacement d'un élément connecté à la carte, par exemple une LED en broche 13, ou bien une valeur quelconque qui sera utilisée par la suite dans le programme, ainsi que des fonctions et des bibliothèques très utilisées en C.

```
// Les deux slashes sont utiliser pour faire un commentaire

#define LED_PIN 13 // Led  attaché à la broche 13 de la carte
#include <Servo.h> // faire appelle a la bibliothèque servo

const int LEDPin = 13 ; //Led  attaché à la broche 13 de la carte
const int t = 15 ; // une constante entière t égale à 30
```

La seconde zone est l'endroit où l'on va initialiser certains paramètres du programme. Par exemple, on pourra dire à la carte qu'elle devra communiquer avec l'ordinateur ou simplement lui dire ce qu'elle devra faire de la LED qui est connectée sur sa broche 13.

```
void setup ()
{
  /* Dans cette zone on initialise généralement tous les éléments
  Que l'on a déclaré dans la première zone */
  /* est une autre façon de faire des commentaires sur une zone plus étendu */
  pinMode(LEDpin, OUTPUT) ; // mettre la Led en sortie
  } // Les crochets sont indispensables
```

La dernière zone est la zone principale où se déroulera le programme. Tout ce qui va être écrit dans cette zone sera exécuté par la carte, ce sont les actions que la carte fera. Par exemple, c'est ici qu'on pourra lui dire de faire clignoter la LED sur sa broche 13.

```
// c'est la fontion principale
void loop()
{
  digitalWrite(13, HIGH); // allumer la Led
  delay(1000);           // temporisé durant une second
  digitalWrite(13, LOW); // eteindre la Led
  delay(1000);           // temporisé durant une second
}
// c'est une boucle infinie
```

Au final, tout ce que va faire la carte est inscrit dans le programme. Sans programme, la carte ne sert à rien ! C'est grâce au programme que la carte Arduino va savoir qu'une LED est connectée sur sa broche 13 et ce qu'elle va devoir faire avec, allumer et éteindre la LED alternativement pour la faire clignoter.

I.4 L'Algorithme de fonctionnement du robot :

Notre robot répond à un certain fonctionnement qui est décrit par l'Algorithme suivant :

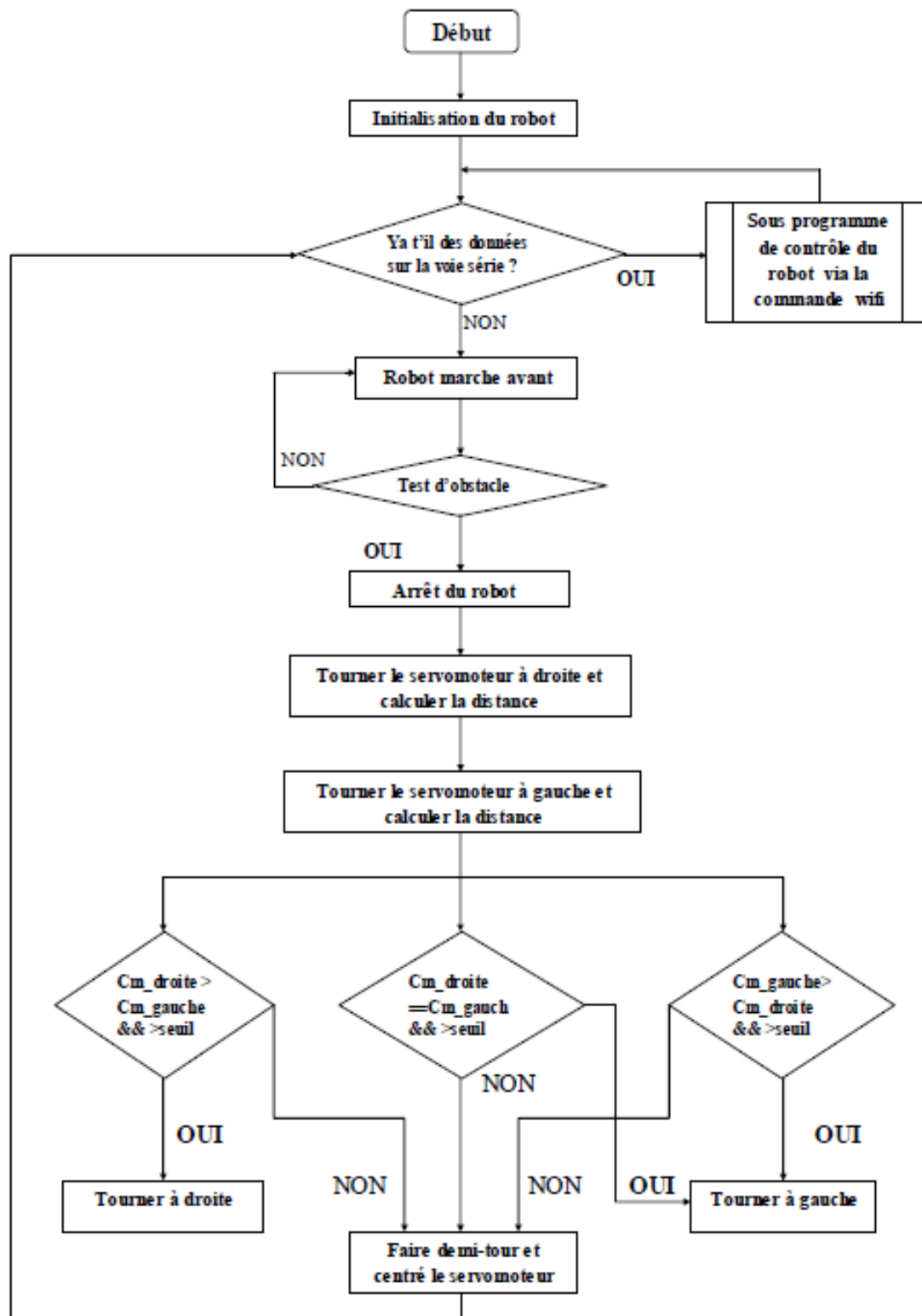


Figure 73 : Algorithme de fonctionnement du robot

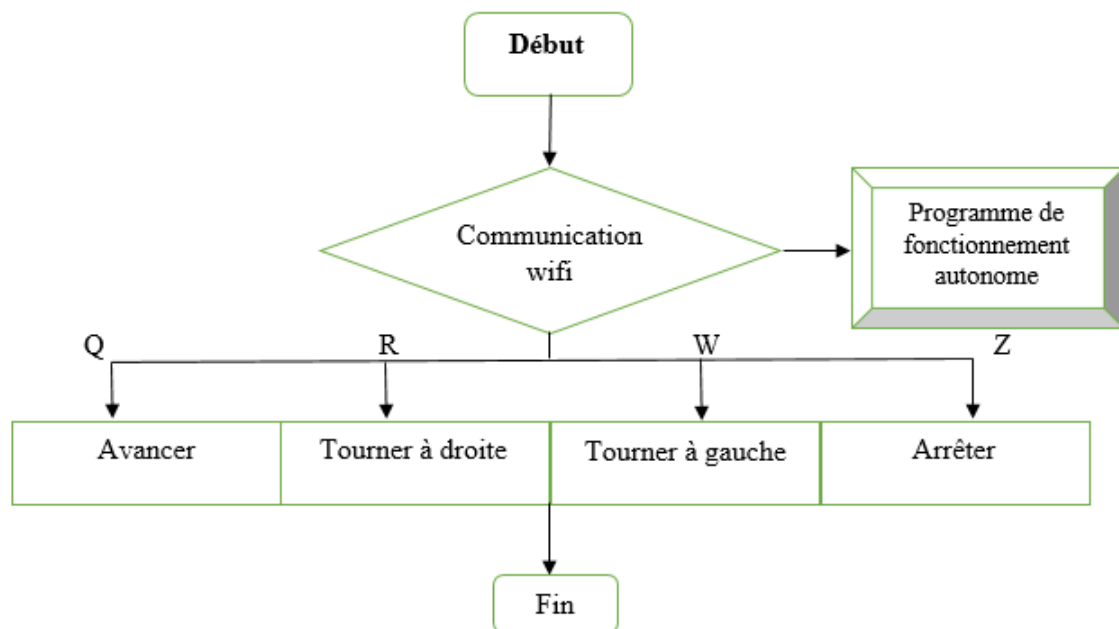
I.4.1 Algorithme du sous-programme de commande par wifi :

Figure 74 : Algorithme de téléguidage via Wifi

I.5 Conclusion:

Dans ce chapitre nous avons procédé à la réalisation pratique d'un robot mobile autonome avec la possibilité d'être téléguidé par Wifi.

Cette partie est la mise en relief des différents composants garantissant la réalisation de ce robot mobile de type voiture. Ce dernier se base sur une carte Arduino Uno, ainsi qu'un moyen de locomotion consistant en deux motoréducteurs (25 GA 370) commandés par un driver (le Shield moteur Arduino) qui se base sur un double pont en H le L298P. La perception quant à elle est assurée par un capteur ultrasonique (HC-SR04) qui permet de mesurer la distance avec les obstacles, celui-ci est fixé sur un servomoteur (SG90) dont le rôle consiste à faire un balayage du champ de perception sans que le robot ne soit obligé de tourner sur lui-même.

Notre robot répond à une certaine logique dans sa façon de fonctionner (Algorithme figure 73). Si une défaillance est enregistrée ou un contrôle manuel est désiré, alors nous avons conçu à cet effet une commande via Wifi basée sur le module ESP8266-07 permettant la transmission de données par le protocole TCP.

Une fois notre robot réalisé, nous avons procédé à une panoplie de tests pour nous assurer de son bon fonctionnement.

CONCLUSION GÉNÉRALE :

Conclusion Générale :

Notre travail traite du problème de la navigation autonome des robots mobiles dans le cadre d'un environnement inconnu. Ce cadre est défini par les spécificités des systèmes considérés et des applications envisagées, que nous pouvons résumer par la navigation pour robots mobiles non holonomes à proximité des obstacles.

Le fil conducteur de notre projet était l'idée de se lancer dans une discipline révolutionnaire et émergente pouvant apporter des solutions multiples où le champ d'intervention de l'être humain est plus ou moins réduit. En premier nous avons dérasé un état de l'art sur la robotique mobile, juste après, une étude théorique des différents éléments constituant notre robot est établi pour ensuite, jumeler cette panoplie de composants en une réalisation pratique fiable. En dernier, nous avons proposé un algorithme répondant à la problématique de la navigation d'une machine intelligente dans un espace non connu.

Dans ce mémoire, on a pu réaliser avec succès un robot mobile autonome basé sur une carte Arduino Uno dotée d'un capteur ultrasonore comme moyen de perception avec un servomoteur qui permet d'orienter le capteur dans une direction ou une autre sans avoir à tourner tout le robot afin de prendre les mesures de distance dans les différentes directions, et de deux motoréducteurs commandés avec un Shield moteur basé sur un double pont en H (L298P) comme moyen de locomotion. Ainsi que le module ESP8266 qui permet d'avoir une liaison Wifi afin de pouvoir télécommander à distance le robot.

Les possibilités d'amélioration du robot présentées dans ce mémoire sont nombreuses, car une fois le robot est opérationnel, on peut modifier son algorithme de fonctionnement pour qu'il puisse par exemple devenir un robot solveur de labyrinthe, ou encore l'employer pour effectuer des tâches de reconnaissance dans des zones dangereuses ou inaccessibles pour l'être humain. On peut aussi l'améliorer du côté Hardware, car on peut par exemple utiliser un autre moyen de perception tel une caméra effectuant une série de traitement d'image afin que la machine puisse se repérer d'une manière efficace dans son environnement.

Après ce travail, certaines questions devraient obtenir une réponse « comment seront les robots du futur ? Peuvent-ils être plus intelligents que nous ? ». Le célèbre physicien Stephen Hawking a dit à la conférence de Zeitgeist à Londres que: « les robots alimentés par l'intelligence artificielle, pourraient dépasser les humains dans les 100 prochaines années [30]. Lorsque cela se produira, nous devons nous assurer que ces machines dotées d'une intelligence artificielle auront des objectifs alignés avec les nôtres ». En tout état de cause, les ingénieurs et les chercheurs devront toujours garder un impact positif sur leur invention et ne jamais créer une machine capable de gouverner un homme [31].

ANNEXES

ANNEXE A

Le programme de notre robot:

```

#include <SoftwareSerial.h>
SoftwareSerial ESP8266(2,4);
#include <Servo.h>

// définition des broches utilisées par le capteur HC-SR04
int trig = 7;
int echo = 6;

// définition des broches utilisées sur le shield contrôle moteur DRI0001
int direction_moteur_droite = 12; // DIR A
int controle_moteur_droite = 3; // PWM A
int frein_moteur_droite = 9; // BREAK A

int direction_moteur_gauche = 13; // DIR B
int controle_moteur_gauche = 11; // PWM B
int frein_moteur_gauche = 8; // BREAK B

long lecture_echo;
long cm, cm_gauche, cm_droite;
Servo monservo; // crée l'objet pour contrôler le servomoteur

void moteur (int moteur_gauche, boolean sens_moteur_gauche, boolean
frein_gauche, int moteur_droite, boolean sens_moteur_droite, boolean
frein_droite)
{
    // définition sens moteur LOW avance, HIGH recule

    digitalWrite(direction_moteur_gauche, sens_moteur_gauche);
    analogWrite(controle_moteur_gauche, moteur_gauche);
    digitalWrite(frein_moteur_gauche, frein_gauche);

    digitalWrite(direction_moteur_droite, sens_moteur_droite);
    analogWrite(controle_moteur_droite, moteur_droite);
    digitalWrite(frein_moteur_droite, frein_droite);
}

void virage_90 (boolean sens)
{
    // et on va jouer sur la durée en ms pour effectuer le virage souhaité
    // sens à HIGH pour gauche, LOW pour droite

    digitalWrite(direction_moteur_gauche, sens);
    digitalWrite(direction_moteur_droite, !sens);
    analogWrite(controle_moteur_gauche, 240);
    analogWrite(controle_moteur_droite, 250);
    delay(2000);
    analogWrite(controle_moteur_gauche, 0);
    analogWrite(controle_moteur_droite, 0);
}

void calcul_distance()
{
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    lecture_echo = pulseIn(echo, HIGH);
    cm = lecture_echo / 58;
}

```

```

}

void suivre(char commande)
{
    switch(commande) {
        case 'Q':

            // Robot avance tout droit
            moteur(240, LOW, LOW, 250, LOW, LOW);

            break;

        case 'Z':
            // Robot stop
            moteur(0, LOW, HIGH, 0, LOW, HIGH);

            break;
        case 'W':

            // Robot tourne 90° à gauche
            virage_90(HIGH);
            delay(3000);
            // Robot stop
            moteur(0, LOW, HIGH, 0, LOW, HIGH);
            break;

        case 'R':
            // Robot tourne 90° à droite
            virage_90(LOW);
            delay(3000);
            // Robot stop
            moteur(0, LOW, HIGH, 0, LOW, HIGH);

            break;
        case 'F':

            // Robot recule
            moteur(240, HIGH, LOW, 250, HIGH, LOW);

            break;

    }
}

void setup()
{
    Serial.begin(115200);
    ESP8266.begin(115200);
    initESP8266();

    pinMode(3, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);

    analogWrite(controle_moteur_gauche, 0); // arrêt moteur gauche
    analogWrite(controle_moteur_droite, 0); // arrêt moteur droite
    pinMode(trig, OUTPUT);

```

```

digitalWrite(trig, LOW);
pinMode(echo, INPUT);
monservo.attach(5); // utilise la broche 9 pour le contrôle du
servomoteur
monservo.write(89); // positionne le servomoteur en fait à 90°
}

void loop()
{
  while(ESP8266.available())
  {
    Serial.println(ESP8266.readString());
    delay(1000);
    Serial.find(":");
    while(Serial.available()){
      char c=Serial.read();
      if(c=='Q' || c=='>' || c=='<' || c=='W' || c=='Z' || c=='R' || c=='F'
|| c=='s'){
        suivre(c);
      }
    }
  }

  monservo.write(89);
  delay(1000);
  calcul_distance();
  //Serial.println(cm);
  if (cm >= 40)
  {
    // Robot avance tout droit
    moteur(240, LOW, LOW, 250, LOW, LOW);
  }
  else
  {
    // Robot stop
    moteur(0, LOW, HIGH, 0, LOW, HIGH);
    monservo.write(10);
    delay(1000);
    calcul_distance();
    cm_droite = cm;
    monservo.write(190);
    delay(1000);
    calcul_distance();
    cm_gauche = cm;

    if (cm_droite < cm_gauche)
    {
      // Robot tourne 90° à gauche
      virage_90(HIGH);
    }
    if (cm_droite > cm_gauche)
    {
      // Robot tourne 90° à droite
      virage_90(LOW);
    }
    if (cm_droite == cm_gauche)
    {
      // Robot demi-tour
      virage_90(HIGH);
      virage_90(HIGH);
    }
  }
}

```

```

    }
}
void initESP8266()
{
    envoieAuESP8266("AT+RST");
    recoitDuESP8266(2000);
    envoieAuESP8266("AT+CWSAP=\"esp\", \"strongPass2015\";4,3\r\n");
    recoitDuESP8266(3000);
    envoieAuESP8266("AT+CWMODE=3");
    recoitDuESP8266(5000);
    envoieAuESP8266("AT+CIFSR");
    recoitDuESP8266(1000);
    envoieAuESP8266("AT+CIPMUX=1");
    recoitDuESP8266(1000);
    envoieAuESP8266("AT+CIPSERVER=1,80");
    recoitDuESP8266(1000);

}
void envoieAuESP8266(String commande)
{
    ESP8266.println(commande);
}

void recoitDuESP8266(const int timeout)
{
    String reponse = "";
    long int time = millis();
    while( (time+timeout) > millis())
    {
        while(ESP8266.available())
        {
            char c = ESP8266.read();
            reponse+=c;
        }
    }
    Serial.print(reponse);
}

```

ANNEXE B

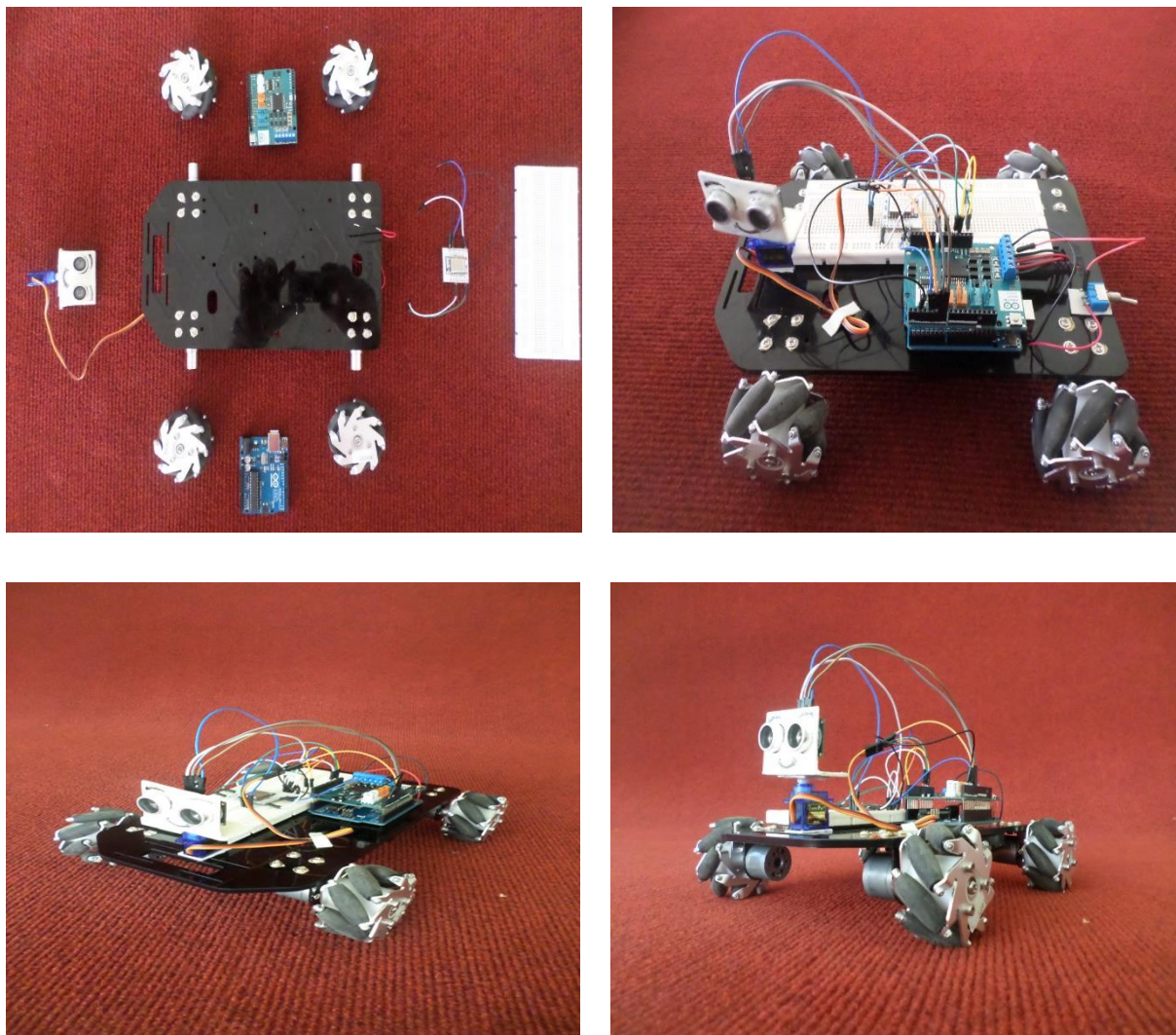


Figure 75 : Photos de notre robot.



108

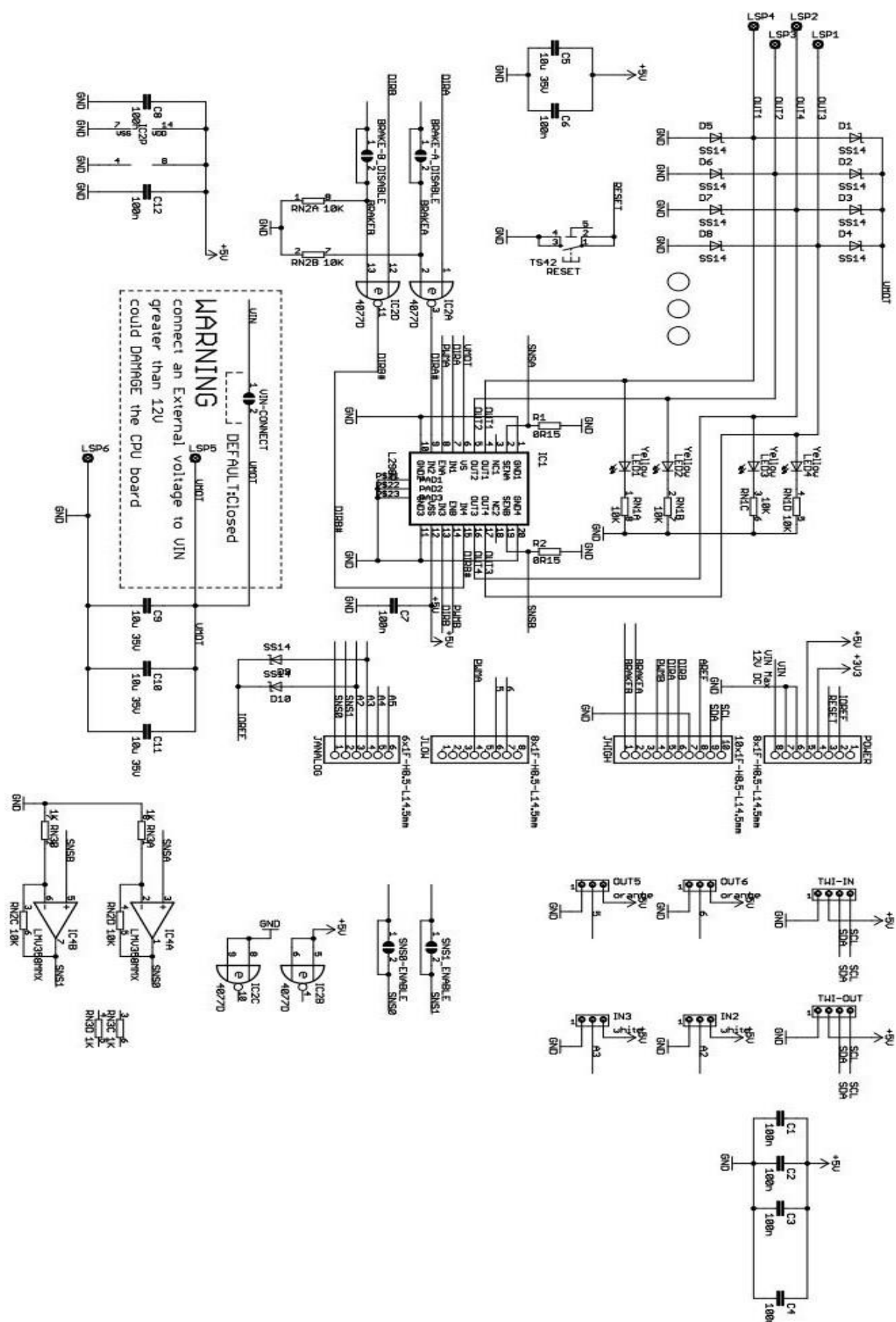


Figure 77: Schéma de l'Arduino Motor Shield.

BIBLIOGRAPHIE

- [1] : W. Richard, L. Daniel et M. Robert, « Systems Engineering the Curiosity Rover : A Retrospective, » chez System of Engineering (SoSE), 2013 8th international Conference ; 2-6 juin 2013.
- [2] :R. Marcel, Qualification d'un télémètre à balayage laser pour la robotique, Grenoble : Mémoire d'ingénieur, Conservatoire National des Arts et Métiers : 2000.
- [3] : H. Joachim et R. Joachim, localization of a mobile robot by matching 3D laser-range-image and preicated sensor imges, October 24-26 éd, Proceedings of the vehicules, 1994 ;pp 345-350.
- [4] : Methode de trackage des robot mobiles, In proceedings of the 1998 IEEE international, 1998.
- [5] : E. Colle, F. Chavand, Dispositif de localisation par télémétrie, vol 5, RAPA.
- [6] : G. Gacia, P. Bonnifait ; 6 DOF dynamic localization of mobile robot, Preprints of the fifth IFAC Symposium on Robot control, 2000, pp. 173-179.
- [7] : J. José, B. Gonzales, Vision Panoramique pour la robotique mobile : stéréovision et localisation par indexation d'images, Toulouse : Thèse de Doctorat de l'Institut National Polytechnique de Toulouse, 2004.
- [8] : G. Pierrick et R. Arnaud, Modélisation des environnements 3D intérieurs et extérieurs, IEEE Conférence Internationale des systèmes automatique et de la robotique.
- [9] : J. Il-Kyun, Cartographie et localisation simultanée en environnements 3D à patrir de la stéréovision, Toulouse : Thèse de Doctorat de l'institut National Polytechnique de Toulouse, 2004.
- [10] : B. M. Claus et S.A. Claus, Optimal Landmak Selection for triangulation of robot position, vol. 23, Journal de la Robotique et des Systèmes Autonomes, 1999, pp 275-295.
- [11] : Juan Gabriel AVINA Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structuré these de Doctorat de l'Institut National Polytechnique de Toulouse 2005.
- [12] : M. Sylvain, Système embarqué de localisation et de perception pour robot mobile, Ecole Polytechnique de Montréal, Cours Séminaire ELE6904, 25-27 janvier 2004, pp. 350-354. 399-408.
- [13] : B. Marhic, L. Delahoche, C.Pégard, Navigation pour robot mobile à base omnidirectionnelle et vision, pp 210-230.
- [14] : D. Cyril, Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels, CREA, Ed., Thèse de Doctorat de l'Université de Paris VI, Février 2002.
- [15] : Keith J. O'Hara, Towards Robot Systems Architecture, Multirobot Systems and Physical Data Structures. Papers from the AAAI 2011.
- [16] : E.Beaudry, planification de tâches pour un robot mobile autonome, thèse de Maître de sciences, université de Sherbrooke, août 2006.

- [17] : R. A. Brooks, A robust layered control system for a mobile robot, 2 éd ; IEEE Journal of Robotics and Automation.
- [18]: B. Fry et C. Reas, « Processing.org » [en ligne] : <https://www.processing.org/>; le 18/05/2016
- [19] :arduino.cc « Arduino – SoftwareSerial »
<https://www.arduino.cc/en/Reference/SoftwareSerial> [en ligne]; le 10/06/2016
- [20] : arduino.cc « Arduino –Wire »
<https://www.arduino.cc/en/Reference/Wire> [en ligne]; le 18/06/2016
- [21] : arduino.cc « Arduino –SPI » :
<https://www.arduino.cc/en/Reference/SPI> [en ligne]; le 18/06/2016
- [22] : arduino.cc : <https://forum.arduino.cc/index.php?topic=86882.0>; [en ligne] le 15/05/2016
- [23] : arduino.cc : <https://www.arduino.cc/en/Tutorial/LibraryExamples> ; [en ligne] le 15/05/2016
- [24] : Arduino : maitrisez sa programmation et ses cartes d’interfaces (shields) de Christian Tavernier
- [25] : Arduino : premiers pas en informatique embarquée de Simon Landrault & Hippolyte Weisslinger
- [26] :<http://www.memoireonline.com/01/16/9368/Conception-et-realisation-d-un-robot-mobile--base-d-arduino.html> ; le 20/06/2016.
- [27] : Les capteurs pour Arduino et Raspberry Pi- Tutoriels et projets de Kimmo Karvinen et ville Valtokari pp : 37- 39 , 61-63.
- [28] :https://cdn-shop.adafruit.com/product-files/2471/0ESP8266__Datasheet__EN_v4.3 [en ligne]
- [29] : <https://zoetrope.io/tech-blog/esp8266-bootloader-modes-and-gpio-state-startup> [en ligne]; le 17/06/2016
- [30] : Follow the Zeitgeist Movement - London at : http://www.meetup.com/fr-FR/Zeitgeist-Movement-London/?chapter_analytics_code=UA-12837668-1.
- [31]: Stephen Hawking predicts robot apocalypse coming within 100 years : <http://www.geek.com/news/stephen-hawking-predicts-robot-apocalypse-in-next-century>.