

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



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Spécialité : Télécommunication et réseaux
Filière : Génie Electrique

Présenté par
MAHIOUT SAMY

Mémoire dirigé par **Mr. BENNAMANE Kamal**

Thème

Conception et réalisation d'un robot autonome mobile

Mémoire soutenu publiquement le 01 octobre 2015 devant le jury composé de :

Mr. Hamid HAMICHE

M.C.A, UMMTO, Président

Mr. Kamal BENNAMANE

M.C.A, UMMTO, Rapporteur

Mr. Rachid ZERMI

M.C.B, UMMTO, Examineur

Mr. Mohamed THAHANOUT

M.A.A, UMMTO, Examineur

REMERCIEMENTS

Je tiens à remercier tout d'abord notre Dieu, le tout puissant, de m'avoir donné la santé et la volonté pour compléter ce modeste travail.

Je tiens à remercier vivement mon promoteur monsieur BENNAMANE KAMAL d'avoir accepté de m'encadrer, et pour tous ses efforts et conseils qu'il a apportés durant la réalisation de ce mémoire, ainsi que d'avoir mis à ma disposition le laboratoire maquette afin que je puisse réaliser mes expérimentations aisément.

J'exprime également mes vifs remerciements pour tous les enseignants qui m'ont encouragé et conseillé durant tous mon cursus universitaire.

Je remercie vivement tous les membres du Jury d'avoir accepté d'examiner et d'évaluer ce modeste travail.

Je remercie aussi tous mes amis, et toutes les personnes qui de près ou de loin ont contribué à l'élaboration de ce mémoire.

Enfin, je tiens à remercier tous les membres de ma famille profondément, pour leurs encouragements et leurs soutiens sans lesquels je n'aurais pas pu terminer ce travail.

GLOSSAIRE :

AC	Alternative Current
AFH	Adaptive Frequency Hopping Feature
CCD	Charge Coupled Device
CPU	Central Processing Unit
CSR	Cambridge Silicon Radio
DC	Direct Current
EDR	Enhanced Data Rate
EEPROM	Electrically-erasable programmable read-only memory
HS	High Speed
I2C	Inter Integrated Circuit
ICSP	In Circuit Serial Programming
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IR	Infra-Red
JTAG	Joint Test Action Group
LED	Light emitting diode
MISO	Master Input, Slave Output
MOSI	Master Output, Slave Input
PIO	Peripheral Input Output
PSD	Position Sensitive Device
PWM	Pulse Width Modulation
RTC	Real Time Clock
SCL	Serial Clock
SCLK	Serial Clock
SDA	Serial Data
SPI	Serial Peripheral Interface

SPP	Serial Port Protocol
SRAM	Static Random Access Memory
SS	Slave Select
TTL	Transistor-Transistor logic
TWI	Two Wire Interface
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WPAN	Wireless Personal Area Network

Table des matières :

INTRODUCTION GENERALE :	1
CHAPITRE I. Généralités sur les robots mobiles :	3
I.1. Introduction :	3
I.2. Présentation du robot mobile :	4
I.2.1. Classification des robots mobiles :	4
a. Classification selon le degré d'autonomie :	4
a.1. Robot télécommandé :	5
a.2. Robot semi-autonome :	5
a.3. Robot autonome :	5
b. Classification selon le domaine d'application :	5
b.1. Les robots industriels et de service :	5
b.2. Les robots militaires :	5
b.3. Les robots de laboratoire :	6
c. Classification selon le type de locomotion :	6
c.1. Les robots mobiles à roues :	6
c.1.1. Robots unicycle :	6
c.1.2. Robots tricycle :	7
c.1.3. Robots voiture :	7
c.1.4. Robots mobiles omnidirectionnels :	9
d. Les robots mobiles à chenilles :	10
e. Les robots mobiles marcheurs :	11
f. Les robots mobiles rampants :	12
I.2.2. Notion de Perception :	13
I.2.3. Les capteurs en robotique mobile :	14
a. Capteurs internes :	16
a.1. Les capteurs de déplacement :	17
a.1.1. Les odomètres :	17
a.1.2. Les accéléromètres :	18
a.2. Les capteurs d'attitude :	18

a.2.1.	Les gyroscopes :	18
a.2.2.	Les gyromètres :	19
a.2.3.	Les gyrocompas :	19
a.2.4.	Le magnétomètre :	19
b.	Capteurs externes :	20
b.1.	Les capteurs télémétriques :	20
b.1.1.	Les capteurs laser :	20
b.1.2.	Les capteurs infrarouges :	22
b.1.3.	Les capteurs ultrasonores :	23
b.2.	Les systèmes de vision :	26
b.2.1.	La stéréovision :	26
b.2.2.	Les capteurs à triangulation active (lumière structurée) :	27
b.2.3.	La vision omnidirectionnelle :	28
I.2.4.	Actionneurs :	29
I.2.5.	Architecture décisionnelles :	29
a.	Architecture délibérative :	29
b.	Architecture comportementale :	30
c.	Architecture hybride :	31
I.2.6.	La navigation :	31
I.3.	Conclusion :	32
CHAPITRE II.	Conception d'un robot autonome mobile :	33
II.1.	Introduction :	33
II.2.	Structure du robot à concevoir :	33
II.3.	Le capteur ultrasonore Parallax 28015 :	34
II.3.1.	Caractéristiques et spécifications :	34
II.3.2.	Principe de fonctionnement :	35
II.3.3.	Considérations pratique :	36
II.4.	Les motoréducteurs :	38
II.4.1.	Les moteurs à courant continu :	38
a.	Les moteurs à courant continu avec balais :	39
b.	Les moteurs à courant continu sans balais :	40

II.4.2.	Les Réducteurs :	41
II.4.3.	Commande des moteurs à courant continu :	44
II.5.	Les servomoteurs :	46
II.5.1.	Définition et principe de fonctionnement :	46
II.5.2.	Commande des servomoteurs :	49
II.6.	Téloguidage par Bluetooth :	50
II.6.1.	La technologie Bluetooth :	50
II.6.2.	Le module HC-05 :	52
II.7.	La carte Arduino UNO :	53
II.7.1.	Caractéristiques :	55
II.7.2.	Alimentation de la carte Arduino Uno :	56
II.7.3.	Les Entrées/Sorties :	57
a.	Numérique (digital) :	57
b.	Analogique :	57
II.7.4.	La communication avec l'extérieur :	58
II.7.5.	Le microcontrôleur ATmega328:	59
II.8.	Conclusion :	63
CHAPITRE III.	Réalisation pratique :	64
III.1.	Introduction :	64
III.2.	Architecture du robot :	64
III.3.	Réalisation matérielle :	64
III.3.1.	La perception :	64
III.3.2.	La locomotion :	67
a.	Les motoréducteurs :	67
b.	Commande des motoréducteurs :	68
b.1.	Le pond H double L298 :	68
b.2.	L'Arduino Motor shield :	73
III.3.3.	La communication via Bluetooth :	75
III.4.	Réalisation logicielle :	77
III.4.1.	Algorithme :	77
III.4.2.	Code source :	80

III.4.3. Programmation de l'Arduino :	85
III.5. Tests et Résultats :	87
III.6. Conclusion :	91
CONCLUSION GENERALE :	92
Bibliographie :	96

Liste des figures :

Figure 1: Robot mobile de type unicycle	6
Figure 2: Robots de type tricycle.....	7
Figure 3: Robot mobile de type voiture.....	8
Figure 4: Google Self-Driving Car.....	9
Figure 5: Robot de type omnidirectionnel.....	9
Figure 6: Robot Tactique / Surveillance Mastiff HD2-S avec Bras 5 Degrés de Liberté SuperDroid.....	10
Figure 7: Humains et machines unis pour lutter contre les mines terrestres	10
Figure 8: Le robot réalisé par Boston Dynamics PETMAN.....	11
Figure 9: Le robot réalisé par Boston Dynamics Big Dog	12
Figure 10: Le robot Octopodes T8 de Robugtix	12
Figure 11: Robot rampant développer par l'université Carnegie Mellon	13
Figure 12: Principe de fonctionnement d'un Capteur	14
Figure 13 : Courbe d'étalonnage d'un capteur.....	15
Figure 14: Boucle de commande.....	16
Figure 15: Télémétrés infrarouges	22
Figure 16: principe de fonctionnement d'un capteur ultrasonore.....	23
Figure 17: Obstacle perpendiculaire à l'axe du capteur, détecté.....	24
Figure 18: Obstacle en biais non détecté par le capteur	24
Figure 19: Obstacle en biais détecté par le capteur	25
Figure 20: Principe de la stéréovision.....	27
Figure 21: Architecture hybride	31
Figure 22: Le cycle perception/décision/action que doit réaliser un robot.....	32
Figure 23: contrôle de la direction de déplacement du robot	34
Figure 24: Brochage du capteur Parallax 28015	35
Figure 25: Principe de fonctionnement du capteur ultrasonore Parallax 28015.....	36
Figure 26: Protocole de communication du capteur Parallax 28015	36
Figure 27: Considérations pratique (limite du capteur ultrason).....	37
Figure 28: détection d'un objet cylindrique (et représentation du cône d'ouverture)	38
Figure 29: Eclaté d'un moteur à courant continu	40
Figure 30: EMAX Brushless motor	41
Figure 31: Réducteurs Emerson à dentures droite (gauche) et hélicoïdale (droite).....	42
Figure 32: Réducteurs Maxon compacts à dentures droite.....	42
Figure 33: Schéma de fonctionnement d'un réducteur.....	43
Figure 34: montage de commande d'un moteur pas à pas	44
Figure 35: pont H	45
Figure 36: montage en pont H	46
Figure 37: Correction de la position d'un servomoteur.	47

Figure 38: Vue de l'intérieur d'un servomoteur.....	47
Figure 39: la variation du poids maximale que peut supporter un servomoteur selon la distance.	48
Figure 40: Commande de la position d'un servomoteur.....	49
Figure 41: le module HC-05.....	52
Figure 42: Carte Arduino Uno	53
Figure 43: Schéma explicite de la carte Arduino Uno	55
Figure 44: brochage du microcontrôleur ATmega328.....	59
Figure 45: Schéma synoptique du microcontrôleur ATmega328.....	61
Figure 46: Schéma synoptique de l'architecture AVR.....	63
Figure 47: Schéma synoptique du robot.	64
Figure 48: la partie responsable de la perception du robot.....	65
Figure 49: Connecteur du servomoteur.	66
Figure 50: Vue de dessus de la partie perceptive du robot.	66
Figure 51: Motoréducteur DG01D	67
Figure 52: Vue de dessus du châssis du robot.....	68
Figure 53: Brochage du circuit intégré L298.....	69
Figure 54: Contrôle de vitesse via la PWM.....	71
Figure 55: Schéma synoptique du circuit intégré L298.....	71
Figure 56: Arduino Motor shield.	73
Figure 57: Arduino Motor shield monté sur une carte Arduino UNO.	74
Figure 58: branchement des moteurs vers le shield moteur.	75
Figure 59: le module HC-05 FC-114.....	76
Figure 60: schéma de branchement du module HC-05 vers l'arduino.....	77
Figure 61: Algorithme du fonctionnement du robot.	78
Figure 62: Sous-programme de commande du robot via Bluetooth.....	79
Figure 63: Câble USB (fiche A vers fiche B).....	85
Figure 64: l'interface de l'environnement de développement Arduino IDE.....	86
Figure 65: vue des différentes perspectives du robot.	88
Figure 66: Paramètres Bluetooth.	89
Figure 67: Fenêtre du choix de connexion du logiciel Tera Term.	90
Figure 68: Fenêtre principale du logiciel Tera Term.	90
Figure 69: Schéma de la carte Arduino UNO.	94
Figure 70: Schéma du Arduino shield motor	95

INTRODUCTION GENERALE :

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

Dans ce travail on se focalise sur les robots mobiles, qui sont largement utilisés dans les environnements industriels Le plus souvent pour effectuer des tâches répétitives et pénibles. L'intérêt principal des robots mobiles c'est de permettre aux êtres humains d'effectuer des tâches dangereuses dans des endroits inaccessibles et hostiles. C'est pour cela que l'un des domaines les plus populaires où les robots mobiles sont utilisés est le domaine scientifique ; en s'aperçoit rapidement que l'exploration des fonds marins, et de l'espace présente une grande porte pour l'évolution de la science et de notre civilisation, ainsi comme on peut le constaté le développement du domaine de la robotique mobile est très essentiel pour l'avancement de l'humanité. Dans ce genres d'environnements l'autonomie est primordiale, car l'intervention humaine est difficile ou voire même impossible. Si on prend l'exemple des robots explorateurs de planètes, tel le robot qui explore la planète mars Curiosity rover [1], Il est impossible de contrôler en temps réel un robot qui se trouve à une distance de 225 300 000 km de la terre, on ne peut pas négliger le retard que met un signal de commande pour arriver jusqu'à la planète mars, et sa sans parler des bruits (radiation) qui peuvent perturber le signal, c'est pour cela que la capacité pour un robot de prendre lui-même ses propres décisions est essentielle afin qu'il puisse réaliser et coordonner des missions complexes. Le robot doit sélectionner les meilleures actions à effectuer pour réussir adéquatement sa mission.

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 mobile doit être doté de différents capteurs, comme des sonars ou des dispositifs à balayage laser servant à mesurer des distances entre lui-même et les obstacles à proximité. Une fois localisé dans son environnement, le robot doit être capable de se déplacer d'un point à l'autre en trouvant des chemins

efficaces et sécurisés afin d'éviter des collisions avec les obstacles, pour cela les robots mobiles utilisent plusieurs moyens de locomotion selon l'environnement au quel ils sont destinés. De plus, un robot est souvent appelé à communiquer avec les gens ou d'autres agents situés à proximité. Cela peut être fait de différentes façons, comme par une interface wifi, Bluetooth, zigbee, ...Etc.

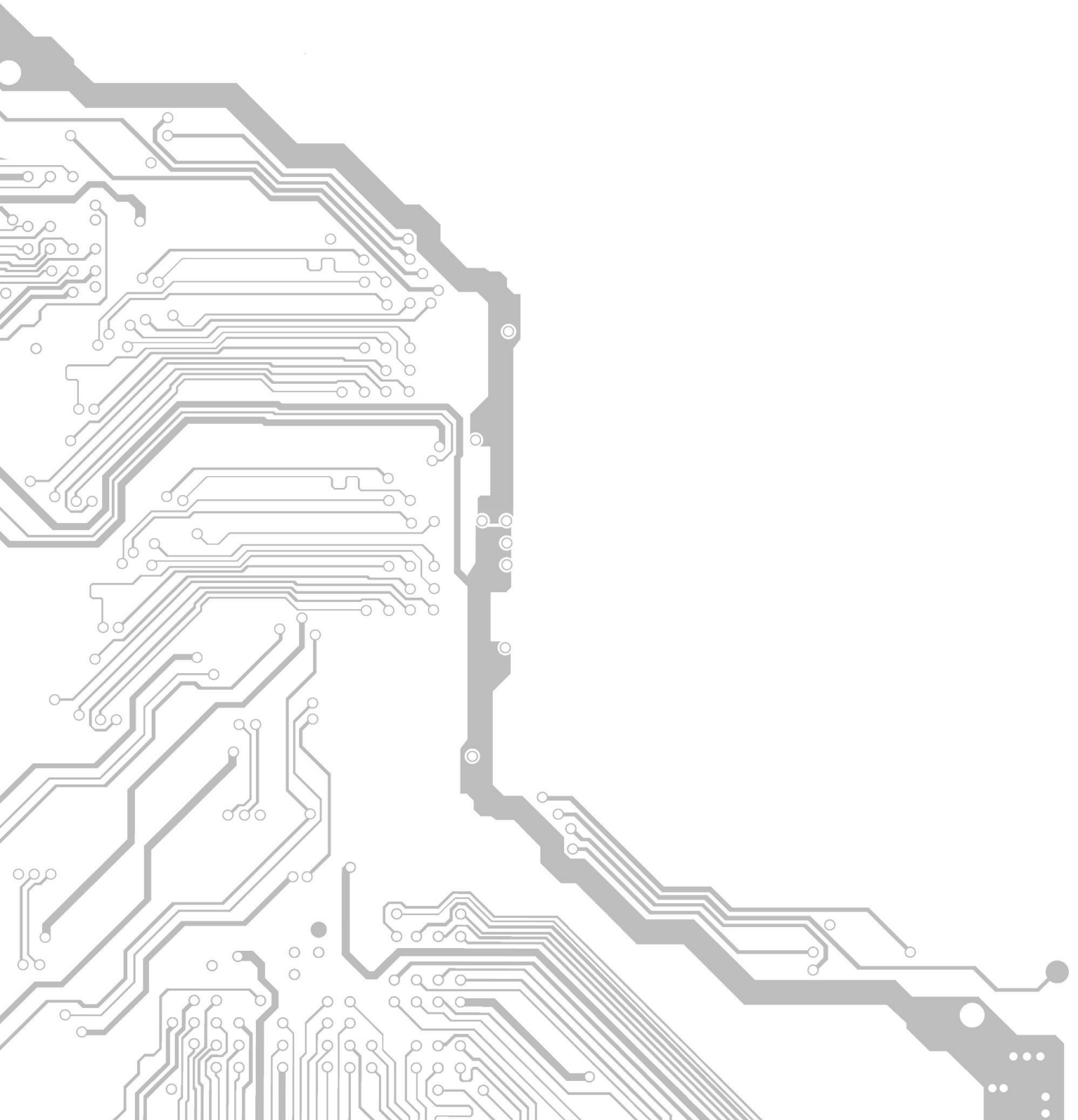
Bien évidemment les robots mobiles sont dotés de bien beaucoup plus d'autres capacités de perception et d'action, car le domaine de la robotique mobile est un domaine hautement multidisciplinaire qui fait l'objet de beaucoup de recherches dans des disciplines très diversifiées. Pour cette raison, dans le présent mémoire, nous faisons abstraction de la plupart de ces capacités robotiques et nous considérons avoir accès à celles-ci puisqu'ils découlent des autres travaux, indépendamment de ceux présentés ici.

Le but de ce mémoire est la conception et réalisation d'un robot mobile autonome avec la possibilité d'être télécommandé à distance par Bluetooth. Le robot sera basé sur une carte Arduino UNO comme composant cœur avec différents organes de perception (capteurs) et de locomotion (moteurs), ainsi qu'un moyen de communication par voie Bluetooth SPP qui offre un moyen de contrôle et de feedback à distance.

Ce mémoire est organisé de la façon suivante. Le premier chapitre introduit des notions de base en robotique mobile. Dans ce chapitre on étudiera les généralités sur les robots mobiles (classification, capteurs en robotique mobile, les actionneurs, ...etc.). Le chapitre deux sera consacré à l'étude théorique de la conception d'un robot autonome mobile, dans ce chapitre on va décrire les différents organes du robot (caractéristiques et principe de fonctionnement). Enfin pour terminer dans le troisième chapitre on va passer à la réalisation pratique du robot (matériel et logiciel).

CHAPITRE 0x01

GENERALITES SUR LES ROBOTS MOBILES



CHAPITRE I. Généralités sur les robots mobiles :

I.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 le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens.

On peut estimer que les robots mobiles à roues constituent le gros des robots mobiles, c'est pour cela qu'on se limitera volontairement à une présentation des robots mobiles à roues et des 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 en ambiance hostile (nucléaire, marine, spatiale, lutte contre l'incendie, surveillance...etc.). L'aspect particulier de la mobilité impose une complexité technologique et méthodologique qui s'ajoute en général aux problèmes rencontrés par les robots manipulateurs. La résolution de ces problèmes passe par l'emploi de toutes les ressources disponibles tant au niveau technologique (capteurs, motricité, énergie) qu'à celui du traitement des informations par l'utilisation des techniques de l'intelligence artificielle. L'autonomie du robot mobile est une faculté qui lui permet de s'adapter ou de prendre une décision 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 de planètes, comme le fameux *Curiosity rover* [1] qui est en phase d'exploration de la planète mars actuellement. L'autonomie est un point fondamental puisque la télécommande est alors impossible par le fait de la durée du 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 des capteurs, comme des sonars et un dispositif à balayage laser servant à mesurer des distances entre lui-même et les obstacles à proximité.

Une fois localisé dans son environnement, le robot doit être capable de se déplacer d'un point à l'autre

en trouvant des chemins efficaces et sécurisés afin d'éviter des collisions avec les obstacles. De plus, un robot est souvent appelé à communiquer avec les gens ou d'autres agents situés à proximité. Cela peut être fait de différentes façons, comme par communication vocale ou depuis une interface graphique.

I.2.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 capable de s'adapter à certaines variations de ses conditions de fonctionnement. Elle est dotée de fonctions de perception, de décision et d'action, lui permettant ainsi d'être capable d'effectuer des tâches diverses, de plusieurs manières, et accomplir correctement sa tâche, même s'il rencontre de nouvelles 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 degré 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 peuplés d'obstacles, nécessitant une adaptabilité à la tâche.

I.2.1. Classification des robots mobiles :

La classification des robots se fait selon plusieurs critères, mais principalement elle se fait selon trois critères : le système de locomotion, le degré d'autonomie, le domaine d'application.

Il est important de noter que le terme « robots mobiles » bien que désignant l'ensemble des robots à base mobiles (par opposition aux robots manipulateurs), est généralement employé pour désigner les robots mobiles *terrestres*. Ce sont ces robots qui sont considérés dans ce manuscrit.

a. 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 ou voire même absent

un certain nombre de tâches, dans un environnement inconnu. Selon ce critère on peut classer les robots mobiles selon les catégories suivantes :

a.1. Robot télécommandé :

Ce sont des robots commandés par un opérateur (machine ou être humain), qui leurs dicte chaque tâche élémentaire à faire (avancer, reculer, tourner à droite...etc.).

a.2. Robot semi-autonome :

Ce type de robot effectuent un certain nombre de tâche par eux même d'une façon complètement autonome mais peuvent être interrompu pour recevoir des commandes de control par un opérateur.

a.3. Robot autonome :

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

L'autonomie est la capacité propre d'un système sans équipage, à capter, percevoir, analyser, communiquer, planifier, prendre des décisions et agir afin d'atteindre les buts qui lui ont été assignés par un opérateur humain à l'aide d'une interface homme/machine dédiée.

b. Classification selon le domaine d'application :

L'un des plus grand avantage des robots mobile est le faite que leurs domaine d'application est illimité, c'est pour cela que nous allons présenter ici quelques domaines d'application :

b.1. Les robots industriels et de service :

Il existe des robots mobiles destinés à des applications industrielles. Celles-ci concernent principalement le transport et la distribution (dans les usines, les mines, les hôpitaux et les ateliers), le nettoyage, l'entretien et la maintenance, la surveillance et la manutention. Quant aux robots de service, ils sont destinés à aider des handicapés moteurs, à guider les aveugles et à piloter des voitures automatiques.

b.2. Les robots militaires :

Les applications militaires de la robotique mobile sont nombreuses. Ce champ d'application présente l'intérêt de fournir des spécifications serrées telles que la vitesse des véhicules, leurs capacités de

franchissement des obstacles (la robustesse de ce type de robots est essentielle), et leur rapidité de réaction.

b.3. Les robots de laboratoire :

Afin de valider des travaux théoriques sur la perception ou en planification de mouvement, de nombreux laboratoires travaillent dans le domaine de la robotique. À l'origine tous les robots viennent d'un laboratoire, issu des travaux de recherche dans ce domaine.

c. Classification selon le type de locomotion :

c.1. Les robots mobiles à roues :

La mobilité par roues est la structure mécanique la plus utilisée. 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 quatre classes principales de robots à roues.

c.1.1. Robots unicycle :

Un robot de type unicycle est actionné soit par une seule roue ou 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 des robots de type unicycle est donné en Figure 1.

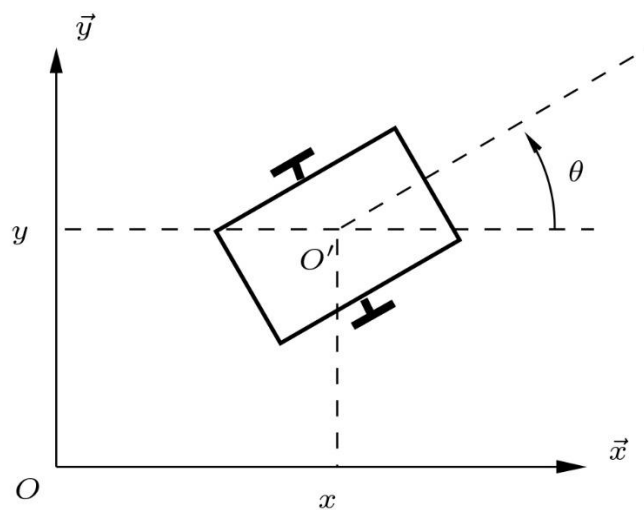


Figure 1: Robot mobile de type unicycle

Un robot unicycle est un robot qui bouge dans un plan 2D ayant une certaine vitesse allant d'avant, mais zéro mouvement latéral instantané, car les robots unicycle sont des systèmes non-holonomie. Il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

c.1.2. Robots tricycle :

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal (voir la Figure 2). Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable.

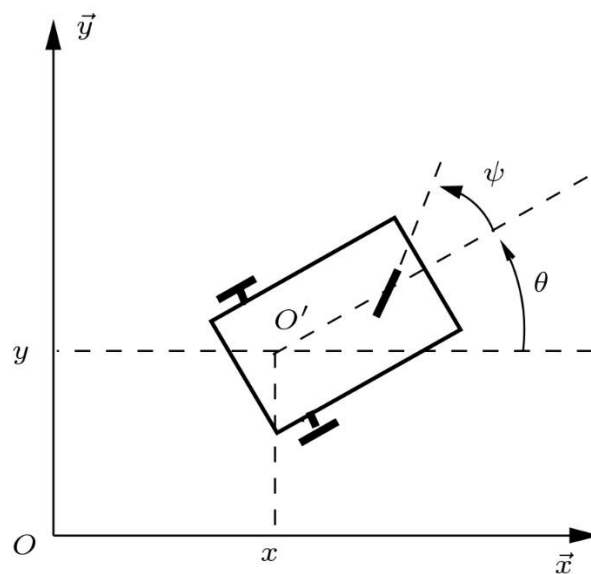


Figure 2: Robots de type tricycle

C'est un robot non-holonomie. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes.

c.1.3. Robots voiture :

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe voire Figure 3. Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être

ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé.

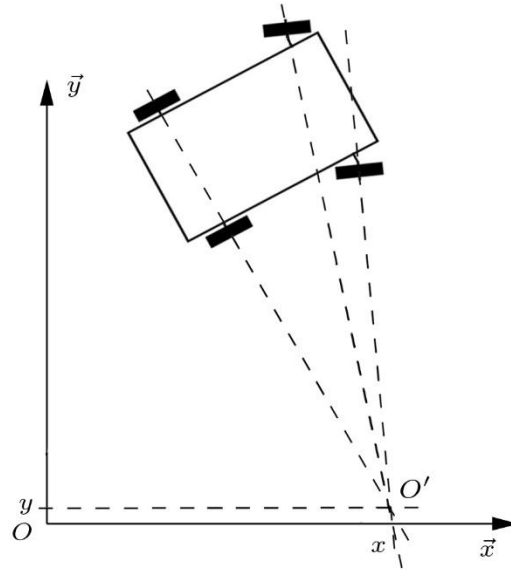


Figure 3: Robot mobile de type voiture

On parle de robot dès lors que la voiture considérée est autonome, donc sans chauffeur ni télé-pilotage. L'un des plus grands projets prometteur dans ce domaine actuellement est le projet « Google Self-Driving Car Project » (voir la Figure 4), ce sont des voitures autonomes truffées de capteurs leurs permettant d'avoir un champ de vision jusqu'à deux terrains de football dans toutes les directions. Ainsi détecté toutes sort d'objets (obstacles) comprenant des cyclistes, d'autre véhicules, des oiseaux, ou voire même des sachets en plastique. Ces informations aident à une navigation en toute sécurité sans intervention humaine.

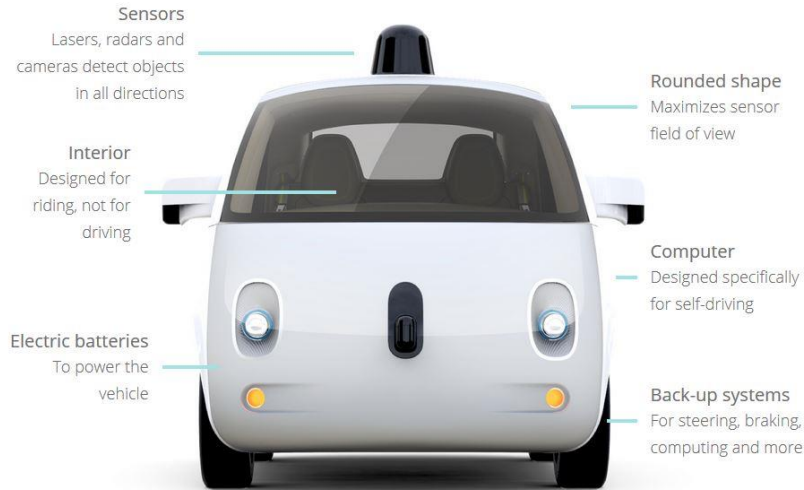


Figure 4: Google Self-Driving Car

c.1.4. Robots mobiles omnidirectionnels :

Un robot mobile est dit *omnidirectionnel* si l'on peut agir indépendamment sur les vitesses : vitesse de translation selon les axes x et y et vitesse de rotation autour de z voire Figure 5. Il peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral.

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

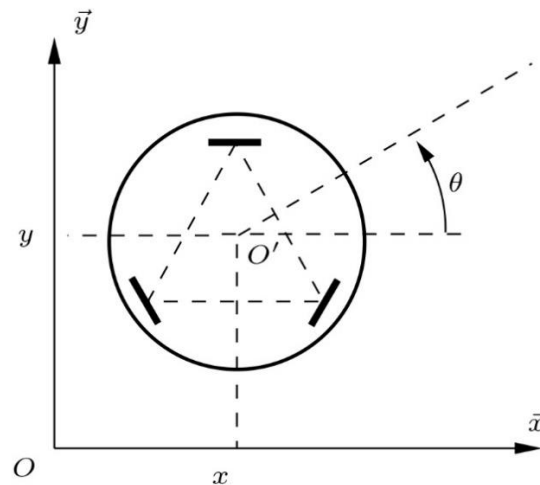


Figure 5: Robot de type omnidirectionnel.

d. 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 démineurs voire les figures suivantes :



Figure 6: Robot Tactique / Surveillance Mastiff HD2-S avec Bras 5 Degrés de Liberté SuperDroid



Figure 7: Humains et machines unis pour lutter contre les mines terrestres

e. Les robots mobiles marcheurs :

Les robots mobiles marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est difficile, dangereux ou impossible à l'homme. Leur anatomie à nombreux degrés de liberté permet un rapprochement avec les robots manipulateurs. La locomotion est commandée en termes de coordonnées articulaires. Les méthodes de commande des articulations définissent le concept d'allure qui assure le déplacement stable de l'ensemble. Les différentes techniques étudiées se rapprochent de la marche des êtres humaine (robots de type humanoïde voir Figure 8). C'est des robots bipèdes qui se balancent d'une façon complètement autonome.

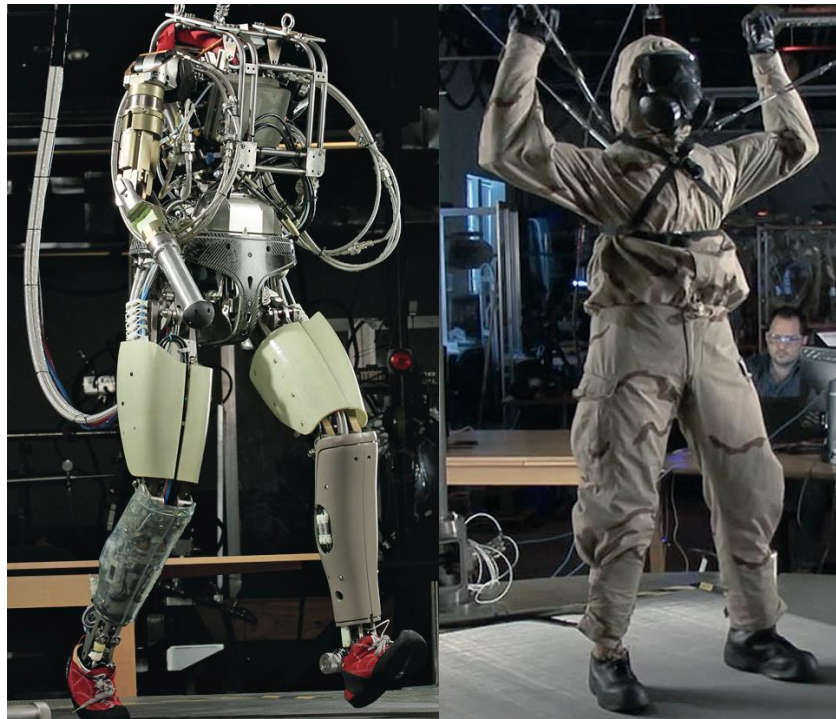


Figure 8: Le robot réalisé par Boston Dynamics PETMAN

Et des animaux (soit à quatre pattes de type cheval voir Figure 9), dans cette catégorie je voudrais mentionnée que cette année plus précisément le 29 mai 2015 le robot **cheetah** de Boston Dynamics a marqué l'histoire car il a réussi une succession de sauts, c'est la première fois qu'un robot à quatre jambes à réussi à courir et sauter des obstacles de façon autonome.



Figure 9: Le robot réalisé par Boston Dynamics Big Dog

Et notamment de celle des insectes (à quatre pattes, six pattes ou même plus de type araignée voir Figure 10). L'adaptation au support est un problème spécifique aux marcheurs. Il consiste à choisir le meilleur emplacement de contact en alliant l'avance et la stabilité avec l'aide de capteurs de proximité, de contact ou de vision.



Figure 10: Le robot Octopodes T8 de Robugtix

f. Les robots mobiles rampants :

La reptation est une solution de locomotion pour un environnement de type «tunnel» qui conduit à réaliser des structures filiformes (voir Figure 11). Le système est composé d'un ensemble de module

ayant chacun plusieurs mobilités. Ici aussi 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 voisins.

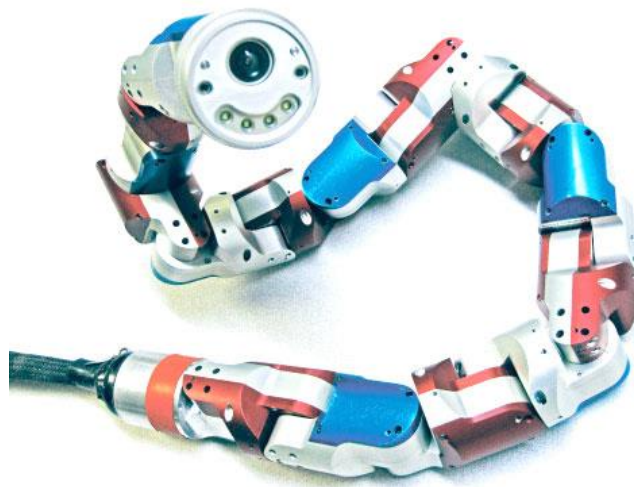


Figure 11: Robot rampant développé par l'université Carnegie Mellon

Le robot de la Figure 11 a été développé afin d'explorer les tuyaux de la centrale nucléaire Zwentendorf en Autriche qui a été construite en 1970.

I.2.2. Notion de Perception :

La notion de *perception* en robotique mobile est relative à la capacité du système à recueillir, traiter et mettre en forme des informations utiles au robot pour 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.

I.2.3. 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 permettant d'appréhender, en temps réel, son environnement.

Le capteur est un instrument qui permet de déceler 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 voire Figure 12. Il va assurer la mesure permanente des grandeurs réelles en question.



Figure 12: Principe de fonctionnement d'un Capteur

Ainsi on obtient la grandeur électrique **S** de sorte en fonction de la mesurande **m** comme représenté dans la figure suivante :

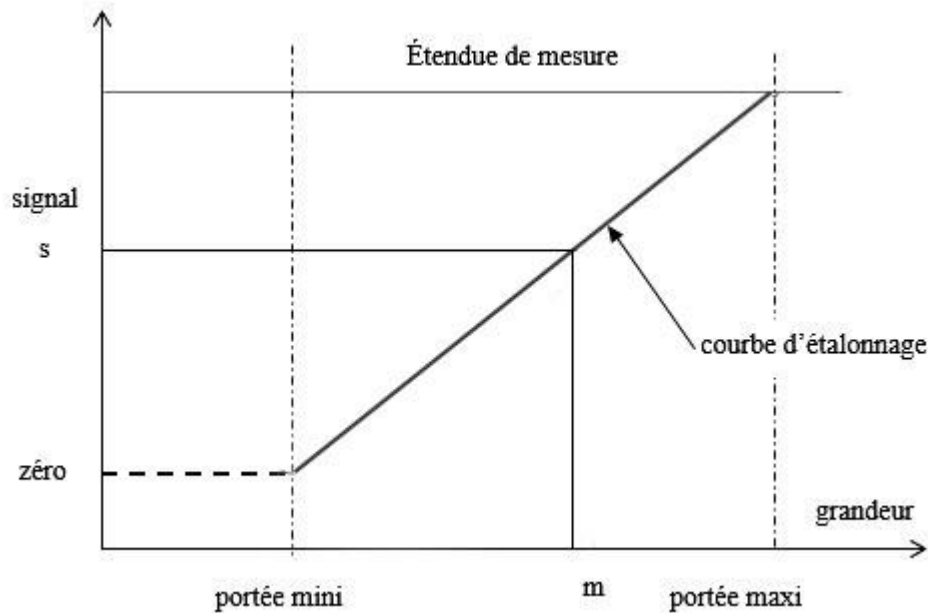


Figure 13 : Courbe d'étalonnage d'un capteur

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

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

Les capteurs typiquement installés sur un robot mobile sont des capteurs ultrasonores, un capteur laser de proximité, des encodeurs de roues (odomètres), une ou deux 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 la forme de l'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 son environnement. Dans le premier cas, à l'image de la perception chez les êtres vivants, on parle de *proprioception* et donc de *capteurs proprioceptifs*. On trouve par exemple dans cette catégorie les capteurs de position ou de vitesse des roues et les capteurs de charge de la batterie. Les capteurs renseignant sur l'état de l'environnement, donc de ce qui est extérieur au robot lui-même, sont eux appelés *capteurs extéroceptifs*. Il s'agit de capteurs donnant la distance du robot à l'environnement, la température, signalant la mise en contact du robot avec l'environnement, etc.

La Figure 14 montre les deux types de capteurs :

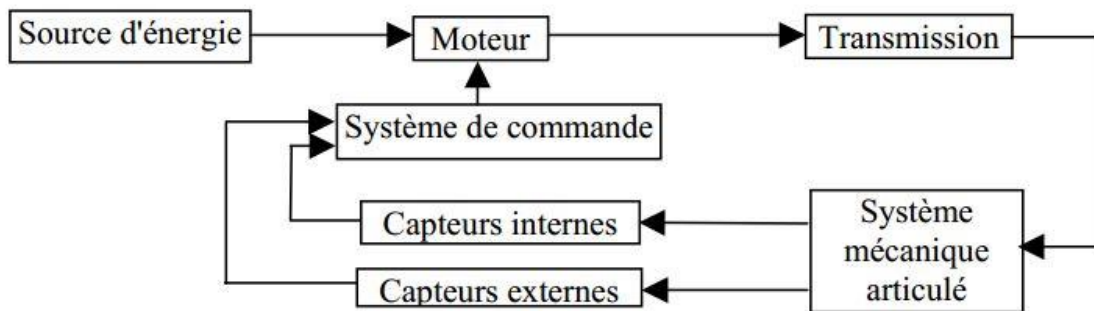


Figure 14: Boucle de commande

Il existe une autre classe de capteurs non inclus dans les boucles d'asservissement, dont l'intérêt est de veiller sur l'état du robot lui-même que sur l'environnement. Ce sont des capteurs de sécurité qui se déclenchent en cas de mal fonctionnement ou de panne des boucles internes ou externes. Ce type de capteur génère un ordre d'arrêt du robot lorsque le dépassement de certaines valeurs a lieu.

a. Capteurs internes :

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

Les informations sensorielles gérées dans ce cadre sont généralement des vitesses, des accélérations, des angles de giration 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 les gyroscopes et les gyromètres, les capteurs inertiels composites, les inclinomètres et les magnétomètres. Ces capteurs sont en majorité de type inertielle (c.à.d. ces capteurs sont utilisés pour déterminer le mouvement absolu d'un véhicule).

a.1. Les capteurs de déplacement :

a.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 une précision qui dépend de la résolution du codeur. Des méthodes fondamentales sont utilisées pour la mesure de position. Dans une première méthode, le capteur est lié à la partie mobile du robot (ex : les roues dans notre cas) et fournit un signal dont la variation traduit la position. Il existe des codeurs digitaux absolus où la position est déterminée par la lecture optique d'une graduation codée en binaire portée par un disque solidaire. Dans la seconde le capteur délivre une impulsion à chaque déplacement élémentaire : la position est déterminée par comptage des impulsions émises, ou décomptage 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 cinématique du véhicule.

a.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. Généralement, trois accéléromètres sont associés aux axes du repère du robot. La valeur mesurée selon l'axe vertical comprend l'effet de la gravité, qui doit être compensé afin de pouvoir accéder à l'accélération verticale propre du robot. Pour cela, il est nécessaire soit de stabiliser la plateforme inertielle dans le plan horizontal, soit d'utiliser des capteurs supplémentaires, les inclinomètres, qui fourniront une estimation de la verticale locale permettant la compensation des mesures. La première solution se révèle coûteuse et délicate lors de la mise en œuvre ; de ce fait, elle est peu utilisée en robotique. La seconde solution est simple, du point de vue de la mise en œuvre, mais présente l'inconvénient de ne fournir une information fiable qu'en régime quasi statique. En effet, la bande passante des inclinomètres est très basse (de 0 à 2.5 Hz) vis-à-vis de celle des accéléromètres (de 100 à 1000 Hz). En raison de leur mode de calcul (double intégration de la mesure), les positions obtenues à partir d'accéléromètres sont entachées d'erreurs, principalement liées aux vibrations et à une mauvaise estimation de la verticale. Ces erreurs se manifestent sous la forme d'une dérive (couramment de l'ordre de 1 à 8 cm/s). En raison de l'importance de ces dérives, les mesures de position et d'orientation estimées à partir de mesures brutes provenant de capteurs inertiels (gyromètres et accéléromètres) sont inutilisables sans un filtrage qui prend en considération une modélisation des termes de dérive.

a.2. 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.

a.2.1. 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, connu pour leur grande précision, ont vu leur prix chuter et sont donc devenus une solution attractive pour la navigation en robotique mobile.

a.2.2. Les gyromètres :

Le gyromètre est un capteur qui permet de mesurer une vitesse angulaire. Deux technologies sont actuellement disponibles, permettant un rapport coût/performance compatible avec des applications robotiques : les gyromètres piézoélectriques et les gyromètres à fibres optiques. Dans le premier cas, le principe de mesure exploite la variation d'amplitude d'un oscillateur piézoélectrique, induite par la force de Coriolis due au mouvement de rotation appliqué au capteur. Ce type de capteur à faible coût présente des dérives de l'ordre de $0.2^\circ/s$, dont l'influence sur le calcul de l'orientation peut être limitée grâce à l'utilisation de techniques de filtrage temporel. Il faut également noter que ces capteurs sont sensibles aux variations de températures. Les capteurs à fibre optique, plus récents, sont basés sur la différence de trajets optiques entre deux faisceaux laser qui se propagent en sens opposés dans une boucle circulaire de fibres optiques. Lorsque la boucle est immobile, les trajets optiques sont identiques. Soumise à un mouvement de rotation, la différence de trajet optique est alors proportionnelle à la vitesse angulaire (effet Sagnac). Là encore, les mesures d'orientation fournies par l'intégration des vitesses angulaires sont entachées d'erreurs croissantes avec le temps, mais dans un rapport environ cinquante fois plus faible que dans le cas des capteurs piézoélectriques (de l'ordre de $0.005^\circ/s$). En contrepartie, en raison de leur principe de mesure, ils possèdent une vitesse minimum de fonctionnement assez élevée, de l'ordre de $0.05^\circ/s$. A titre d'exemple, en considérant un robot mobile admettant une erreur angulaire de 1° au terme d'un déplacement linéaire de 10 m, cette erreur ne sera observable par un gyromètre à fibre optique que si la vitesse d'avancement du robot est supérieure à 0.5 m/s.

a.2.3. 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.

a.2.4. Le magnétomètre :

Le magnétomètre qui est aussi appelé compas magnétique mesure la direction du champ magnétique terrestre pour 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 et d'être peu coûteuse.

b. Capteurs externes :

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 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 des préventions et sécurité.

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

b.1. Les capteurs télémétriques :

Le fonctionnement des capteurs utilisés classiquement en robotique mobile est présenté dans les sections suivantes. Les lasers d'abord puis les capteurs à infrarouge sont introduits. Les capteurs à ultrasons sont ensuite développés. En effet, en raison de leur coût faible par rapport aux autres capteurs et leurs disponibilités, les capteurs ultrasons ont été choisis dans notre travail.

b.1.1. Les capteurs laser :

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 est souhaité de mesurer la distance, et puis en mesurant le déplacement de l'onde réfléchi de l'objet en déduit la distance. Une étude très détaillée de ce type de capteur est proposée 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, est importante. Une précision de l'ordre du dixième de 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, étant donnée la vitesse de propagation des ondes électromagnétiques, 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 alors 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].

Plusieurs principes sont utilisés pour la mesure de distance avec des dispositifs optiques. Le premier est la triangulation : des mesures d'angles permettent de déterminer la distance de la cible. Les mesures réalisées présentent une résolution correcte et un temps d'acquisition compatible avec la robotique mobile. Mais la précision décroissante avec la distance, il existe des zones d'ombre importantes. Le second principe est la mesure d'intensité. Une onde lumineuse étant émise et réfléchiée par une cible, un récepteur mesure le niveau du signal qu'il reçoit pour en déduire la longueur de la course du faisceau. L'inconvénient majeur provient d'une atténuation très différente suivant l'état de surface de la cible. Des systèmes à plusieurs récepteurs commencent à voir le jour qui permettent de s'affranchir en partie de l'état de surface de la cible. Le troisième est l'interférométrie. Ces mesures à incréments atteignent une très grande précision mais connaissent un problème d'ambiguïté de mesure à un modulo près. Ils doivent donc être couplés avec un autre dispositif pour obtenir une approximation de la distance mesurée. Ces

systemes sont encombrants et sensibles aux vibrations, ce qui les rend peu adaptés à la robotique mobile. Le dernier est la mesure par temps de vol. Pour de grandes distances, une mesure directe est envisageable. Pour des distances plus courtes, une modulation sinusoïdale ramène à une mesure de déphasage sur l'onde modulée. On peut aussi combiner les deux.

b.1.2. 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 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 phases 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 cela est représenté dans la Figure 15 :

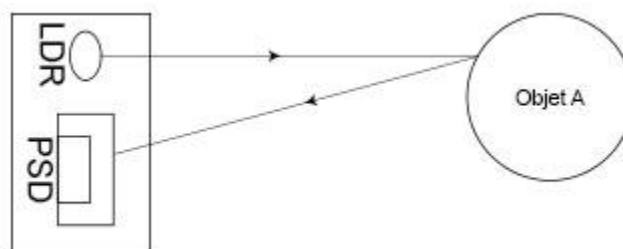


Figure 15: Télémètres infrarouges

b.1.3. 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 20Hz sont appelés *infrasons*.), les fréquences couramment utilisées dans ce type de technologie vont de 20 kHz à 200 kHz. Les ultrasons émis se propagent dans l'air et sont réfléchis partiellement lorsqu'ils heurtent un corps solide voire la Figure 16, 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 ultrason pour faire un allé retour entre capteur et le corps distant.

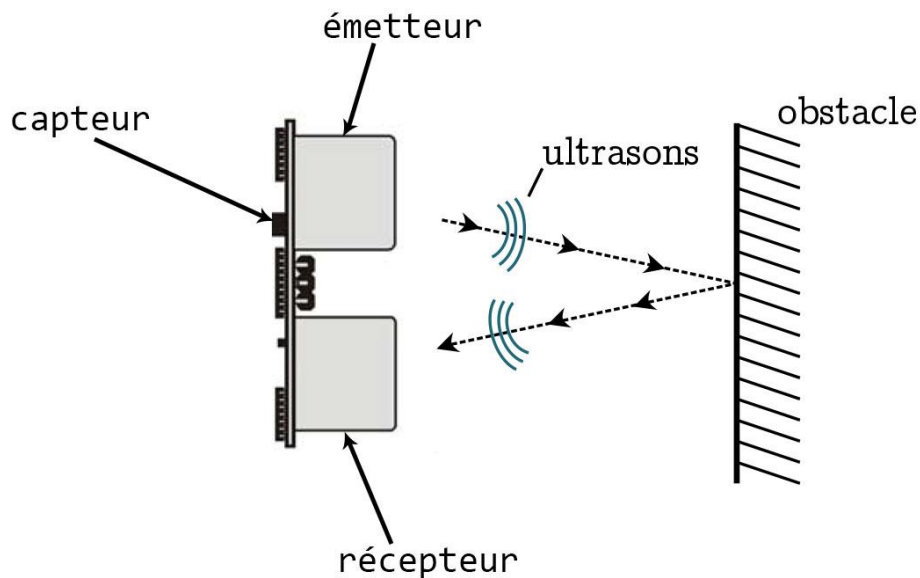


Figure 16: principe de fonctionnement d'un capteur ultrasonore.

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.

Seuls les objets perpendiculaires à un rayon émis sont détectables. L'onde émise n'est pas très directive (typiquement entre 20 et 30 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 Figure 17, Figure 18, Figure 19). On sait seulement que l'onde mesurée est réfléchiée perpendiculairement à l'obstacle rencontré.

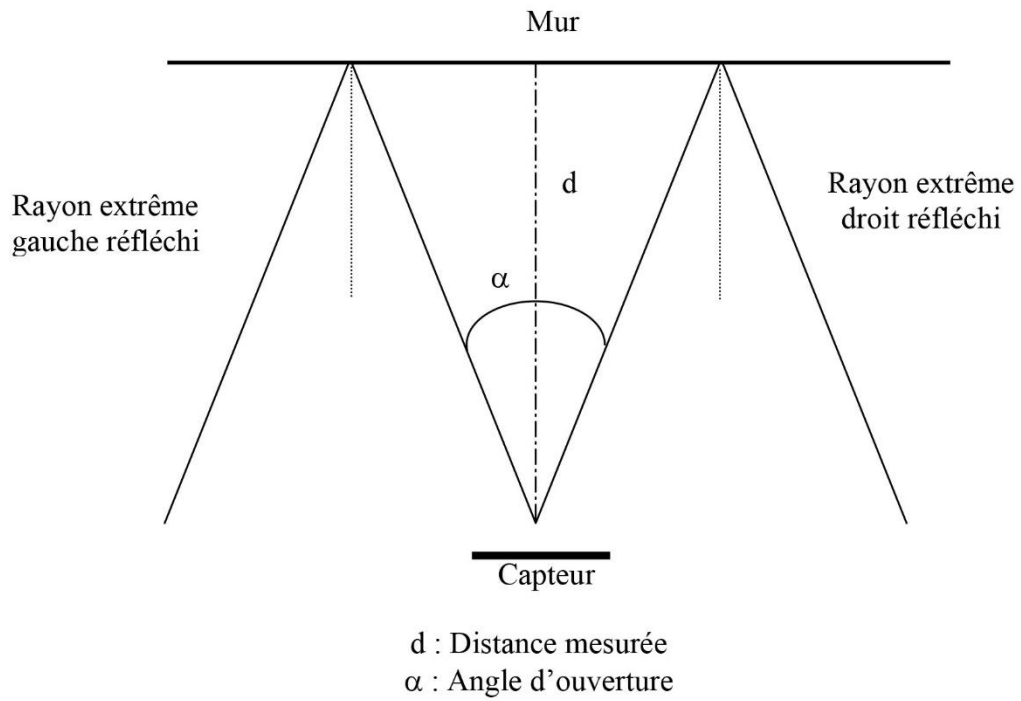


Figure 17: Obstacle perpendiculaire à l'axe du capteur, **déecté**.

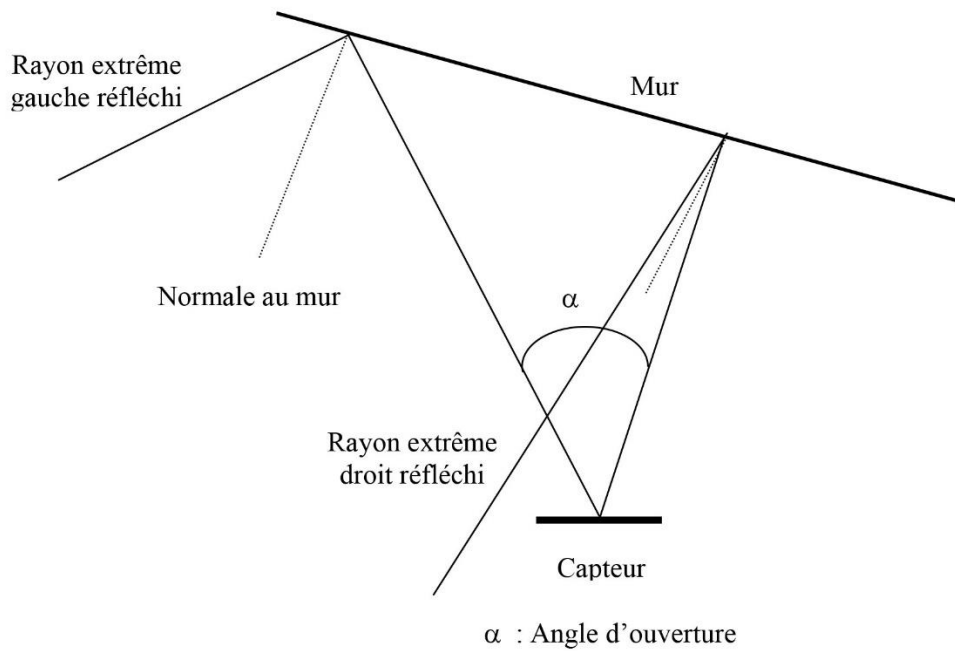


Figure 18: Obstacle en biais **non déecté** par le capteur

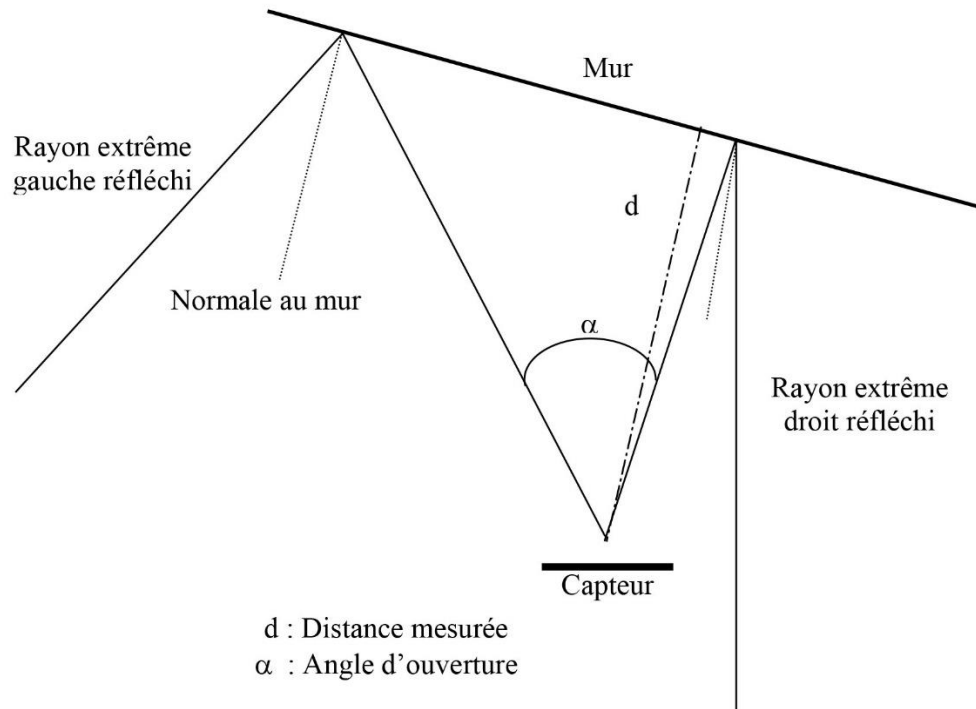


Figure 19: Obstacle en biais **défecté** par le capteur

Rajoutant à ses problèmes classiques du rebond multiple de l'écho multiple 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 certains cas, un second écho peut permettre de détecter un obstacle derrière un autre (objet derrière un pied d'une table par exemple). 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.

Le capteur utilisé 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 un obstacle. Ainsi, la réception doit être bloquée pendant l'émission sinon, la membrane vibrant, une onde est détectée comprise comme un 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 $340.29 \times 0.00238 = 0.81\text{m}$ (l'onde fait l'aller-retour). La zone aveugle est donc $0.81/2 = 0.405\text{m}$. Dans les faits, elle est un peu plus importante et vaut environ 0.44m.

b.2. Les systèmes de vision :

Les systèmes de vision en robotique mobile sont basés sur l'utilisation d'une caméra, 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, précision et 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 d'un tel capteur sont généralement liées à l'adjonction d'un autre capteur. Dans ce cadre nous pouvons identifier les techniques suivantes :

b.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 chez l'être humain. Dans Figure 20 on décrit plus précisément le principe de la stéréovision. Tout point M visible depuis les deux caméras se projette en P1 sur le plan image de la première caméra et en P2 sur le plan image de la deuxième caméra. Inversement, connaissant deux points en correspondance P1 et P2, ainsi que la distance D séparant les centres optiques O1 et O2 des deux caméras, les coordonnées du point P sont données par l'intersection de (O1P1) et (O2P2).

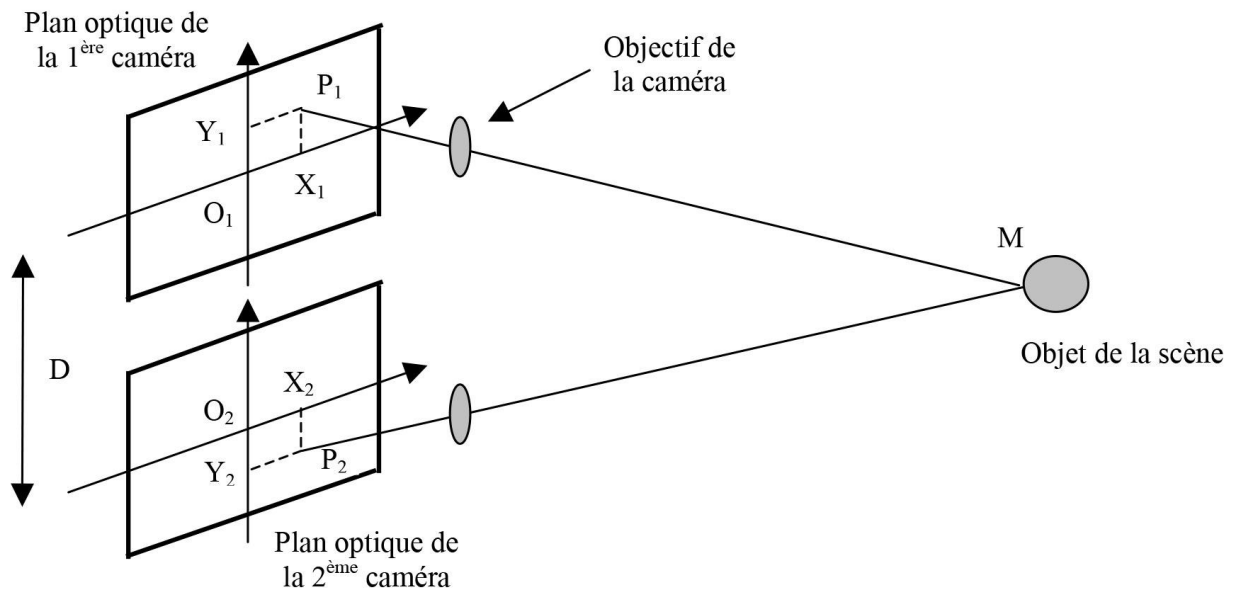


Figure 20: Principe de la stéréovision

La stéréovision apparaît comme un des moyens de perception les plus performants en robotique. Toute la problématique de la stéréovision réside dans la robustesse de la phase de mise en correspondance des informations : elle est souvent liée à de nombreuses ambiguïtés mais aussi à des temps de calcul très importants c'est pour cela des systèmes de traitement d'information rapide sont nécessaires.

La stéréovision est très utilisée en robotique mobile que ce soit pour la modélisation des scènes d'intérieurs [8] ou pour établir une cartographie de l'environnement [9]

b.2.2. Les capteurs à triangulation active (lumière structurée) :

L'obtention de la 3D est gérée 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 [3].

Plusieurs modèles de lumière structurée peuvent être utilisés :

- Capteur à simple triangulation : les capteurs à simple triangulation acquièrent la distance d'un seul point lumineux à la fois. Une application célèbre est celle du véhicule d'exploration Pathfinder destiné à être envoyé sur la planète Mars.
- Les capteurs par plan de lumière : la source de lumière projetée n'est plus un point, mais un plan de lumière. Un système de localisation basé sur ce principe a été développé dans [12].
- Capteurs à plusieurs plans de lumière : il s'agit dans ce cas de projeter un motif tel que des lignes ou de petits disques sur la scène observée.

La technique de triangulation active est plus robuste 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.

b.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 l'élément ou les éléments ajoutés à une caméra, on pourra distinguer quatre techniques pour obtenir une image omnidirectionnelle.

- Génération d'images par utilisation de plusieurs caméras Cette première technique consiste à utiliser plusieurs caméras couvrant un champ de vision égal à 360° : par exemple quatre caméras séparées par des angles de 90°.
- Génération d'images multiples par rotation d'une caméra Cette deuxième 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 effectuer une rotation complète avant d'obtenir une vue de 360°.
- Utilisation de lentille spéciale Cette troisième méthode consiste à ajouter un ensemble optique sur la caméra CCD (Charge Coupled Device) afin de dévier les rayons lumineux, comme par exemple l'œil de poisson.
- Utilisation de miroirs convexes.

Ils sont plus utilisés en robotique mobile au détriment de la vision monoculaire [13], [14], [15].

I.2.4. Actionneurs :

Pour bouger à l'intérieur de son environnement et interagir avec celui-ci, les robots mobiles sont équipés de ce qu'on appelle des « Actionneurs ». Un actionneur est un organe qui transforme l'énergie qui lui est fournie en un phénomène physique utilisable. 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, qui sont classifiés selon trois catégories :

- Actionneurs hydrauliques : explore la propriété d'incompressibilité des fluides.
- Actionneurs pneumatiques : explore les propriétés de compression et de dilatation.
- Actionneurs électriques : explore plusieurs 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 tels les moteurs à courant continu, les moteurs pas à pas, les servomoteurs, et les motoréducteurs.

I.2.5. Architecture décisionnelles :

Un robot mobile autonome est un système doté de capacités décisionnelles lui permettant d'accomplir un certain nombre de tâches, dans un environnement non complètement connu d'avance, avec une intervention humaine réduite ou voire même absent.

Un robot est composé d'un ensemble de modules, chacun étant responsable d'une ou de 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 peuvent être généralement classées en trois grandes catégories : délibérative, comportementale et hybride [16].

a. 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 de y degrés, et ainsi de suite. Lorsqu'un changement dans l'environnement est perçu, 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 ce type d'architecture est que la génération de plans précis demande beaucoup de ressources (temps de calcul et mémoire).

b. Architecture comportementale :

L'architecture comportementale, proposée par Brooks [17], est inspirée par le comportement des insectes. L'idée générale est de développer plusieurs petits modules simples et indépendants les uns des autres et, une fois regroupés, un comportement plus intelligent émerge sans qu'il ait été spécifiquement 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, le suivi de chemin, le suivi d'objets de couleur ou la 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'elles ne contiennent 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 mots, ces tâches complexes ont besoin d'être planifiées [16].

c. Architecture hybride :

Les limitations des deux types d'architecture précédentes justifient l'émergence récente des architectures hybrides [25], 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 21 illustre un exemple d'architecture hybride où un séquenceur s'occupe des capacités délibératives.

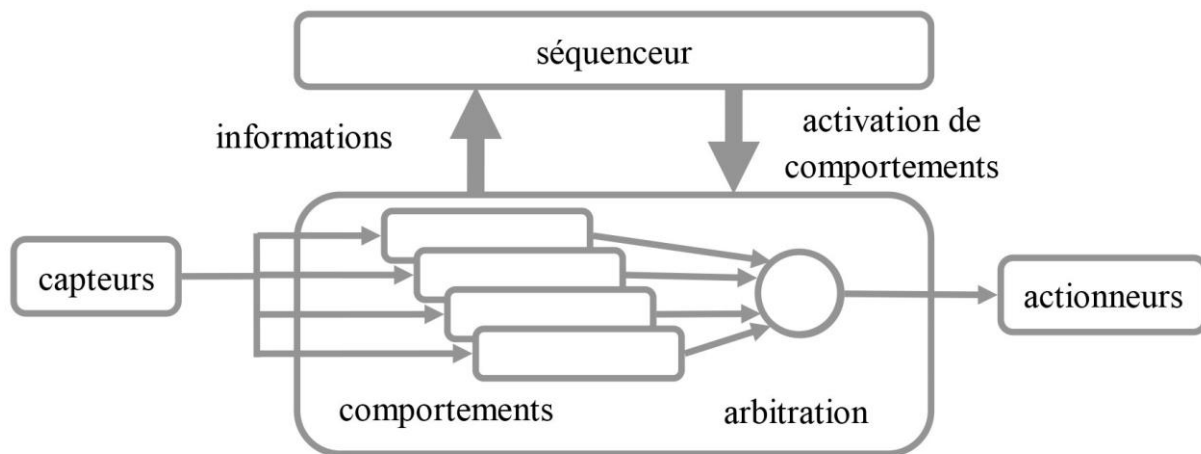


Figure 21: Architecture hybride

I.2.6. 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 de manière autonome.

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 voire Figure 22. Le rôle de ces fonctionnalités se fait comme suite :

- **Perception** : Il s'agit essentiellement de détecter les 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 le type de mouvement à exécuter en générant des consignes de vitesses appropriées à envoyer au robot (pour un mouvement réactif par exemple) 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 par exemple le type de commande adapté.

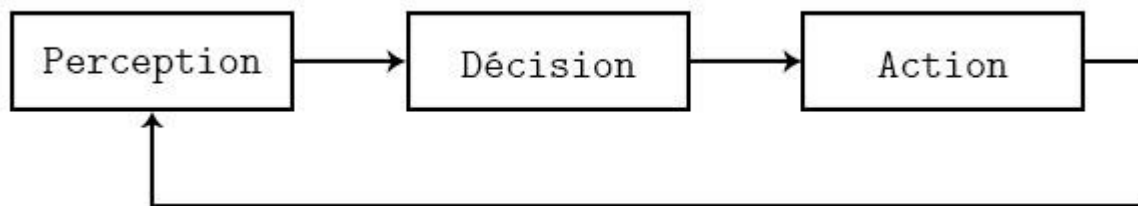


Figure 22: Le cycle perception/décision/action que doit réaliser un robot

I.3.Conclusion :

Dans ce chapitre on s'est intéressé à résumer toutes les notions de base nécessaires à la compréhension du domaine de la robotique mobile. Ainsi qu'un panorama de techniques appliquées dans la planification et la navigation d'un robot mobile autonome. Nous avons aussi abordé dans ce chapitre l'architecture d'un robot mobile et les différents composants matériels et logiciels le constituant.

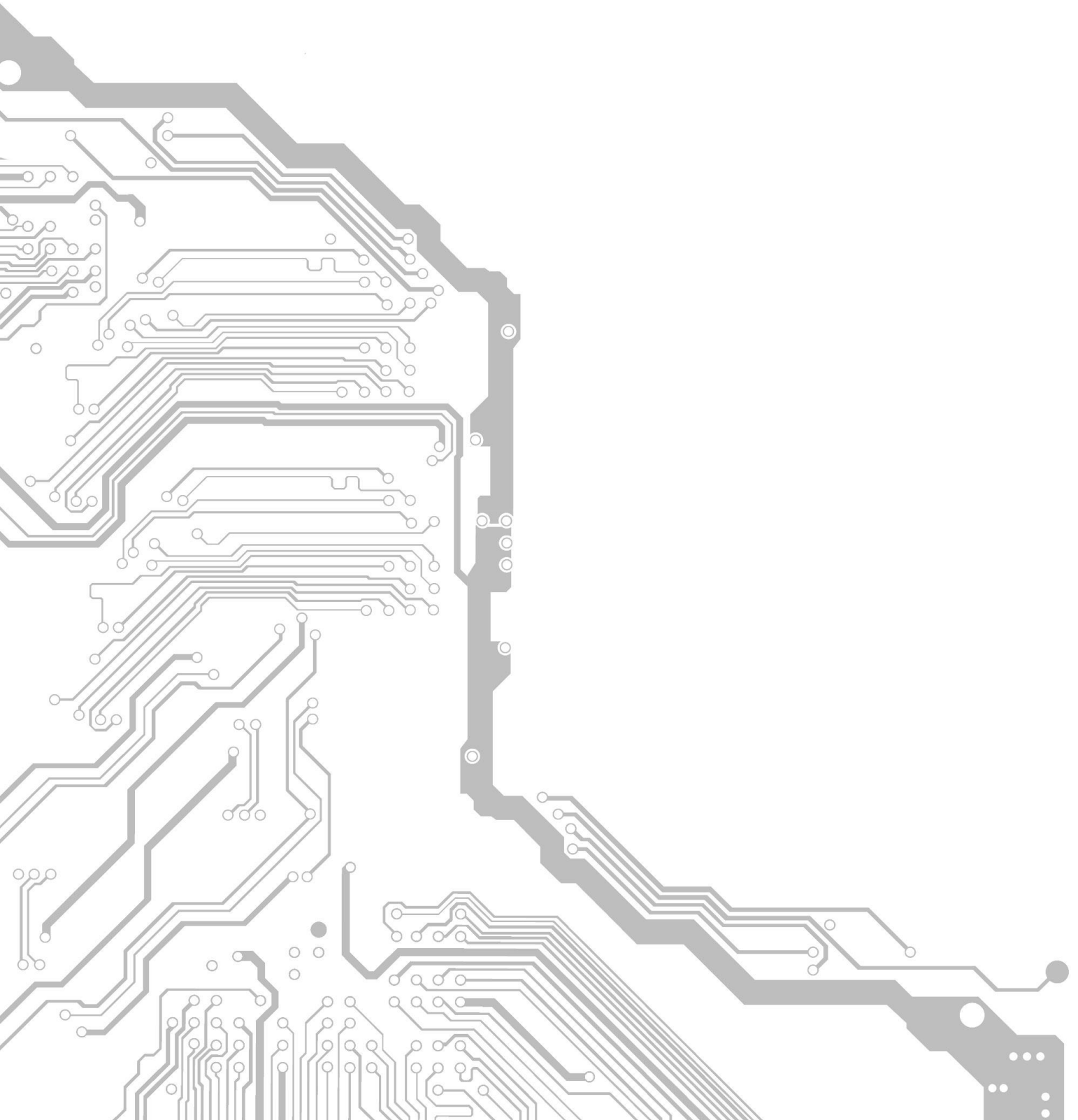
On a étudié les différentes classes des robots mobiles qui sont catégorisés selon différents critères, soit selon le degré d'autonomie, ou le domaine d'application, ou bien selon le type de locomotion utilisé.

Ensuite nous avons également présenté comment un robot mobile aperçoit son environnement grâce aux informations recueillies par les différents capteurs utilisés en robotique mobile, et comment ces informations sont utilisées par le robot suivant une architecture décisionnelle donnée pour naviguer dans son environnement en utilisant des actionneurs.

Après avoir vu les notions de base de la robotique mobile, il nous reste plus que de passer à l'étape de conception d'un robot mobile autonome utilisant ces notions dans le 2^{ème} chapitre, et puis à la réalisation pratique dans le 3^{ème} chapitre.

CHAPITRE 0x02

CONCEPTION D'UN ROBOT AUTONOME MOBILE



CHAPITRE II. Conception d'un robot autonome mobile :

II.1. Introduction :

Après avoir étudié l'état de l'art sur les robots mobiles il est temps d'en créer un. Dans ce chapitre nous allons entamer la partie conception d'un robot autonome mobile. Pour cela on a besoin tous d'abord de choisir la structure du robot, et ces différents éléments.

Afin de pouvoir se localiser dans un environnement, un robot mobile a besoin d'un moyen de perception (capteurs), qui informent le robot sur sa position absolue dans l'espace.

Et pour se déplacer dans un environnement donné le robot a besoin d'avoir un moyen de locomotion.

Et pour terminer, le robot a besoin d'un élément essentiel qui représente le cerveau du robot. Un composant qui sert à orchestrer les différents éléments constituent le robot, de l'acquisition des données qui parviennent des capteurs à la prise de décision, et ensuite l'exécution des différentes tâches de déplacement qui permettent au robot de naviguer dans son environnement de façon complètement autonome.

II.2. Structure du robot à concevoir :

Dans ce travail nous avons choisi de concevoir un robot autonome mobile à roues, de type tricycle, ce type de structure assure un déplacement aisé dans un environnement contrôlé (surface plane sans trous, non incliné, espace fermé protégé des éléments, ...etc.).

Comme moyen de locomotion on a utilisé des motoréducteurs qui sont en fait des moteurs à courant continu attaché à un réducteur afin d'avoir un meilleur rapport entre la vitesse de rotation et le couple, pour mieux contrôler le mouvement du robot, car avant tout, avoir une bonne précision est vitale pour contourner les obstacles rencontrés par le robot. Pour commander ces moteurs on a utilisé un pont H, qui est en fait un jeu de quatre transistors montés sous une configuration permettant de contrôler du sens de circulation du courant qui traverse le moteur ainsi contrôler le sens de rotation du moteur.

Pour contrôler la direction vers laquelle le robot se dirige on s'en sert de deux moteurs contrôlés individuellement pour tourner dans une direction donnée (voir Figure 23). Si les deux moteurs tournent dans le sens des aiguilles d'une montre le robot avance, et s'ils tournent dans le sens inverse des aiguilles d'une montre le robot recule. Pour tourner vers la droite ou vers la gauche les deux moteurs doivent tourner en sens opposés.

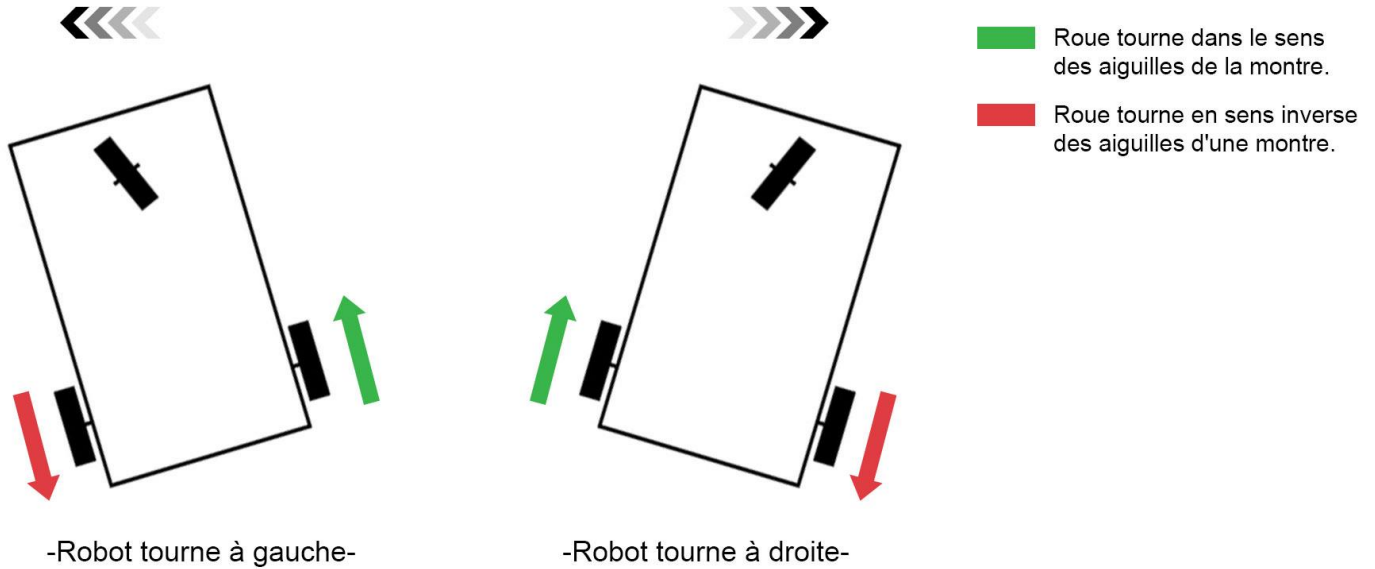


Figure 23: contrôle de la direction de déplacement du robot

En ce qui concerne la perception, un capteur ultrasonore est utilisé, positionné à l'avant du robot, il permet au robot de détecter les obstacles en utilisant des ondes ultrasonores. Les données du capteur sont traitées par une carte Arduino UNO (carte de développement basée sur le microcontrôleur ATmega328).

II.3. Le capteur ultrasonore Parallax 28015 :

Le capteur ultrasonore de Parallax 28015 est un moyen précis pour la mesure de la distance (2 cm à 3 m). Il est très utilisé en robotique pour son efficacité et sa facilité d'interaction avec toute sorte de microcontrôleurs (Arduino, Microchip PIC, Basic Stamp, Propeller chip,...etc.).

II.3.1. Caractéristiques et spécifications :

Ce capteur présente les caractéristiques et spécifications suivantes :

- Portée : de 2cm à 3m.
- Ayant une LED indicatrice de présence d'activité.

- Compatible avec les deux technologies de microcontrôleurs TTL5V et 3.3V.
- Impulsion d'entrée (trigger Input) : impulsion positive de $2\mu\text{s}$ minimum ($5\mu\text{s}$ typiquement).
- Impulsion de retour (impulsion d'écho) : impulsion positive de $115\mu\text{s}$ au minimum, et 18.5ms au maximum.
- Tension d'alimentation : 5VDC.
- Courant d'alimentation : 30mA typiquement, à 35mA au maximum.
- Plage de température : 0-70°C.
- Poids : 9g.

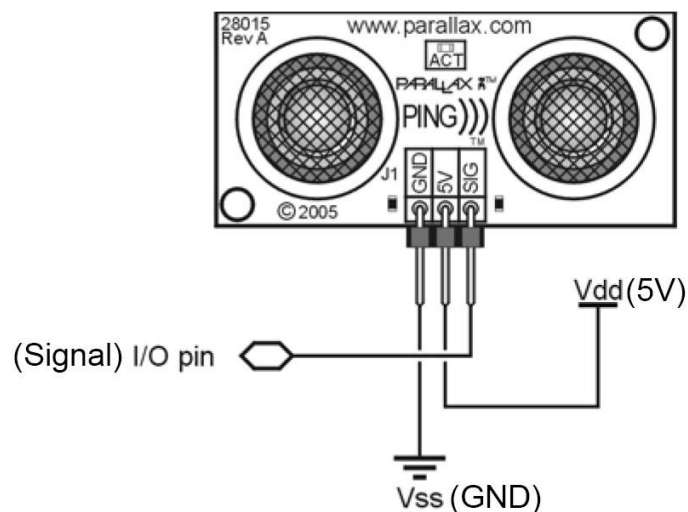


Figure 24: Brochage du capteur Parallax 28015

II.3.2. Principe de fonctionnement :

Son principe de fonctionnement c'est qu'il émet une courte onde ultrasonore (sous forme de burst), et puis il écoute. Avec une pulsation de gâchette (trigger pulse), déclenchée par le microcontrôleur qui le control, le capteur émet une courte onde (burst) de 40KHz (ultrasonique) voir la Figure 26, cette onde ce propage dans l'air, et puis heurtent un obstacle et revient vers le capteur comme illustré dans la figure suivante :

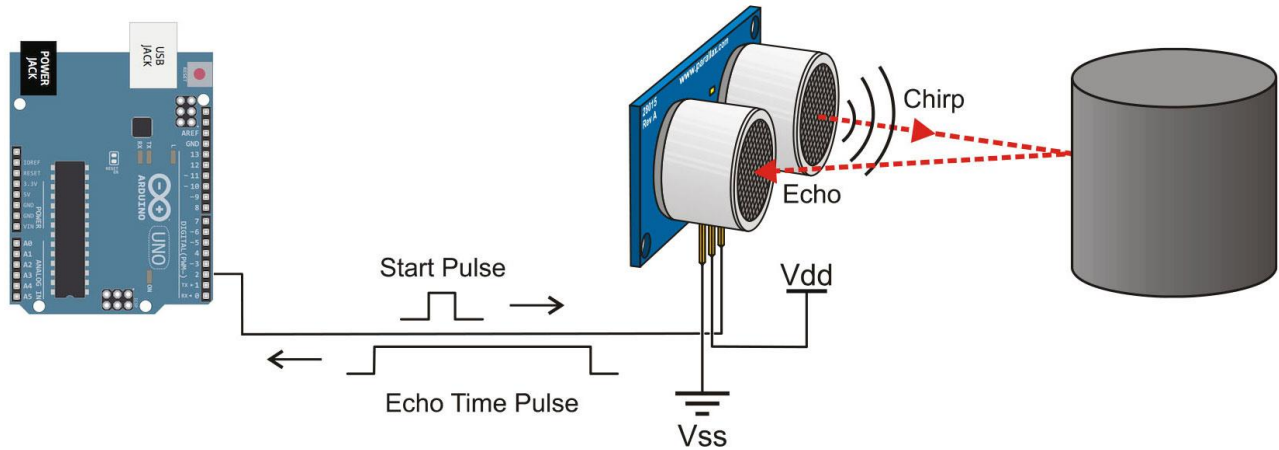
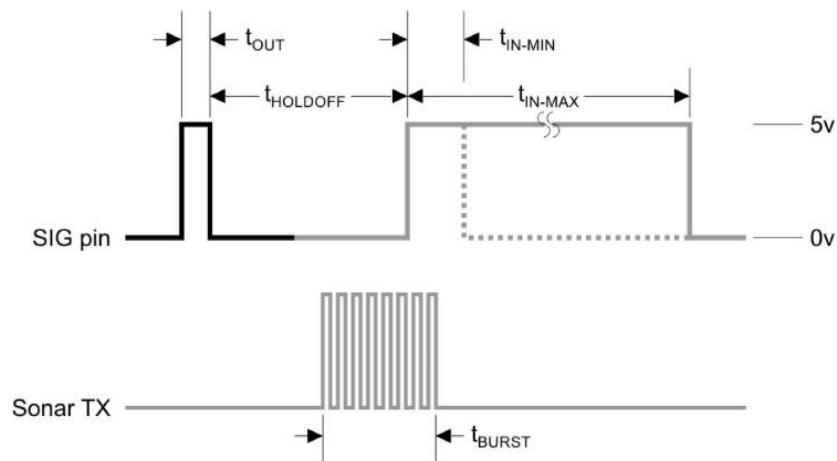


Figure 25: Principe de fonctionnement du capteur ultrasonore Parallax 28015



—	Host Device	Input Trigger Pulse	t_{OUT}	2 μ s (min), 5 μ s typical
—	PING))) Sensor	Echo Holdoff	$t_{HOLDOFF}$	750 μ s
—		Burst Frequency	t_{BURST}	200 μ s @ 40 kHz
—		Echo Return Pulse Minimum	t_{IN-MIN}	115 μ s
—		Echo Return Pulse Maximum	t_{IN-MAX}	18.5 ms
—		Delay before next measurement		200 μ s

Figure 26: Protocole de communication du capteur Parallax 28015

II.3.3. Considérations pratique :

En pratique pour assurer le bon fonctionnement du capteur il faut Respecter les conditions suivantes :

- a. éloignement des objets ne doit pas dépasser 3m.

- b. Il faut prendre en considération que le capteur ne peut pas détecter les objets qui sont à des angles raides.
- c. Si un objet est très petit au point de ne pas refléter les ondes incidentes, alors il est indétectable par le capteur.

Ces trois cas sont illustrés sur la figure suivante :

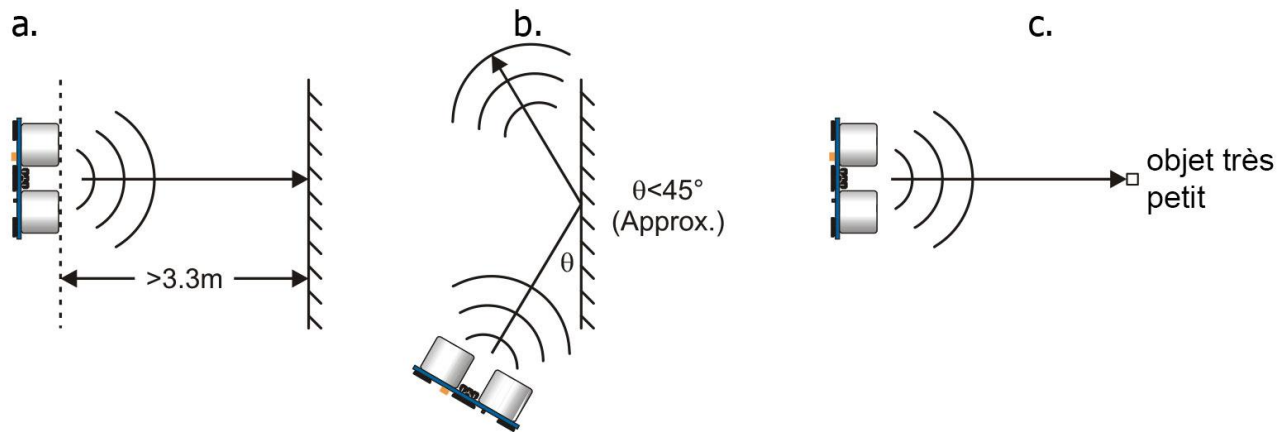


Figure 27: Considérations pratiques (limite du capteur ultrason)

- d. Un objet ayant une surface molle ou irrégulière peut ne pas être détecté (à noter que la surface de l'eau est détectable donc faire attention à ne pas utiliser dans les environnements humides).
- e. 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 comme suite :

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

Avec :

- o C_{air} : Vitesse du son dans l'air.
 - o T_c : Température de l'air.
- f. Seuls les objets perpendiculaires à un rayon émis sont détectables. L'onde émise est typiquement entre 20 à 30 degrés de cône d'ouverture. Pour exemple la Figure 28 ci-dessous montre les résultats d'un test fait par Prallax lab.

Capteur élevé du sol de 101.6 cm.

Obstacle cylindrique de diamètre de 8.9 cm, et de hauteur de 121.9 cm - orientation verticale.

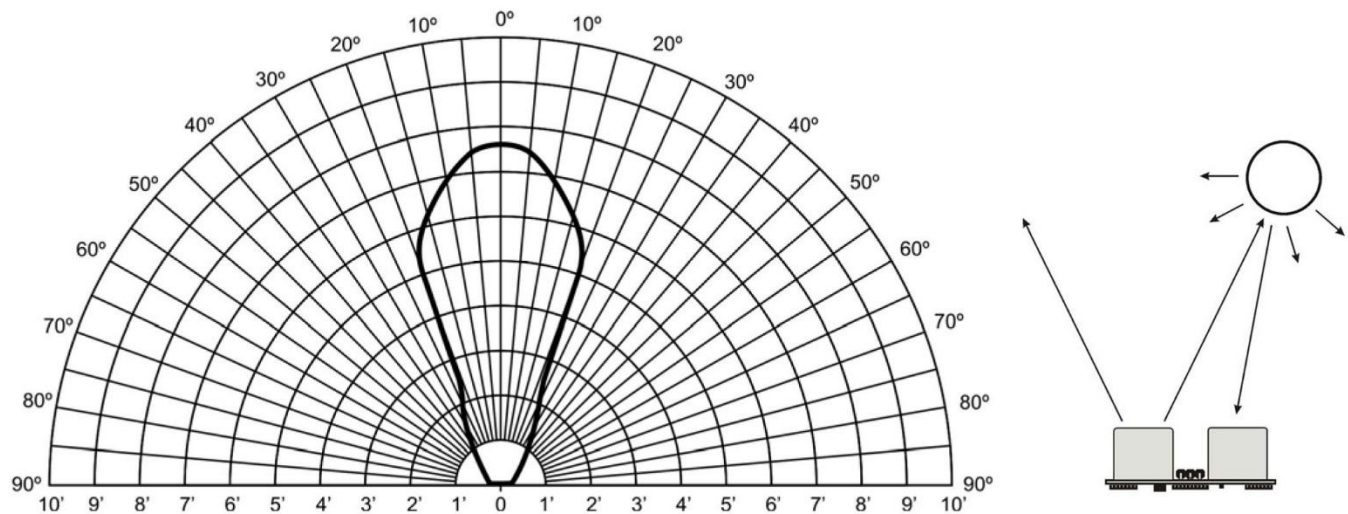


Figure 28: détection d'un objet cylindrique (et représentation du cône d'ouverture)

II.4. Les motoréducteurs :

Un motoréducteur est un moteur à courant continu (appelé aussi machine à courant continu) attaché à un réducteur, deux notions que nous allons aborder dans les paragraphes ci-dessous. Les motoréducteurs sont les actionneurs les plus utilisés en robotique mobile pour plusieurs raisons que nous allons voir tous de suite.

Alors pour commencer, la question à se poser est : c'est quoi un moteur à courant continu ?

II.4.1. Les moteurs à courant continu :

Tout 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. À 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 point 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 parviennent de cette conversion.

Il existe deux types de moteurs à courant continu :

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

a. Les moteurs à courant continu avec balais :

C'est ce genre de moteurs qu'ils ont utilisés dans les motoréducteurs utilisé dans ce projet, vue leur faible coût, les moteurs à courant continu (sans balais) sont utilisé lorsque une grande vitesse de rotation est désiré (par exemple pour faire tourner les hélices de drones) même si ils coût chère.

Les moteurs à courant continu sont composés de deux parties principales (voir Figure 29) :

- **Le rotor (induit):** c'est la partie qui tourne.
- **Le stator (inducteur):** c'est la partie statique (la partie fixe), il peut être un aimant permanent ou un électroaimant (bobines).

Le principe de fonctionnement des moteurs à courant continu est que le bobinage du moteur est disposé sur le rotor. Ce bobinage est placé dans un champ magnétique, permanent ou non, produit par le stator. On supposera pour simplifier que cette excitation est séparée et constante, comme c'est le cas notamment lorsque le stator est constitué d'aimants. Le courant circulant dans les spires de l'induit du moteur, des forces électriques lui sont appliquées et, grâce un dispositif adapté (balais et collecteur, visibles sur la Figure 29), les forces s'additionnent pour participer à la rotation. Ce principe de commutation mécanique du courant dans les spires du moteur n'est pas sans incidence sur le fonctionnement du moteur. Il en limite tout d'abord la vitesse maximale. Par ailleurs, les balais qui appuient sur le collecteur grâce à un système de ressorts sont en graphite pour permettre le passage du courant (et une usure maîtrisée). Le collecteur, qui est lui en cuivre, ne s'use pratiquement pas, ce qui est souhaitable puisque tout le bobinage est soude sur ce collecteur. Le moteur à courant continu avec balais nécessite donc une maintenance régulière. Par ailleurs, le passage du courant à travers le contact glissant

provoque parfois des étincelles, ce qui limite l'usage de ces moteurs dans certains environnements, explosifs notamment.

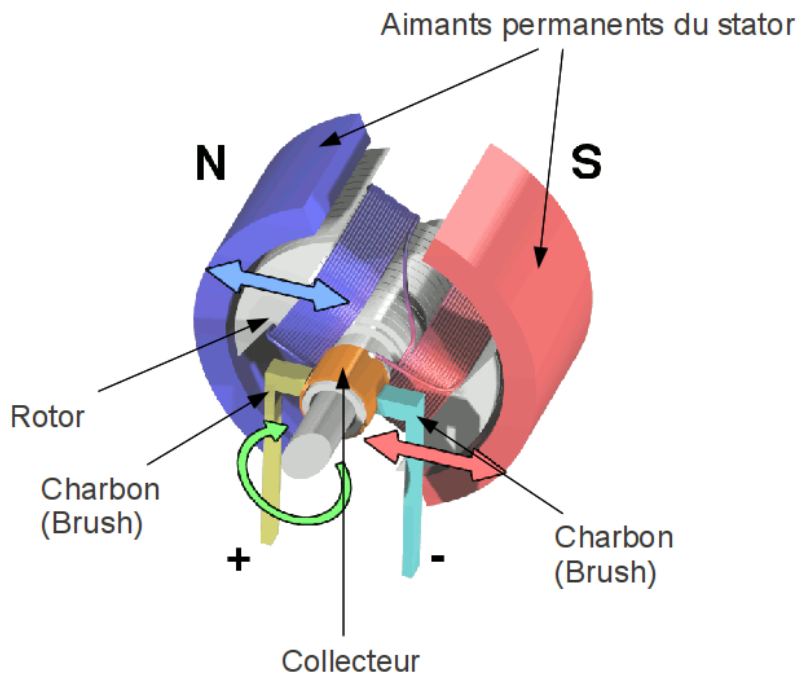


Figure 29: Eclaté d'un moteur à courant continu

b. Les moteurs à courant continu sans balais :

L'appellation de ces moteurs (héritée de l'anglais *DC brushless motors*, voir Figure 30) est trompeuse. Il ne s'agit en effet pas de moteurs à courant continu, mais de moteurs à courant alternatif : des moteurs synchrones autopilotés. Ils furent développés et commercialisés à partir de 1962, notamment pour pallier les 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 des propriétés mécaniques meilleures. Ils offrent notamment de meilleurs couples massiques et des vitesses de rotation maximales plus grandes. Leur fonctionnement, sans frottements entre rotor et stator, leur confère une plus grande fiabilité et un temps de vie supérieur. Ils occasionnent moins de bruit de commutation 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 (autre le prix), comme des effets d'ondulation de couple aux basses vitesses (cogging).



Figure 30: EMAX Brushless motor

Ces meilleures caractéristiques sont en revanche obtenues au prix d'un fonctionnement plus délicat à modéliser, et d'une électronique de commande plus complexe et plus coûteuse. Un moteur à courant continu sans balais est généralement équipé au rotor d'aimants placés dans le champ tournant produit par le bobinage triphasé du stator. De manière très grossière, on peut dire que le dispositif électronique de commande du moteur sans balais se substitue au convertisseur électromécanique constitué de l'ensemble balais et collecteur du moteur à courant continu. Pour cela, il est nécessaire de disposer de capteur à effet Hall pour repérer la position du rotor afin de réaliser l'auto-pilotage du moteur.

II.4.2. Les Réducteurs :

Les réducteurs consistent en un ensemble de roues engrenant les unes sur les autres, grâce au profil de leurs dentures qui peuvent être droites ou hélicoïdales (voir Figure 31). Ces réducteurs offrent un rapport de réduction égal au rapport du nombre de dents des roues dentées. La plus petite des roues, nommée pignon, montée sur l'arbre du moteur, tourne plus vite que la plus grande (nommée simplement roue), comme on le devine pour le réducteur de droite. Il est fréquent de préférer les engrenages à denture hélicoïdale aux engrenages à denture droite pour des raisons de bruit et de contraintes mécaniques sur les dentures.



Figure 31: Réducteurs Emerson à dentures droite (gauche) et hélicoïdale (droite).

Bien que l'on puisse utiliser des roues possédant des diamètres très différents 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 (voir Figure 32). On peut ainsi retrouver une sortie concentrique, ce qui est aussi un avantage.



Figure 32: Réducteurs Maxon compacts à dentures droite.

Le but de rajouter un réducteur pour un moteur à courant continu est d'avoir un couple plus élevé. Qu'est-ce que cela veut dire ?

Le couple est exprimé en Newton-Mètre. Ce qui veut dire que le couple est lié à la fois à une distance (Mètre) et à une force (Newton). Ce qui implique que le couple est la capacité du moteur à faire tourner quelque chose sur son axe. Donc plus le couple est élevé plus le moteur sera capable de supporter une charge plus lourde. Ainsi l'addition de réducteurs aux moteurs à courant continu nous permet aux motoréducteurs de supporter le poids de l'ensemble du robot, mais aussi d'avoir une meilleure précision car la vitesse est réduite, ceci est très important en robotique mobile car on a besoin d'avoir des moteurs

qui peuvent supporter le poids de du robot et d'avoir une certaine précision qui permet au robot d'exécuter certaines instructions dans les délais afin d'éviter les obstacles.

Le couple est calculé de la manière suivante :

$$\mathbf{C} = \mathbf{F} \times \mathbf{r} \quad (2.2)$$

Avec :

C : le couple, en Newton-Mètre.

F : la force exercée, en Newton.

r : le rayon de l'action, en Mètre.

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

$$R = \frac{V_e}{V_s} = \frac{C_s}{C_e} \quad (2.3)$$

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 exercée par l'axe en entrée du réducteur.

C_s : couple exercée 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 :

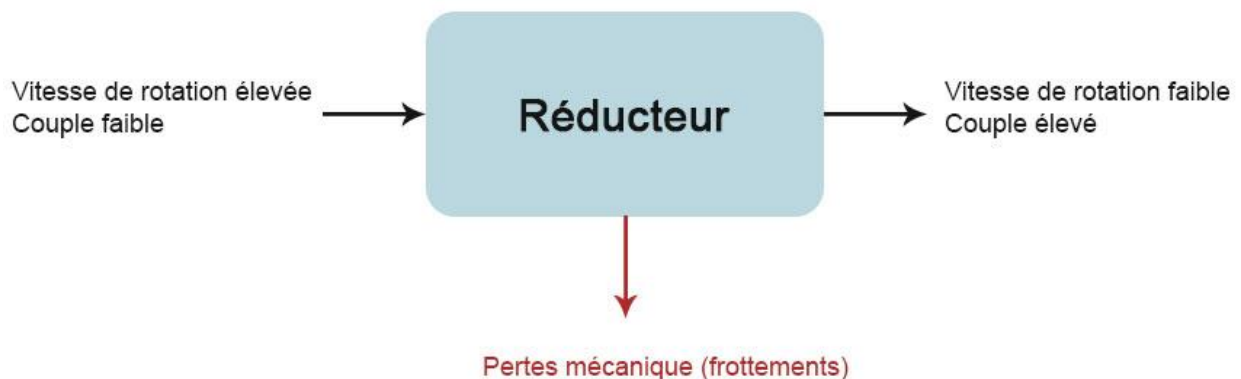


Figure 33: Schéma de fonctionnement d'un réducteur

II.4.3. Commande des moteurs à courant continu :

Pour faire tourner un moteur à courant continu le montage est simple, on le schématise comme suit :

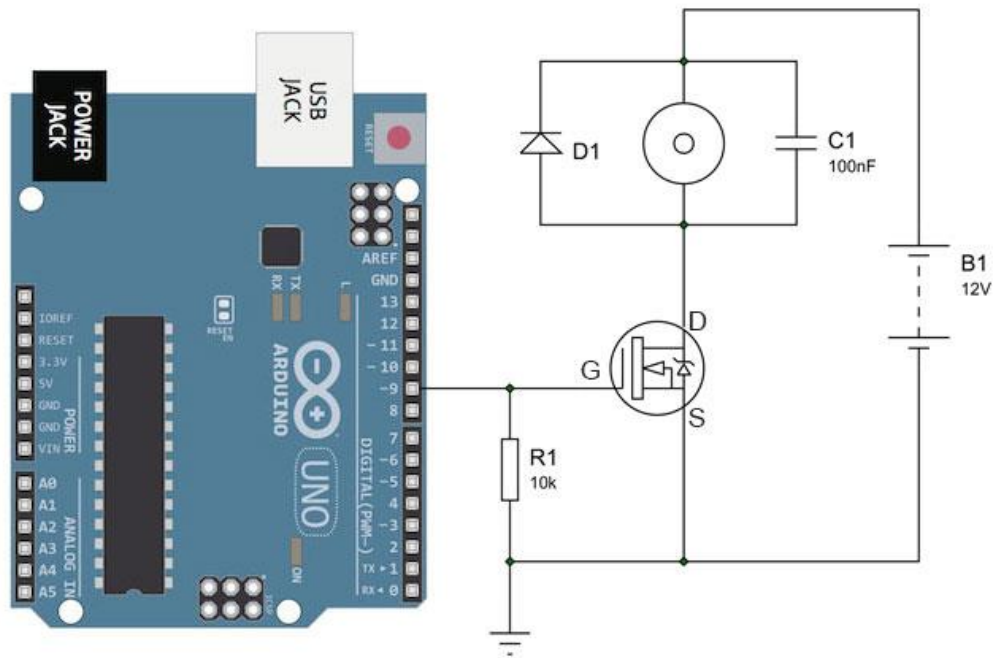


Figure 34: montage de commande d'un moteur pas à pas

Comme vous pouvez le voir on ne branche pas directement le moteur à la carte arduino (ici j'ai choisi d'utiliser la carte arduino, mais vous pouvez utiliser n'importe quel 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 endommager la carte arduino. Quand les moteurs à courant continu tournent, ils génèrent tout 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 assez fort pour commander le moteur, on a utilisé des transistors de puissance, dans notre schéma on a utilisé un transistor de type MOSFET optimisé à fonctionner à des régimes de fort courant. Leur fonctionnement est simple : une différence de potentiel sur la broche « gate » et ils commutent (ils laissent passer le courant entre le drain (D) et la source (S) sinon ils sont bloqués. Dans le schéma de la Figure 34 on a utilisé la 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 le moteur ne va pas tourner par défaut.

Pour remédier au problème de parasites on utilise un condensateur monté en parallèle avec le moteur et une diode de roue libre.

Le montage utilisé jusqu'à maintenant nous permet seulement de faire tourner le moteur dans un seul sens pour changer de sens on a besoin d'un moyen d'inverser 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 le monde de la robotique c'est ce qu'on appelle **le pont H** (H Bridge en anglais).

Pour pouvoir aller dans les deux sens il nous faut donc un dispositif qui serait capable de faire passer le courant dans le moteur dans un sens ou dans l'autre. Pour cela on utilise le pont H représenté dans la Figure 35 bien évidemment on utilise des transistors de puissance à la place des interrupteurs.

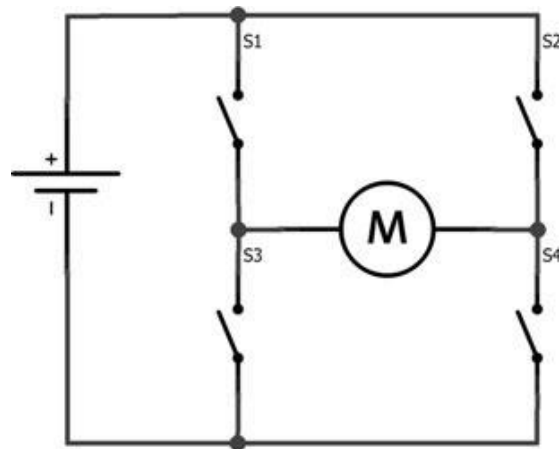


Figure 35: pont H

En remplaçant les interrupteurs par des transistors et en complétant le schéma on obtient le montage de la figure suivante :

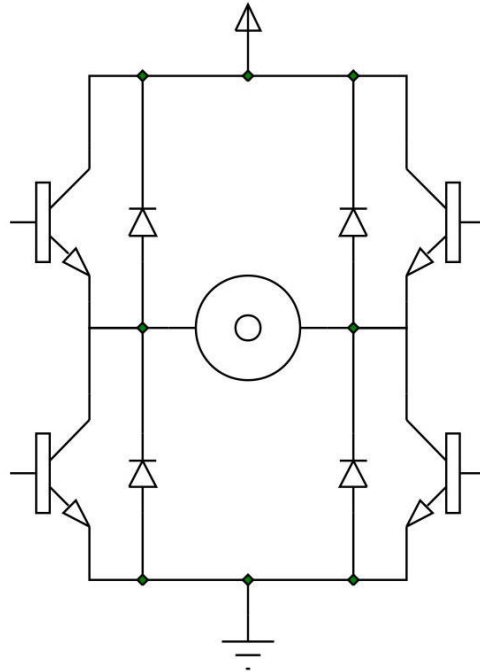


Figure 36: montage en pont H

II.5. Les servomoteurs :

II.5.1. Définition et principe de fonctionnement :

Les servomoteurs sont des moteurs particuliers, car leur rôle n'est pas seulement de tourner mais de maintenir une position donnée. Le mouvement de sortie est une rotation qui entraîne un changement de position de l'axe du servomoteur. Comme on peut le voir sur la Figure 38, ils intègrent dans un même boîtier un moteur à courant continu jumelé avec un train d'engrenages (en plastique ou en métal) démultipliant (réducteur), qui entraîne un axe avec une grande force de torsion mais une vitesse réduite (la vitesse est inversement proportionnelle au couple), et un capteur de retour de position (habituellement un potentiomètre), ainsi qu'un dispositif électronique de commande qui corrige la position la position de l'axe pour qu'elle soit égale à la consigne émise par le microcontrôleur (voir Figure 37), car les servomoteurs sont des systèmes asservis.

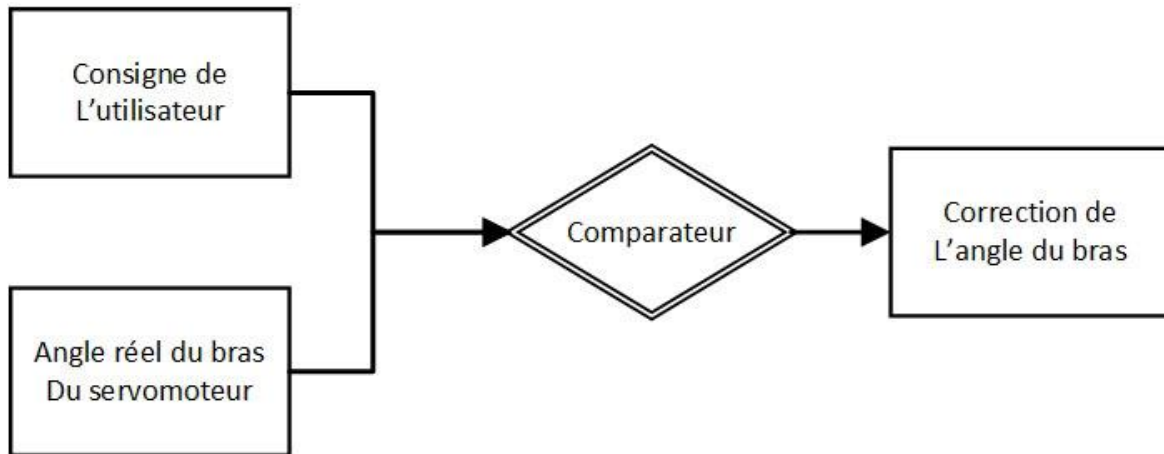


Figure 37: Correction de la position d'un servomoteur.

Généralement la rotation de ce type de moteurs est limitée à 180° par un mécanisme interne. Cela dit, il existe aussi des servomoteurs à rotation continue.

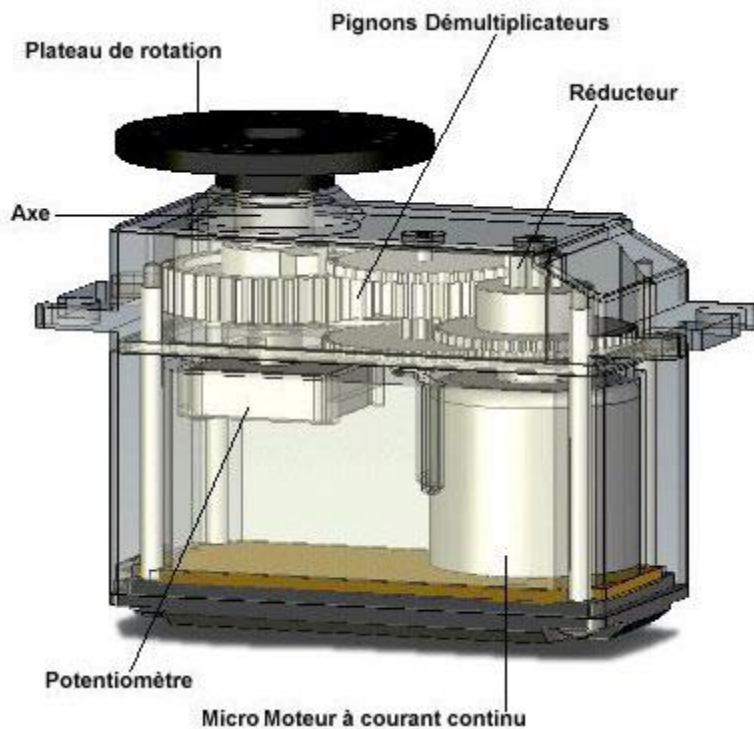


Figure 38: Vue de l'intérieur d'un servomoteur.

Les modèles diffèrent en taille, en puissance, en fiabilité et surtout en solidité des matériaux utilisés : plastique, laiton, acier. Certains sont équipés d'un ou plusieurs roulements à bille pour résister aux

pressions transverses sur l'axe moteur. Il existe une grande variété et qualité de servomoteurs se différenciant par leur vitesse de rotation/seconde, par leurs couples exprimés en kg.cm.

Sur le datasheet de chaque servomoteur on mention le couple, par exemple si on dit 3.2Kg.cm cela veut dire qu'au bout du bras du servomoteur s'il fait 1cm, il pourra soulever un poids de 3.2Kg. Après selon la taille du bras la charge maximale que peut supporter le servomoteur varie et peut se calculer avec la formule suivante :

$$P_{\max} = \frac{C}{D} \quad (2.4)$$

Avec :

P_{\max} : poids maximal de charge en kilogramme (kg).

C : couple du servomoteur, en kilogramme centimètre (kg.cm).

D : distance à laquelle le poids est placé en centimètre (cm).

Donc comme on peut le voir, plus la distance est grande plus le poids maximal que le servomoteur peut supporter est petit, voir la figure suivante.

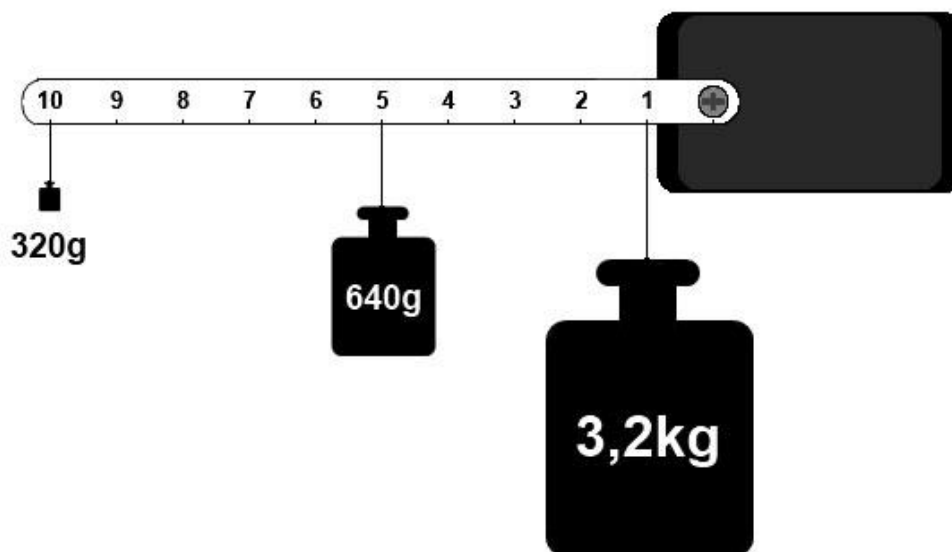


Figure 39: la variation du poids maximale que peut supporter un servomoteur selon la distance.

Les servomoteurs ont besoin de trois fils de connexion pour fonctionner. Deux fils servent pour l'alimentation, et un fil pour recevoir le signal de commande. On reconnaît ces fils par un code de couleur qui est comme suit :

- Rouge : pour l'alimentation positive (4.5V à 6V en général).
- Noire ou Marron : pour la masse (GND).
- Orange, jaune ou blanc : pour le signal de commande.

II.5.2. Commande des servomoteurs :

La consigne envoyée au servomoteur n'est autre qu'un signal électronique *codé en largeur d'impulsion* plus communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur.

Le signal est répété périodiquement toutes les 20 ms (fréquence de 50 Hz). La Figure 40 explique plus en détail comment ce fait la commande.

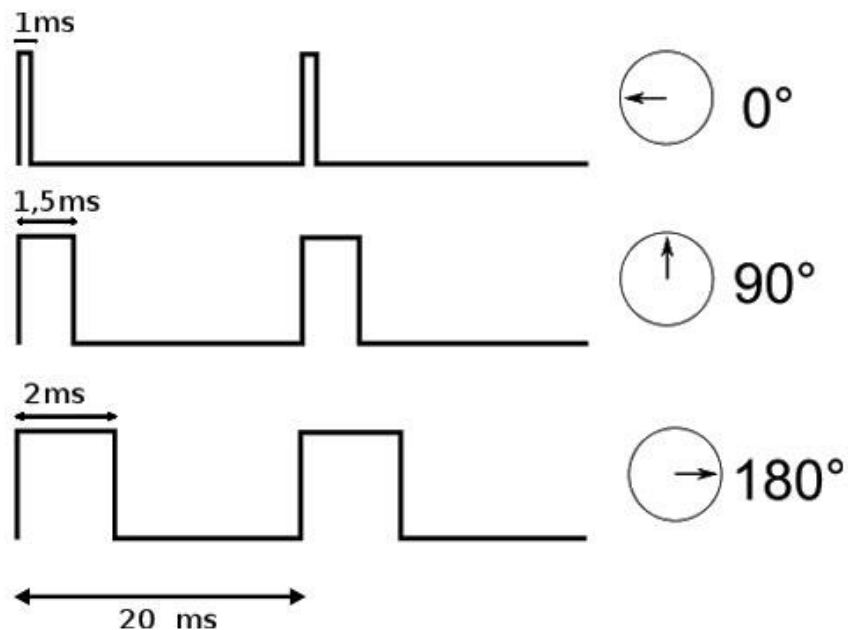


Figure 40: Commande de la position d'un servomoteur.

Note :

Il existe un autre type de servomoteur qui peut utiliser car ils sont très chers, et ils utilisent parfois des protocoles bien particuliers rendant leurs commandes plus difficiles.

II.6. Téléguidage par Bluetooth :

Afin de pouvoir télécommander à distance notre robot, on doit disposer d'un moyen de communication bidirectionnel et sans filaire à de faible consommation d'énergie, facile à mettre en œuvre et à utiliser par l'utilisateur et peu coûteux. C'est pour cela qu'on a choisi dans ce travail la technologie Bluetooth qui présente toutes ces caractéristiques.

II.6.1. La technologie Bluetooth :

Bluetooth est un standard de communication permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio UHF (2.4GHz). Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires. En effet Bluetooth est une technologie de réseau personnel sans fils noté WPAN (Wireless Personal Area Network) qui ne nécessite pas une ligne de vue directe pour communiquer.

La technologie **Bluetooth** a été originairement mise au point par *Ericsson* en 1994. En février 1998 un groupe d'intérêt baptisé *Bluetooth Special Interest Group (Bluetooth SIG)*, réunissant plus de 2000 entreprises dont Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia et Toshiba, a été formé afin de produire les spécifications Bluetooth 1.0, qui furent publiées en juillet 1999. Le nom « Bluetooth » (littéralement « dent bleue ») se rapporte au nom du roi danois Harald II (910-986), surnommé Harald II Blâtand (« à la dent bleue »), à qui on attribue l'unification de la Suède et de la Norvège ainsi que l'introduction du christianisme dans les pays scandinaves.

La technologie Bluetooth présente les caractéristiques suivantes :

Le Bluetooth permet d'obtenir des débits de l'ordre de 1 Mbps, correspondant à 1600 échanges par seconde en full-duplex, avec une portée qui dépend de la classe de l'émetteur utilisé. Le standard Bluetooth définit 3 classes d'émetteurs qui sont représenté dans le tableau suivant :

Classe	Puissance (affaiblissement)	Portée
1	100 mW (20 dBm)	100 mètres
2	2,5 mW (4 dBm)	15-20 mètres
3	1 mW (0 dBm)	10 mètres

Le standard Bluetooth se décompose en différentes normes :

- Bluetooth v1.0 et v1.0B ;
- Bluetooth v1.1, normalisé en 2002 sous le nom IEEE 802.15.1-2002 ;
- Bluetooth v1.2, normalisé en 2005 sous le nom IEEE 802.15.1-2005 ;
- Bluetooth v2.0 + EDR, rendue publique en 2004 ;
- Bluetooth v2.1 + EDR, rendue publique en 2007 ;
- Bluetooth v3.0 + HS, rendue publique en 2009 ;
- Bluetooth v4.0, rendue publique en 2010 ;
- Bluetooth v4.1, rendue publique en 2013.
- Bluetooth v4.2, rendue publique le 2 Décembre 2014 (mise à jour matérielle).

Les éléments fondamentaux d'un produit Bluetooth sont définis dans les deux premières couches protocolaires :

1. la couche *radio*,
2. et la couche *bande de base*.

Afin d'échanger des données, les appareils doivent être appairés. L'appairage se fait en lançant la découverte à partir d'un appareil et en échangeant un code. Dans certains cas, le code est libre, et il suffit aux deux appareils de saisir le même code. Dans d'autres cas, le code est fixé par l'un des deux appareils (appareil dépourvu de clavier, par exemple), et l'autre doit le connaître pour s'y raccorder. Par la suite, les codes sont mémorisés.

II.6.2. Le module HC-05 :

Le module HC-05 (voir la Figure 41) nous permet d'avoir une liaison Bluetooth SPP (Serial Port Protocol), qui nous permet d'envoyer et recevoir des données sous forme d'une communication série sans fil. Ce module utilise la norme Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps avec une modulation de 2.4GHz en bande de base, il utilise la puce CSR Bluecore 04-External en technologie CMOS et avec AFH (Adaptive Frequency Hopping Feature).

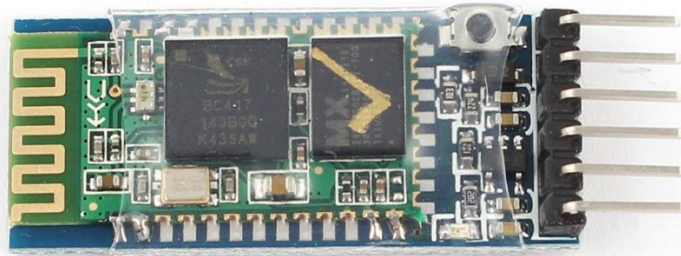


Figure 41: le module HC-05.

Spécifications matériel :

- Sensibilité de -80dBm.
- Puissance de transmission RF jusqu'à +4dBm.
- Opère à de faible puissance de 1.8V à 3.6V I/O.
- Interface UART (Universal Asynchronous Receiver Transmitter) avec débit (bauderate) programmable.
- Antenne intégré.
- Contrôle PIO.
- Avec connecteur de bord.

Spécification logiciel :

- Vitesse de transmission par défaut : 38400 bauds, les données sur 8bits, bit de stop sur 1bit, pas de bit de parité, contrôle de données, les vitesses de transmission supportée sont : 9600, 19200, 38400, 57600, 115200, 230400, 460800.
- Si un niveau haut est envoyé sur PIO0, l'appareil sera débranché.
- L'état des ports d'instruction PIO1 : niveau bas déconnecté, niveau haut connecté.

- PIO 10 et PIO 11 peuvent être connecté à la led rouge ou bleu séparément. Quand le maitre et l'esclave sont appairés (ou associés), ces deux leds clignote, par contre si ils sont déconnecté seul la led bleu qui clignote.
- Auto-connect au dernier appareil auquel il a été connecté dernièrement dès la mise en marche.
- Pin code par défaut est : 1234.

Le module fonctionne selon deux modes :

- Automatic connection : ce mode à trois rôles de fonctionnement (maitre, esclave, boucle), dans ce mode le module suit le chemin établie précédemment pour transmettre les données automatiquement.
- Order-responce : ce mode est utilisé pour envoyer les commandes AT au module afin de le configurer (comme par exemple changer le nom du module, changer la vitesse de transmission ...etc).

La sélection entre ces modes ce fait en contrôlant niveau logique de la broche PIO11.

II.7. La carte Arduino UNO :

La carte Arduino UNO (voir Figure 42) est un outil de développement crée a base du microcontrôleur ATmega328 d'Atmel, elle a été créé par un groupe de développeurs italien (Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, et David Mellisqui).



Figure 42: Carte Arduino Uno

Arduino a été créé en Italie à l'Interaction Design Institute Ivrea 2005 comme un moyen de prototypage, destiné aux étudiants en manque de notions d'électronique et de prototypage. Le nom « Arduino » 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, qui a été le roi d'Italie de 1002 à 1014 ap. J.-C.

dès qu'elle a atteint une plus grande communauté, la carte Arduino a commencé à changer et à s'adapter aux nouveaux besoins et défis imposés par cette dernière, la rendant de plus en plus à travers les années le cœur de milliers de projets, que ce soit pour des projets d'amateur ou pour des projets scientifiques complexes.

Maintenant la question qui se pose pourquoi avoir choisi la carte Arduino et pas une autre carte de développement, car il en existe plusieurs, tel les cartes de microchip à base de microcontrôleurs PIC, les cartes Parallax Basic Stamp, Phidgets, Netmedia's BX-24, ...etc. comme toutes ces cartes la carte Arduino Uno simplifie l'utilisation des microcontrôleurs et offre d'autres avantages à ne pas ignorer :

- Peu coûteuses : les cartes arduino sont relativement peu coûteuses (3500 DA à 4000 DA).
- Multi-plateforme : l'IDE Arduino qui sert à la programmation des cartes arduino marche pratiquement sur tous les systèmes d'exploitation que ce soit sur Windows ou Macintosh OSX ou linux, par contre les autres systèmes à microcontrôleurs ce limite généralement à Windows seulement.
- Open source software : la nature open source permet au 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 comment elle marche, et aussi de créer ce qu'on appelle des shield (ou boucliers 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 au langage rapide par la majorité des utilisateurs (car les langages C/C++ et java sont des langages de programmation très populaires).

- On peut programmer la carte Arduino directement par le port USB, sans programmeur externe. Le bootloader ne faisant que 0.5Ko, c'est 31.5 Ko de programme qui sont disponibles avec l'ATmega 328, ce qui est très largement suffisant dans la plupart des cas.

II.7.1. Caractéristiques :

La carte Arduino Uno est une carte de développement à base du microcontrôleur ATmega328, elle a 14 pins digitaux en entrée/sortie, dont 6 sont utiliser comme sorties PWM (Pulse width modulation ou en français modulation de largeur d'impulsion), un quartz à 16MHz, un connecteur USB pour communiquer avec un PC, un connecteur POWER JACK pour fournir une source de tension externe (7-12V), un connecteur mâle à six broches ICSP pour la programmation série de l'arduino (permet de se passer du port USB), et un bouton reset. Dans la Figure 43 on peut observer l'emplacement des différents éléments cités précédemment.

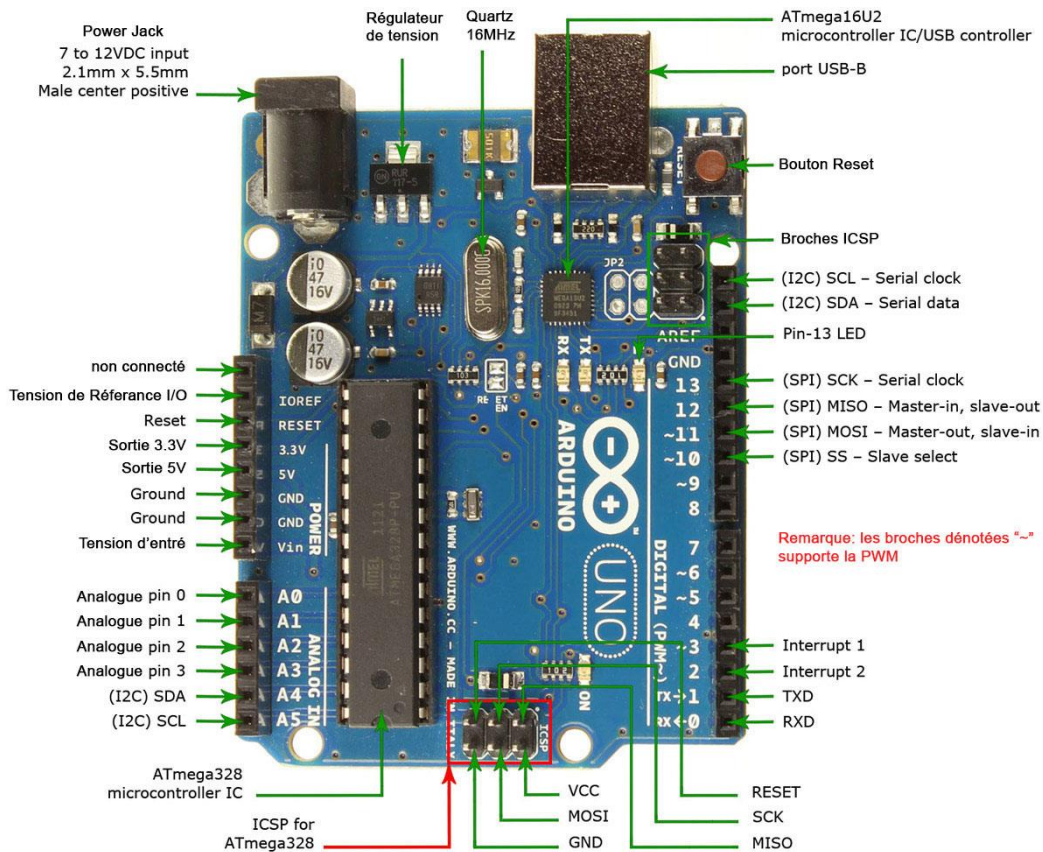


Figure 43: Schéma explicite de la carte Arduino Uno

Le tableau ci-dessus nous donne toutes les références hardware de la carte arduino uno :

Microcontrôleur	ATmega328
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
Courant par broche en 3.3V	50Ma
Mémoire Flash	32KB (dont 0.5KB sont utiliser pour le bootloader)
SRAM	2KB
EEPROM	1KB
Horloge	16MHz
Longueur	68.6 mm
Largeur	53.4 mm
Poids	25 g

II.7.2. Alimentation de la carte Arduino Uno :

La carte Arduino Uno est alimenter via le port USB ou par une source d'alimentaion externe, la source d'alimentation est selectionnée automatiquement.

En ce qui concerne la source externe il lui faut un adaptateur AD vers DC ou une battery.

La carte arduino opère avec une alimentation externe de 6 à 20 volts, mais si la tension fournit est inférieur à 7V la broche de 5V peut ne pas fournir les 5V ce qui va causer une instabilité de la carte. Et si on utilise plus de 12V, le régulateur de tension va ce chauffé rapidement ce qui va endommager la carte. C'est pour cela qu'il est recommandé d'opérer avec une tension entre 7 à 12V.

La carte arduino uno dispose de plusieurs broches d'alimentation qui sont :

- **VIN** : c'est la tension d'entrée de la carte quand 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. Si on alimente la carte via cette broche on contourne le régulateur de tension et cela peut endommager la carte.
- **3.3V** : fournit une tension de 3.3V comme son nom l'indique, elle épuise 50 mA. Comme la broche 5V cette broche contourne le régulateur de tension et cela peut endommager la carte.
- **GND**: broche de terre.
- **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.

II.7.3. Les Entrées/Sorties :

a. Numérique (digital) :

Chacune des 14 broches digital peuvent être utilisé en entrée ou en sortie, elles opèrent avec une tension de 5V et utilisent 40 mA et ils ont des résistances (20-50 K Ω) (pull-up resistors) qui sont déconnectées par défaut.

Certaines broches ont des fonctionnalités spéciales :

- **Communication Serial 0 (RX) et 1 (TX)**: utilisé pour recevoir (RX) et envoyer (TX) les données TTL série. Ces broches sont connectées aux broches correspondant sur la puce ATmega8U2USB vers TTL.
- **Interruptions externes 2 et 3**: ces broches peuvent être configuré comme source d'interruptions soit sur un niveau logique bas ou haut, ou à front montant ou descendant.
- **PWM 3, 5, 6, 9, 10 et 11** : ces broches fournissent l'option de fonctionnement en PWM (modulation de largeur d'impulsion).
- **SPI 10(SS), 11(MOSI), 12(MISO), 13 (SCK)**: Ces broches supportent la communication SPI.
- **LED 13** : il y a une led (Light Emetting Diode) incluse sur la carte, connectée à la broche 13.

b. Analogique :

La carte arduino uno dispose de 6 broches analogiques d'A0 à A5, chacune à une résolution de 10bits (1024 valeurs). Par défaut elles fonctionnent entre la masse (GND) et 5V, mais c'est possible de modifier cette marge en utilisant la broche AREF.

Comme les broches digitales, certaines de ces broches ont des fonctionnalités spécialisées :

- **TWI A4 (SDA), A5 (SCL):** ces broches utilisées pour la communication TWI (WIRE).
- **AREF :** sert comme référence pour les broches analogiques.
- **Reset :** activée au niveau bas, cette broche permet la réinitialisation du microcontrôleur.

II.7.4. La communication avec l'extérieur :

La carte arduino dispose de plusieurs moyens de communication que ce soit avec une autre carte, ou avec un ordinateur, ou un autre microcontrôleur. Le microcontrôleur ATmega328 fournit un moyen de communication série UART TTL (5V) sur les broches 0 (RX) et 1 (TX). Le circuit intégré ATmega16U2 permet d'avoir à travers le port USB un port virtuel COM pour le logiciel sur l'ordinateur. L'ATmega16U2 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 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 ATmega328 permet aussi d'utiliser la communication I2C (TWI) 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 avec les broches 10, 11, 12, 13 qui représente respectivement SS, MOSI, MISO, SCK de la communication SPI.

Bien évidemment pour chacune de c'est communication il y a une librairie qui permet l'interfaçage avec l'IDE Arduino. Pour la communication série on utilise la librairie SoftwareSerial [19], pour la communication I2C on utilise la librairie Wire [20], et pour la communication SPI on utilise la librairie SPI [21].

II.7.5. Le microcontrôleur ATmega328:

Le microcontrôleur ATmega328 est au 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 ATMEL de la famille AVR 8bits, la signification de ATmega328 est comme suite :

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

Tous les ports et broches (voir Figure 44) peuvent être configurés en entrée ou en sortie de manière individuelle. Certaines broches ont quelquefois une fonction secondaire, voire même un triple emploi. Pour utiliser les fonctions auxiliaires, il faudra toujours les configurer de manière logicielle, les fonctionnalités principales et auxiliaires des différents ports et broches sont :

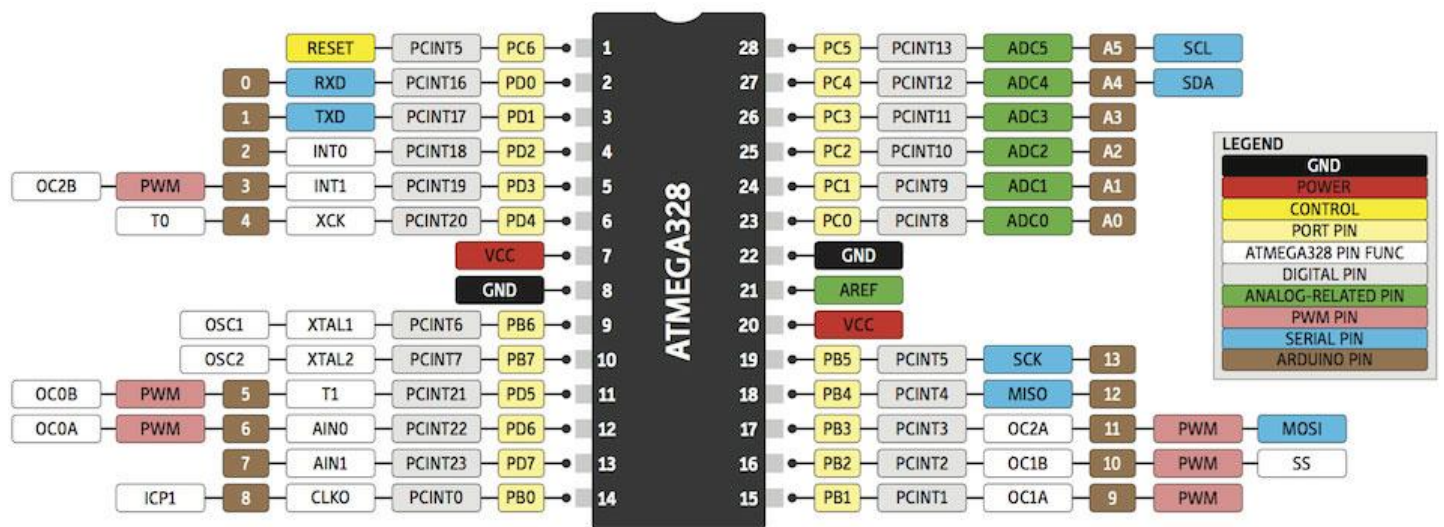


Figure 44: brochage du microcontrôleur ATmega328

- **Port A (PA7.. PA0)** le Port A est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi pour les entrées analogiques du convertisseur A/D. Le Port A (comme le B, C et D) est en position trois états quand une condition de reset devient active, même si l'horloge ne court pas.

- **Port B (PB7.. PB0)** le Port **B** est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (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 un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi comme oscillateur pour le Timer/Compteur2 et d'interface **I2C**.
- **Port D (PD7.. PDO)** le Port **D** est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi d'**USART** et d'entrées pour les interruptions externes.
- **RESET** déclenché par un front descendant maintenue plus de 50 ns il produira 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 analogue pour le Convertisseur **A/D** avec une tension dans la gamme de **2 V** à **AVCC** avec 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** broches d'alimentation du microcontrôleur (+3 à +5V).
- **GND** masse de l'alimentation.

Le schéma de l'architecture interne de l'ATmega328 (voir la Figure 45) nous permet de remarquer les fonctionnalités suivantes la:

- 2 Timer/Compteur 8 bits avec facteur de pré-division indépendant.
- 1 Timer/Compteur 16 bits avec facteur de pré-division indépendant.
- 1 Horloge Temps réel (RTC) avec quartz externe.
- 4 canaux PWM.

- 1 Convertisseur Analogique/Numérique 8 canaux avec une résolution de 10bits.
- 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.

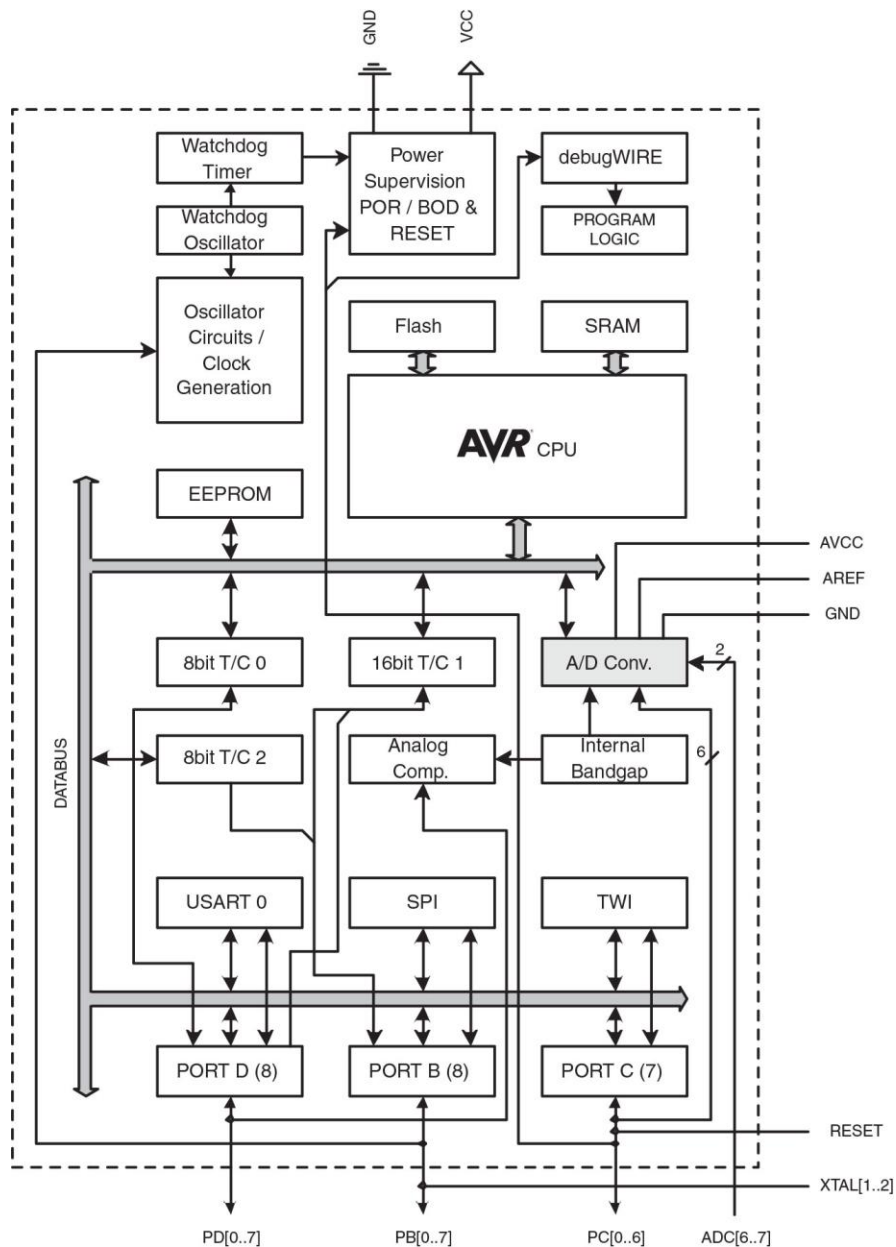


Figure 45: Schéma synoptique du microcontrôleur ATmega328

Comme on peut le voir, trois types 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, qui est largement suffisant pour la majorité des applications. Sa capacité est de 32Ko, une petite zone 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 divers opérations.
- **La mémoire de données SRAM** : la mémoire de données contient les 32 registres de travail, les 64 registres de commande (voir la Figure 46), et mémorise aussi toutes les différentes informations temporaires lors de l'exécution. Sa capacité est de 2Ko.
- **La mémoire morte EEPROM** : la mémoire morte de type EEPROM, l'accès est complexe, elle contiendra la configuration du programme et les données importantes qui seront sauvées en l'absence de courant.

Pour référence le cœur du microcontrôleur ATmega328 est un processeur AVR dont l'architecture interne est représentée dans la figure suivante :

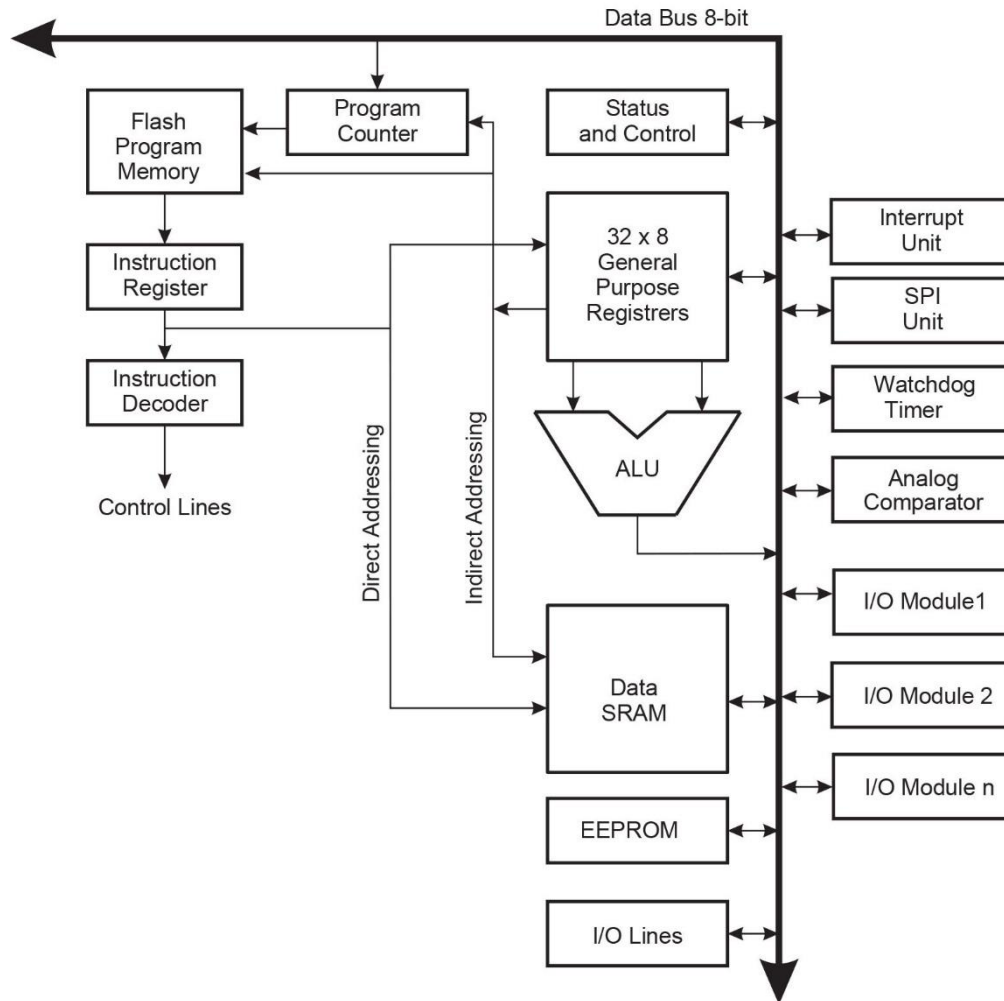


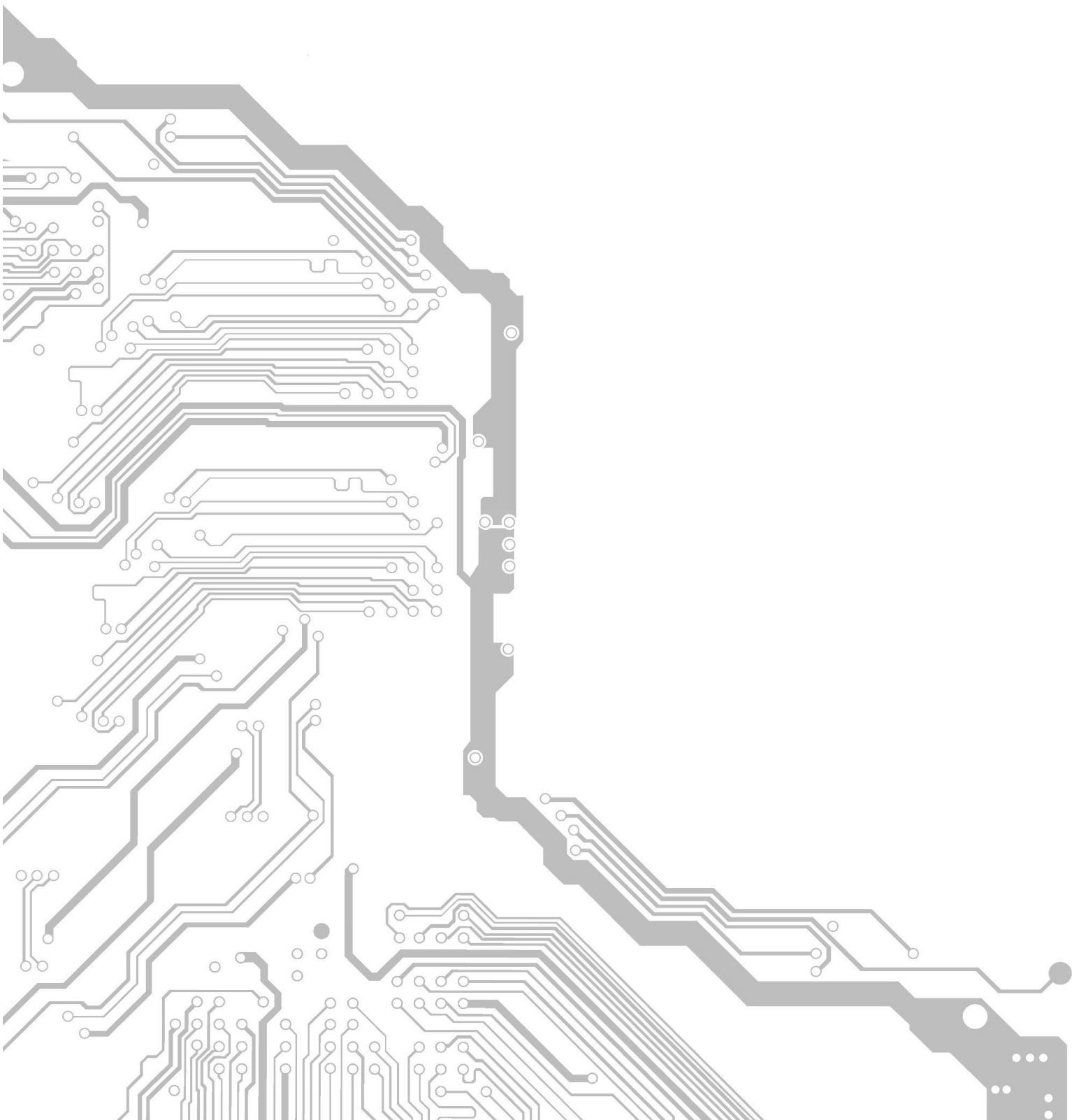
Figure 46: Schéma synoptique de l'architecture AVR

II.8. Conclusion :

Dans les sections précédentes, nous avons présenté l'approche suivie afin de concevoir un robot autonome mobile. Nous nous sommes focalisés en premier temps sur l'élaboration d'une structure à suivre pour la réalisation du robot. Une fois on a une structure à suivre on a entamé l'étude du principe de fonctionnement des différents éléments qui constituent le robot.

Après l'étude théorique le chapitre qui va suivre on va conclure le travail en passant à la réalisation du robot mobile.

REALISATION PRATIQUE



CHAPITRE III. Réalisation pratique :

III.1. Introduction :

Après avoir fait l'étude théorique on passe à la réalisation pratique. En pratique il y a plusieurs défis à relever, on commence par choisir le bon châssis pour un robot de type tricycle, puis choisir les différents organes du robot et leur placement optimal afin d'assurer un bon fonctionnement.

Dans le chapitre précédent on a établi une structure pour la conception du robot, maintenant il ne reste plus qu'à passer à la mise en œuvre. Pour cela on doit établir une méthode à suivre pour la réalisation matérielle et logicielle, faire les tests et interpréter les résultats.

III.2. Architecture du robot :

L'architecture du robot s'articule autour de la carte Arduino UNO qui assure l'interfaçage entre les différents organes du robot et l'acquisition de données via le capteur ultrasonore ainsi que la prise de décision la Figure 47 représente les différentes parties qui constituent le robot.

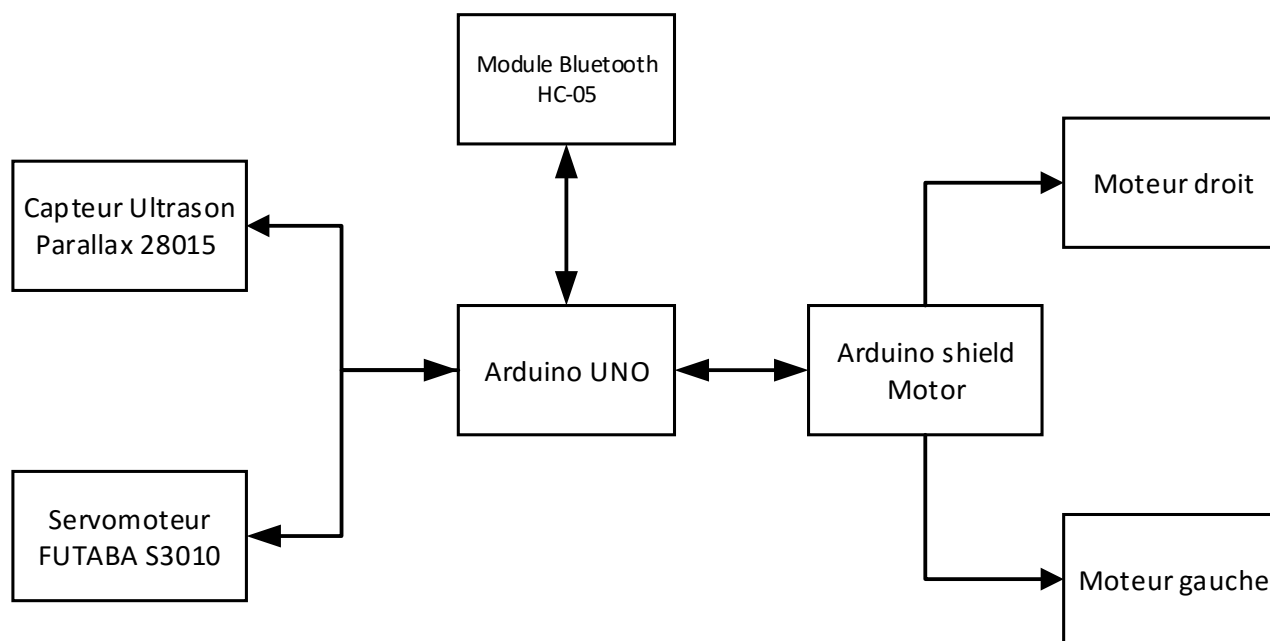


Figure 47: Schéma synoptique du robot.

III.3. Réalisation matérielle :

III.3.1. La perception :

La perception est atteinte grâce à un capteur ultrasonore Parallax 28015 dont le principe de fonctionnement a été traité dans les chapitres précédant. Attaché à un servomoteur (FUTABA S3010) comme en peut l'observer sur la **Figure 48**, Le capteur ultrasonore joue le rôle des « yeux » du robot lui permettant de visualiser son environnement (c.-à-d. détecté les obstacles via les ondes ultrason), et le servomoteur le « cou » du robot qui permet au robot de regarder dans toutes les directions sans avoir à tourner son corps.

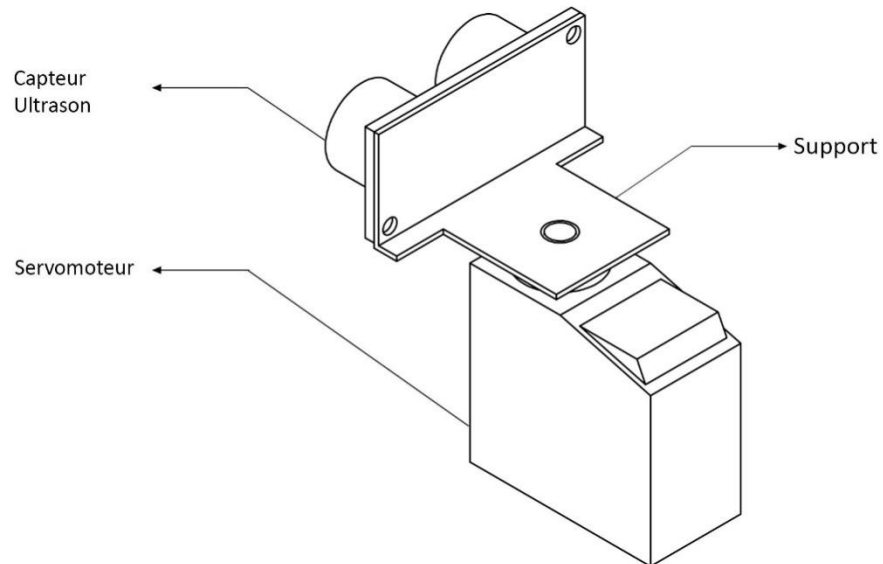


Figure 48: la partie responsable de la perception du robot.

Le servomoteur qu'on utilise dont la référence est FUTABA S3010 a les spécifications suivantes :

- Vitesse de 0.20 sec/60° à 4.8V.
- Vitesse de 0.16 sec/60° à 6V.
- Couple de 5.2Kg.cm à 4.8V.
- Couple de 6.5Kg.cm à 6V.
- Poids de 41g.
- Dimension de 40 x 20 x 38mm.

Son connecteur est représenté dans la figure suivante :

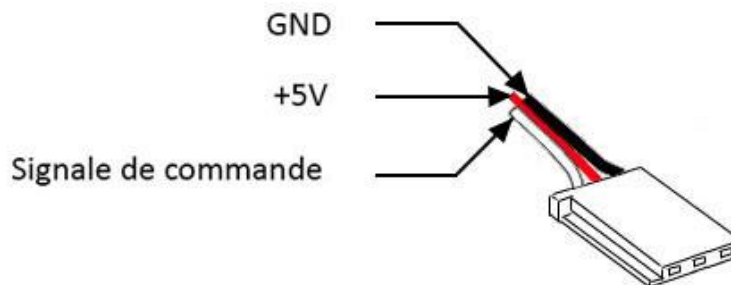


Figure 49: Connecteur du servomoteur.

Le robot support trois configurations possibles pour pouvoir regarder dans trois positions différentes (regarder à droite, à gauche et bien évidemment devant) comme en peut l'observer sur la figure suivante :

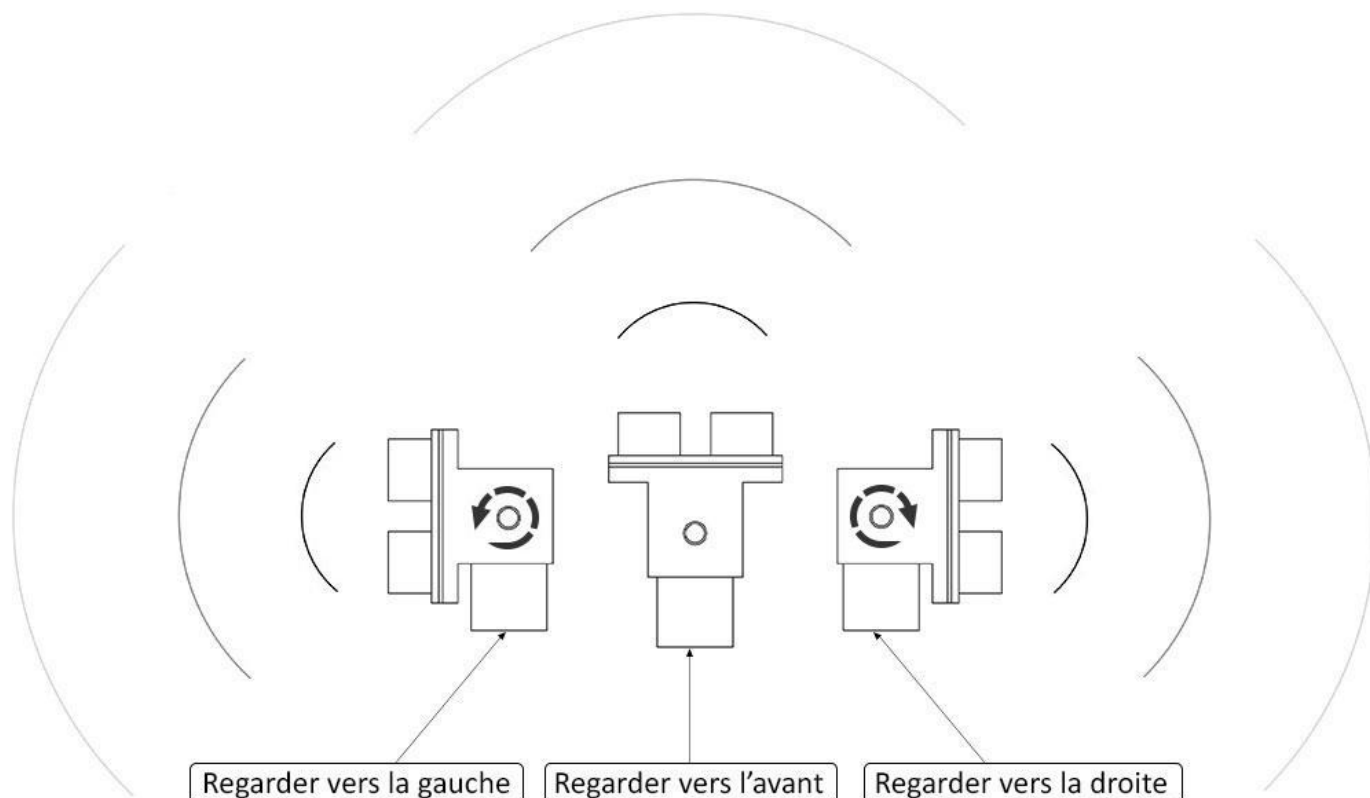


Figure 50: Vue de dessus de la partie perceptive du robot.

- Pour regarder vers l'avant on tourne le servomoteur vers la position 90° puis en mesure la distance avec le capteur ultrasonore.

- Pour regarder vers le côté gauche on tourne le servomoteur vers la position 180° puis en mesure la distance avec le capteur ultrason.
- Pour regarder vers le côté droit on tourne le servomoteur vers la position 0° puis en mesure la distance avec le capteur ultrason.

III.3.2. La locomotion :

a. Les motoréducteurs :

Les moteurs utilisés dans ce travail (voir Figure 51) dont la référence est **DG01D**, sont des motoréducteurs très populaires en robotique mobile. Ce type de moteur est caractérisé comme suit :

- Tension d'alimentation recommandée : 4.5V DC.
- Vitesse sans charge : 90 ± 10 rpm.
- Courant sans charge : 190 mA (250mA MAX).
- Couple maximum permet 800g.cm.
- Rapport de réduction de vitesse 48:1.



Figure 51: Motoréducteur DG01D

A chacun de ces moteurs on attache une roue avec pneu gomme de diamètre \varnothing 65mm. Les emplacements des moteurs et des-roues sur le châssis sont représentés sur la Figure 52.

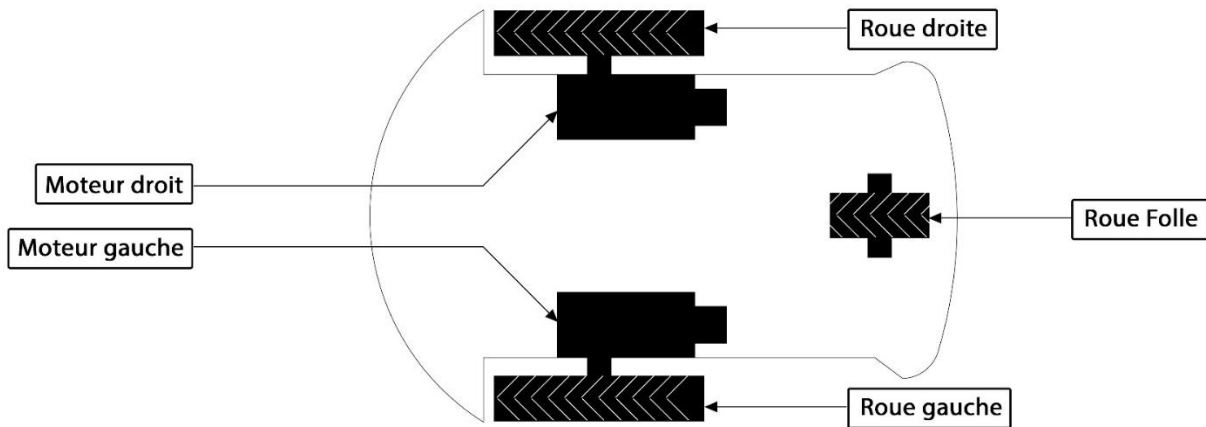


Figure 52: Vue de dessus du châssis du robot.

En contrôlant ces moteurs individuellement on peut contrôler la direction dans laquelle le robot se déplace. Le mouvement global du robot dépend de la combinaison des mouvements de rotation de chaque roue.

b. Commande des motoréducteurs :

Pour contrôler le sens de rotation du robot on contrôle individuellement les moteurs gauche et droit du robot comme expliqué dans le chapitre précédent. Si les deux moteurs tournent dans le sens des aiguilles d'une montre le robot avance, et s'ils tournent dans le sens inverse des aiguilles d'une montre le robot recule. Pour tourner vers la droite ou vers la gauche les deux moteurs doivent tourner en sens opposés (voir Figure 23).

Pour contrôler le sens de rotation des moteurs on va utiliser un Pont H double (Dual Full-Bridge driver), le L298 qu'on retrouve quasiment sur la majorité des shields moteurs Arduino. C'est une solution intégrée très efficace, car ça nous évite de faire beaucoup de branchements.

b.1. Le pont H double L298 :

Le L298 est un circuit intégré à 15 broches (voir la Figure 53), il intègre deux ponts en H (le pont A et le pont B), idéal pour notre application, il permet un pilotage assez aisé des deux moteurs au même temps. Il fonctionne à de fortes tensions et à de forts courants, Il est capable de débiter jusqu'à 2A nominal, et une tension de 46V (il peut aller jusqu'à 50V) par pont. Pour un fonctionnement optimal il est conseillé de placer des condensateurs de 100nF sur chaque ligne d'alimentation (en général on utilise des

condensateurs de 100 μ F 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), la tension d'alimentation logique est entre 4.5V à 7V.

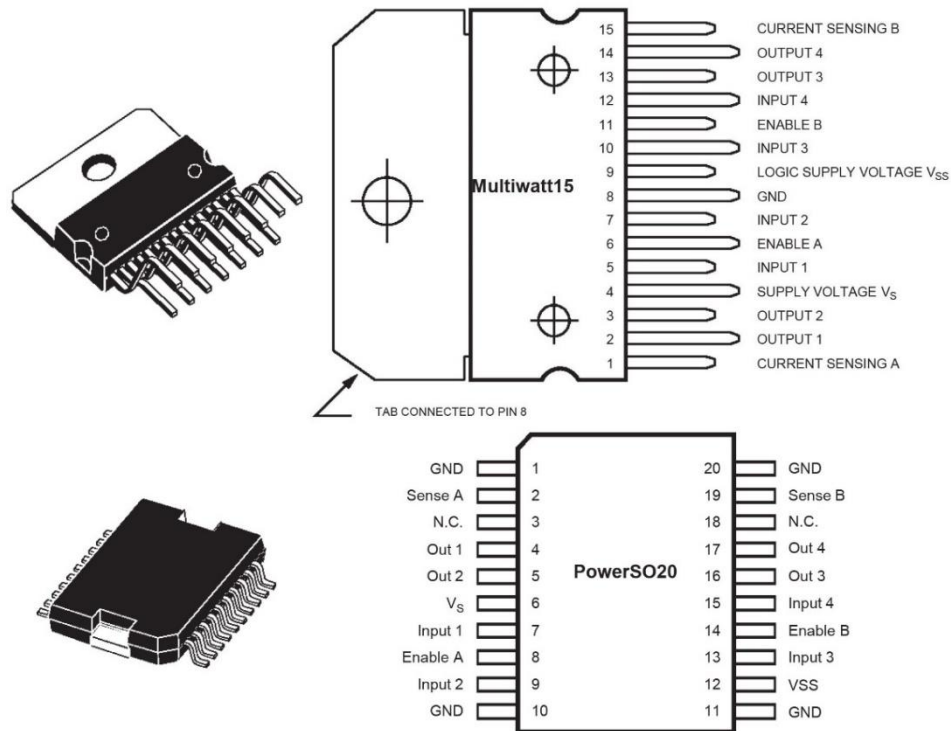


Figure 53: Brochage du circuit intégré L298

Le tableau ci-dessous représente les différentes fonctions des broches du pont H L298 :

Broches sur le paquet Multiwatt15	Broches sur le paquet PowerSO20	Nom	Fonction
1 et 15	2 et 19	Sense A et Sense B	Entre ces broches et GND en place une résistance pour contrôler le courant de la charge (les moteurs dans notre cas).

2 et 3	4 et 15	Out 1 et Out 2	Les sortie du pont A; le courant qui travers la charge qui est connecté entre eux est surveillé par la broche 1.
4	6	Vs	Tension d'alimentation de puissance (5V >Vs> 46V). un condensateur de 100nF doit être connecté entre cette broche et 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'activé/désactivé des pont A et B respectivement. Ils sont à l'état désactivé au niveau bas (L).
8	1, 10, 11 et 20	GND	La terre.
9	12	VSS	Tension d'alimentation pour le block logique (4.5V >VSS> 7V). recommander a utilisé une tension de 5V, et d'ajouter un condensateur de 100nF entre cette broche et GND.
10 et 12	13 et 15	Input 3 et Input 4	Ce sont les entrées du pont B (compatible à la technologie TTL)
13 et 14	16 et 17	Out 3 et Out 4	Les sortie du pont B; le courant qui travers la charge qui est connecté entre eux est surveillé par la broche 15.
-	3 et 18	N.C.	Non connecté.

Note :

Pour contrôler la vitesse de rotation des moteurs, 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 54), ainsi on contrôle la vitesse de rotation des canaux A et B respectivement.

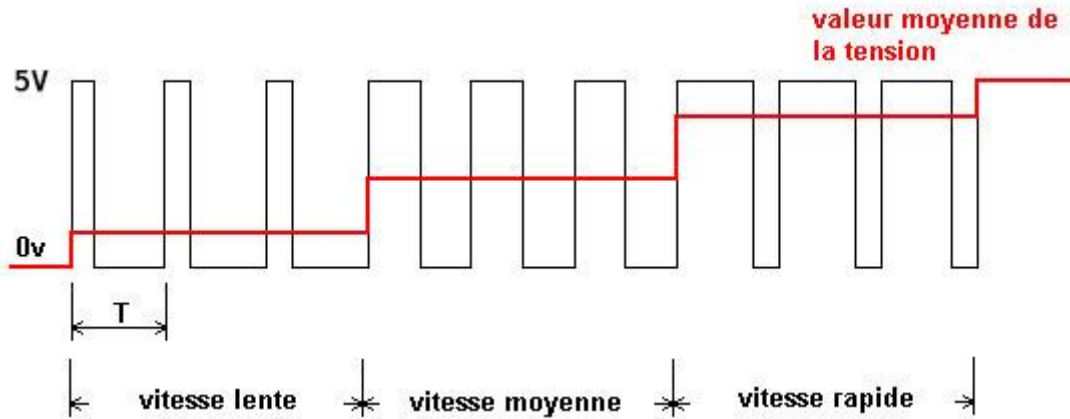


Figure 54: Contrôle de vitesse via la PWM

Comme on peut l’observer sur la Figure 55, à l’intérieur du circuit L298 en a deux pont H : le pont A à gauche, et le pont B à droite.

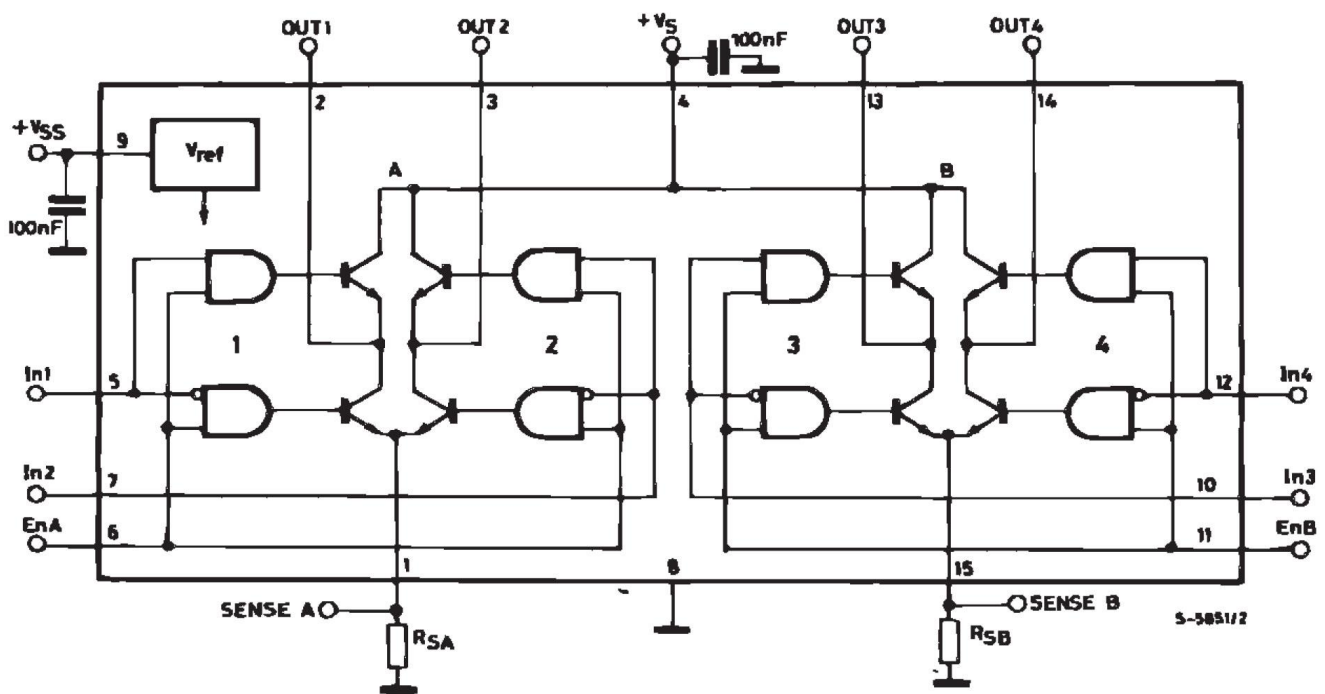


Figure 55: Schéma synoptique du circuit intégré L298.

Les signaux de contrôle des ponts A et B:

Le Pont A :

	Input 1	Input 2 (IN2)	Fonction
	1	(IN2)	

	(IN1)		
Enable A (ENA) = L	-	-	Le pont A désactivé.
Enable A (ENA) = H	H	L	Le moteur branché sur le pont A (entre Output 1 (OUT1) et 2(OUT2)) tourne dans le sens des aiguilles de la montre.
	L	H	Le moteur branché sur le pont A (entre Output 1 (OUT1) et 2(OUT2)) tourne dans le sens inverse des aiguilles de la montre.
	Input 1 = Input 2		Freinage rapide du moteur branché sur le 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 (OUT3) et 4 (OUT4)) tourne dans le sens des aiguilles de la montre.
	L	H	Le moteur branché sur le pont B (entre Output 3 (OUT3) et 4 (OUT4)) tourne dans le sens inverse des aiguilles de la montre.
	Input 3 = Input 4		Freinage rapide du moteur branché sur le pont B.

Avec :

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

Note :

Comme on peut le voir sur la Figure 55, il n'y a pas de diodes de roue libre, c'est pour cela qu'il faut en rajouter pour protéger des courants de retour, de préférence des diodes Schottky rapide pour fonctionner dans des régimes de hautes fréquences.

b.2. L'Arduino Motor shield :

L'un des avantages de l'utilisation des cartes Arduino est le fait qu'il existe des cartes d'extension (appelée shield ou bouclier en français), qui permettent de rajouter des fonctionnalités à la carte arduino, que ce soit pour rajouter la capacité de communiquer via Ethernet ou WIFI, XBee ou la capacité de lecture/écriture sur une carte mémoire ou un écran LCD ...etc. il existe aussi des shields motor driver qui permettent une commande assez aisée des moteurs à courant continu ou même des moteurs pas à pas dans certains cas, c'est pour cela qu'on va utiliser l'Arduino Motor shield (voir la Figure 56), qui est basé sur le circuit intégré L298P (la version du pond H double L298 en paquet PowerSO20 monté en surface), il permet de contrôler la vitesse et la direction de chaque moteur individuellement, on peut aussi avec ce shield mesurer les courants absorbés par chaque moteurs.

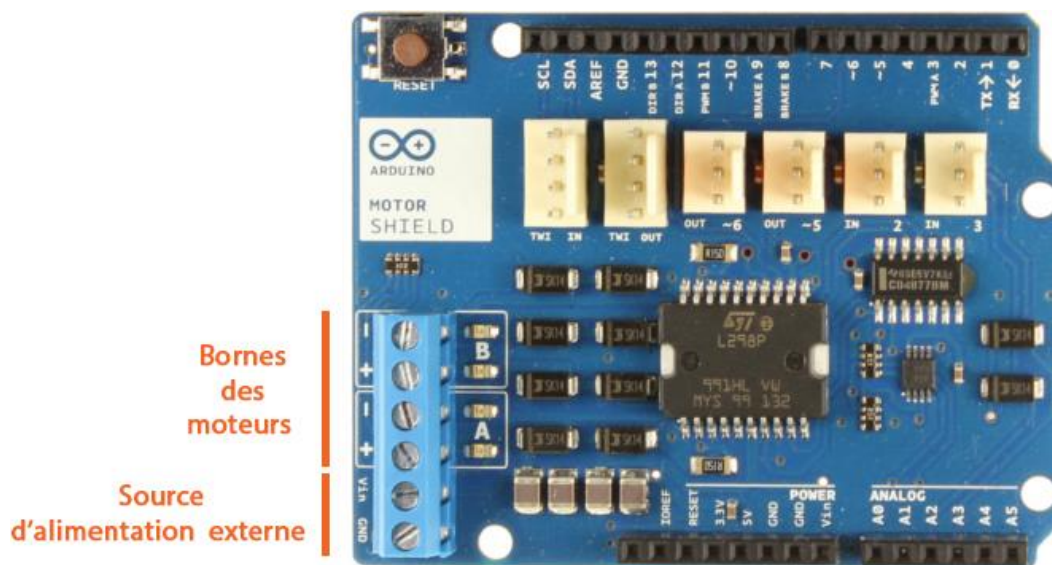


Figure 56: Arduino Motor shield.

Les caractéristiques du shield sont comme suit :

- Tension d'alimentation : de 5V jusqu'à 12V.
- Détection de courant : 1.65V/A.

- Peut contrôler deux moteurs à courant continue, ou un moteur pas à pas.
- Courant maximum : 2A par canal ou 4A (avec une source d'alimentation externe).
- Deux modes de freinage : fonction d'arrêt immédiat, et arrêt de roulement libre (qui met un peu plus de temps).

Afin de protéger l'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 ceux de la carte arduino en coupant le jumper « Vin Connect » sur la partie arrière du shield.

Les entrées/sorties du shield se présentent comme suit :

Fonction	Broche du canal A	Broche du canal B
Direction	12	13
PWM	3	11
Frein	9	8
Détection de courant	A0	A1

A noter que si l'on désire pas utiliser la fonction de détection de courant ou du freinage, ces deux fonctions peuvent être désactivées en coupant les jumpers sur le côté arrière du shield.

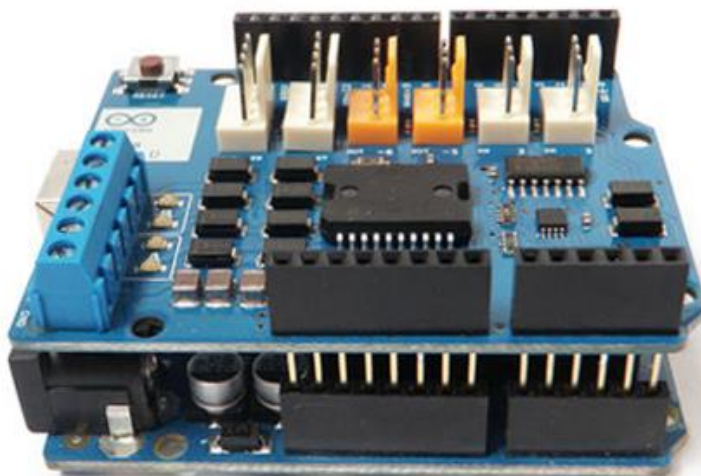


Figure 57: Arduino Motor shield monté sur une carte Arduino UNO.

L'alimentation utilisée est un pack de 4 batteries Alkaline de taille AA dont la capacité est de 6V (1.5V x 4), ainsi en respect la consigne qui nous oblige à utiliser une source d'alimentation de 5V à 12V.

Le branchement des différents éléments est représenté sur la figure suivante :

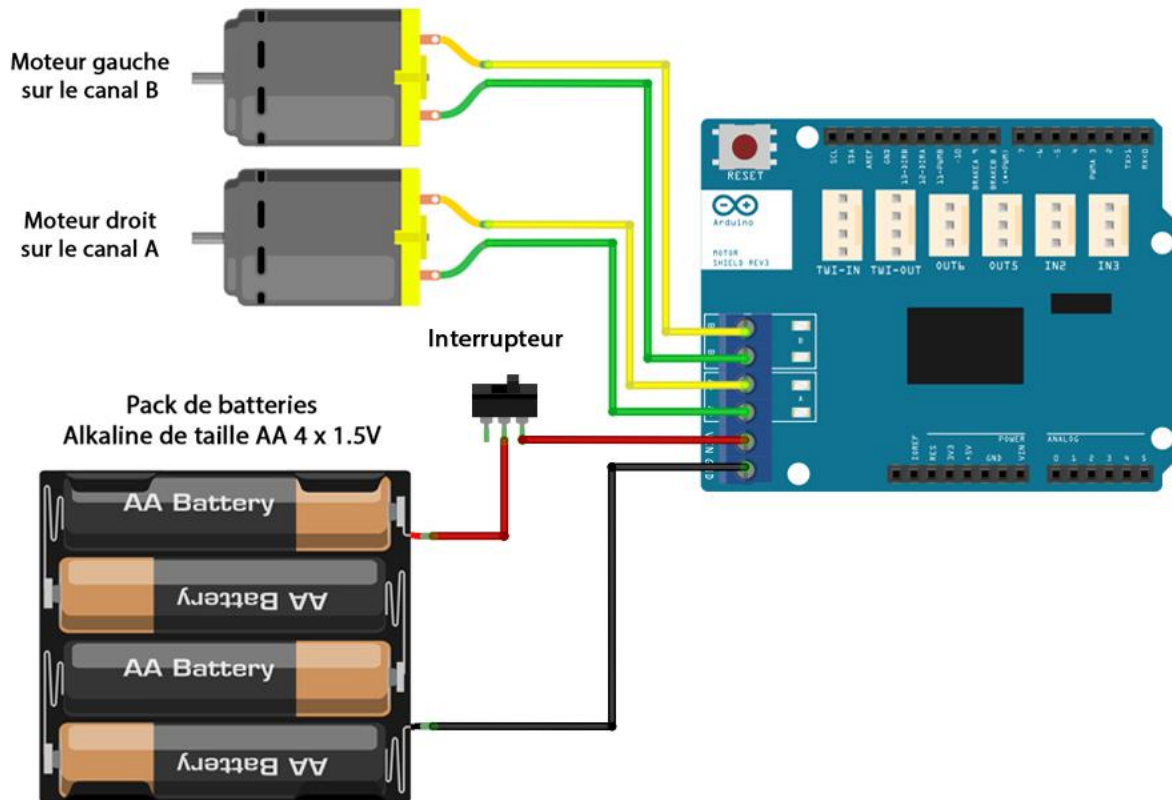


Figure 58: branchement des moteurs vers le shield moteur.

III.3.3. La communication via Bluetooth :

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 avoir (disfonctionnement d'un composant du robot, faille de logique, environnement non contrôlé, données erronées issues du capteur, ...etc.). Le contrôle via Bluetooth a été choisie dans ce travail afin d'accomplir cette tâche bien précise mais d'autre moyen de communications (sans filaires) peuvent être utilisé comme solution alternative (IR, WIFI, XBee, Zigbee, ...etc.). L'avantage de l'utilisation de la technologie Bluetooth c'est qu'elle offre un moyen simple de contrôle du robot via plusieurs appareils (ordinateur, téléphone portable, tablette), ce qui réduit considérablement le coût (pas besoin de réaliser l'émetteur).

La commande à distance du robot ce fait en utilisant le module HC-05 (FC-114), le module HC-05 à 6 pattes dont 4 uniquement sont utilisées généralement (voir la **Figure 59**):

- **EN** : non utilisée (cette broche est utilisée pour activé ou désactivé le module).
- **VCC** : alimentation de 5V à relier vers 5V de l'arduino.
- **GND** : la terre à relier vers le GND de l'arduino.
- **TXD** : à relier vers RX de l'arduino.
- **RXD** : à relier vers TX de l'arduino, mais par le biais d'un diviseur de tension car la tension d'entrée du module acceptée est de 0 à 3.3V, alors que le TX de l'arduino émet jusqu'à 5V.
- **STATE** : non utilisée (il informe sur l'état du module).

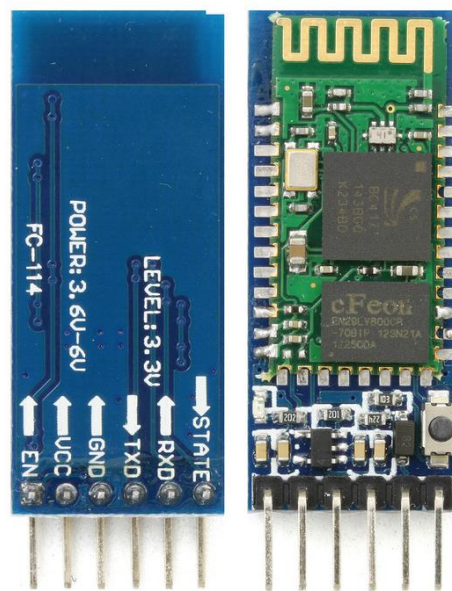


Figure 59: le module HC-05 FC-114.

Le branchement du module vers l'arduino ce fait comme illustré sur la figure suivante :

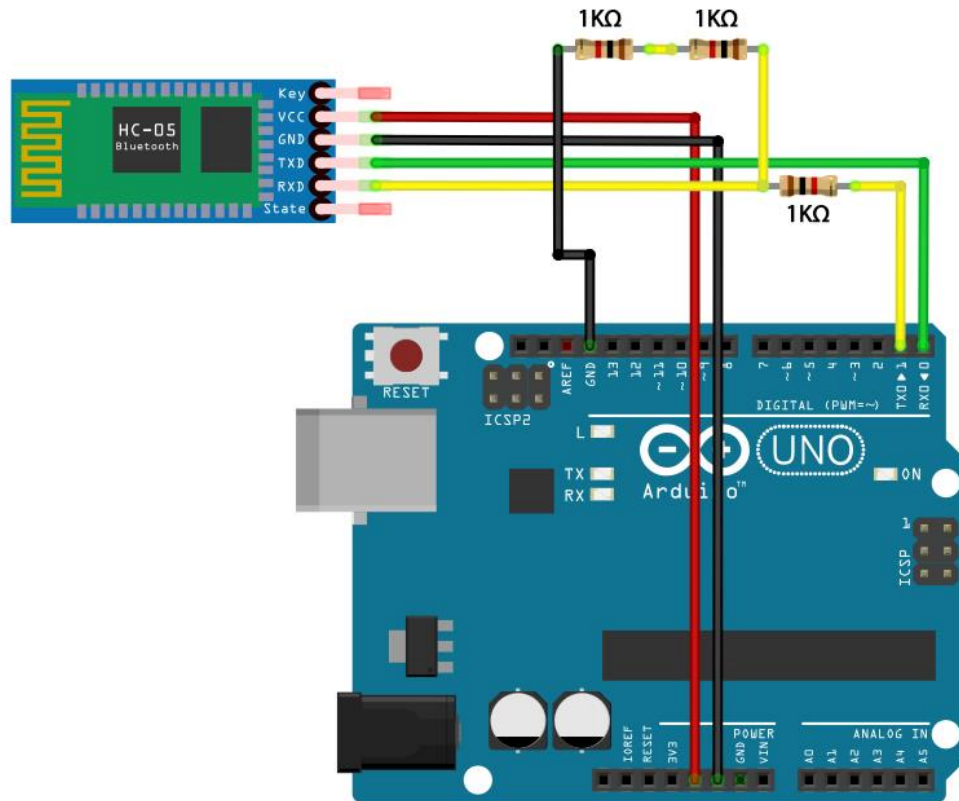


Figure 60: schéma de branchement du module HC-05 vers l'arduino.

Une fois le module est alimenté on peut se connecter au module dont le nom par défaut est HC-05 avec le code pin 1234.

III.4. Réalisation logicielle :

III.4.1. Algorithme :

Le robot utilise une certaine logique afin de calculer la bonne trajectoire à suivre toute en évitant les obstacles sur son chemin. L'algorithme de la Figure 61, et Figure 62 expliquent comment le robot procède pour accomplir sa mission.

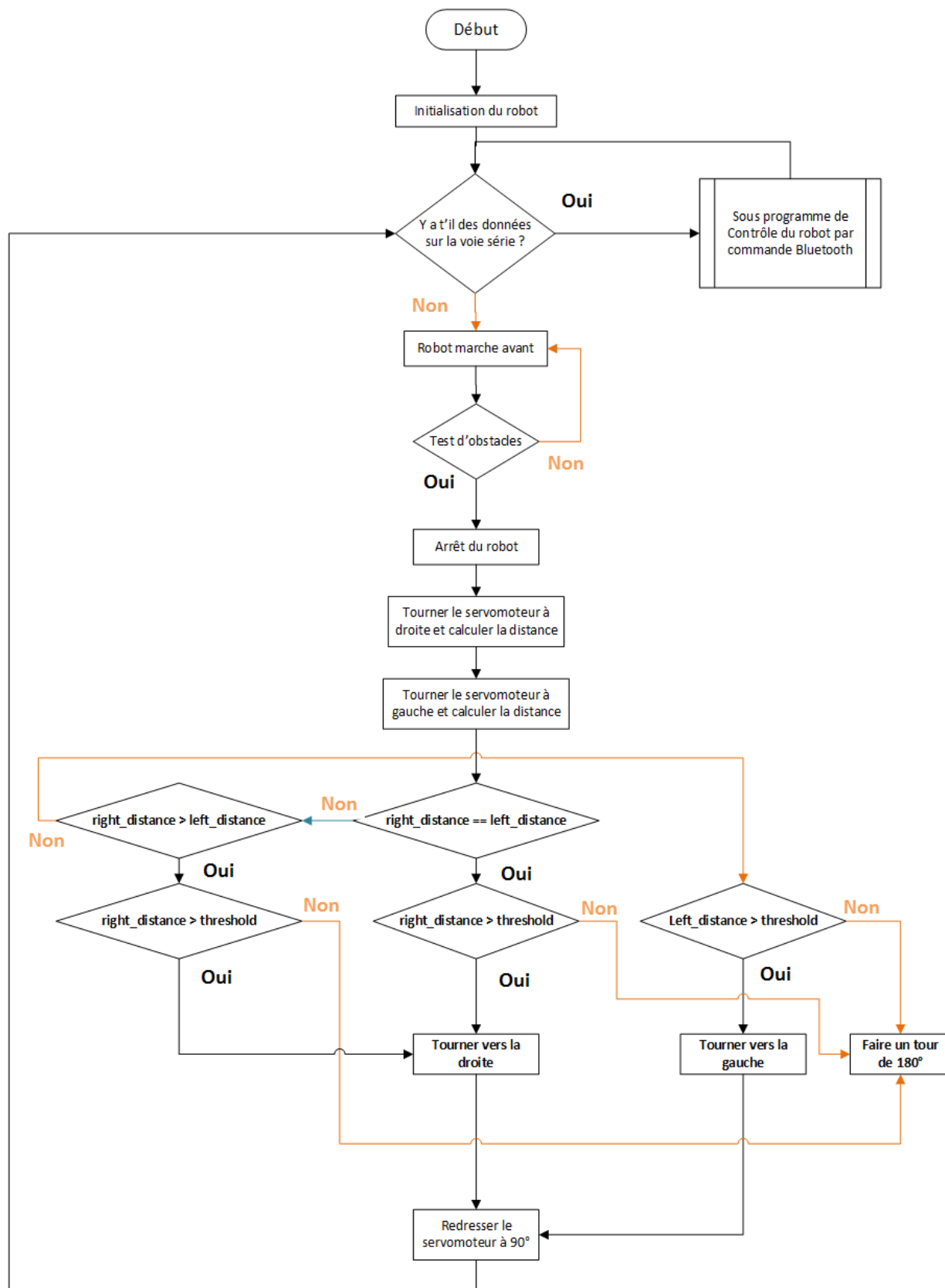


Figure 61: Algorithme du fonctionnement du robot.

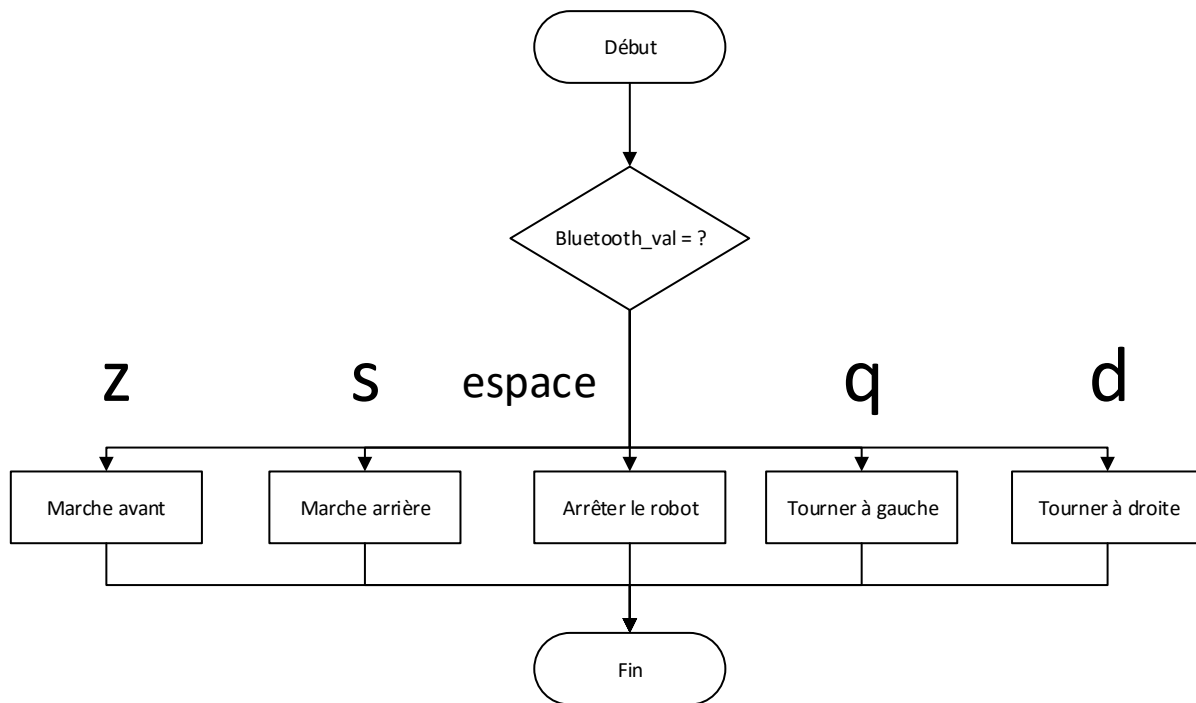


Figure 62: Sous-programme de commande du robot via Bluetooth.

Comme on peut le voir sur la **Figure 61**, après l'initialisation des différentes variables et paramètres du robot, un test de disponibilité de données sur la voie série (Bluetooth SPP) est faite. Les commandes envoyées via le Bluetooth sont enregistrées dans la variable `Bluetooth_val`, et puis selon le contenu de cette variable le robot effectue une manœuvre (si on envoie le caractère « Z » le robot marche en avant, si le caractère « S » est envoyé par contre le robot fait marche arrière, ...etc.).

Si aucune donnée n'est disponible sur la voie série le robot fonctionne en mode autonome, il avance continuellement et ne s'arrête que si un obstacle est détecté. Une fois un obstacle est détecté le robot fait un arrêt immédiat, et tourne sa tête (le servomoteur + capteur ultrasonore) vers le côté droit et mesure la distance avec le capteur ultrasonore (la distance mesurée est sauvegardée dans la variable `right_distance`), il tourne ensuite sa tête vers le côté gauche et mesure la distance de nouveau (la distance mesurée est sauvegardée dans la variable `left_distance`).

Après avoir pris les mesures le robot suit une certaine logique pour déterminer quel chemin à suivre. Cette logique consiste à déterminer la distance la plus grande (en faisant une comparaison entre `left_distance`

et `right_distance`), ensuite il compare la distance la plus grande à un seuil donné (dont le nom de variable est `threshold`).

Si la distance est plus grande que le seuil, le robot se tourne vers cette direction, et redresse le servomoteur à 90°, et continue d'avancer. Par contre si la plus grande distance est inférieure à la distance seuil le robot fait un demi-tour (il tourne 180°).

III.4.2. Code source :

```
#define PING_PIN          2 //capteur ultrasonore attaché à la broche 2
#define SERVOMOTOR_PIN   5 //servomoteur attaché à la broche 3
//broches associé avec le moteur droit
#define PWM_A            3
#define DIR_A            12
#define BRAKE_A          9
//broches associé avec le moteur gauche
#define PWM_B            11
#define DIR_B            13
#define BRAKE_B          8

#include <Servo.h>

//déclarations du capteur ultrasonore
constint mesureDelay = 700; //temps de entre chaque mesure de distance (gauche et droite)
long duration;
long distance; //distance vers l'avant
long left_distance; //distance du côté gauche
long right_distance; //distance du côté droit
constint threshold = 30; //la distance seuil

//création d'un objet servomotor
Servo servomotor;

//DC Motors declarations
//pour corriger la trajectoire du robot on a besoin de synchroniser la vitesse des deux moteurs.
constint RMSpeed = 245; //la vitesse de rotation du moteur droit
constint LMSpeed = 240; //la vitesse de rotation du moteur gauche
constint turningDelay = 210; //temps que prend le robot pour ce tourner dans une direction donné.

char bleutooth_val; //commande reçue via bluetooth

voidsetup()
{
  servomotor.attach(SERVOMOTOR_PIN);
  servomotor.write(90); //centrer le servomoteur dès le début
```

```
pinMode(PWM_A, INPUT);
pinMode(DIR_A, OUTPUT);
pinMode(BRAKE_A, OUTPUT);

pinMode(PWM_B, INPUT);
pinMode(DIR_B, OUTPUT);
pinMode(BRAKE_B, OUTPUT);

Serial.begin(9600);
Serial.println(F("CONNECTED..."));
}
```

```
void loop()
{
  if(Serial.available())
  {
    bluetooth_val =Serial.read();

    if(bluetooth_val == 'z')
    {
      moveForward();
    }
    elseif(bluetooth_val == 's')
    {
      moveBackward();
    }
    elseif(bluetooth_val == 'd')
    {
      turnRight();
      stopMotors();
    }
    elseif(bluetooth_val == 'q')
    {
      turnLeft();
      stopMotors();
    }
    elseif(bluetooth_val == ' ')
    {
      stopMotors();
    }
  }
  else
  {
    delay(200); //faire une pause de 300ms entre chaque mesure
    distance = ping();
    if(distance == 0) return;
    Serial.print("Distance vers l'avant : ");
    Serial.print(distance);
    Serial.println(" CM");

    if(distance <= threshold)
    {
      stopMotors();
    }
  }
}
```

```
//mesurer la distance du côté droit
servomotor.write(0);
delay(mesureDelay);
right_distance = ping();

Serial.print("Distance vers la droite : ");
Serial.print(right_distance);
Serial.println(" CM");

//mesurer la distance du côté gauche
servomotor.write(180);
delay(mesureDelay);
left_distance = ping();

Serial.print("Distance vers la gauche : ");
Serial.print(left_distance);
Serial.println(" CM");

servomotor.write(90);

//comparer les distance
if(right_distance == left_distance)
{
  if(right_distance > threshold)
  {
    turnRight();
  }
  else
  {
    Serial.println("Tourner 180 degret...");
    turnRight();
    stopMotors();
    turnRight();
    stopMotors();
  }
}
else
{
  if(right_distance > left_distance)
  {
    if(right_distance > threshold)
    {
      turnRight();
    }
    else
    {
      Serial.println("Tourner 180 degret...");
      turnRight();
      stopMotors();
      turnRight();
      stopMotors();
    }
  }
  else
  {
```

```
        if(left_distance > threshold)
        {
            turnLeft();
        }
        else
        {
            Serial.println("Tourner 180 degret...");
            turnRight();
            stopMotors();
            turnRight();
            stopMotors();
        }
    }
}
else
{
    servomotor.write(90);
    moveForward();
}
}
```

```
long ping()
{
    //le capteur ultrason est activé avec une pulsation de niveau haut >= 2 microseconde
    //on envoie une pulsation de niveau bas en avance pour assurer une pulsation de
    //niveau haut.

    pinMode(PING_PIN, OUTPUT);

    digitalWrite(PING_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(PING_PIN, HIGH);
    delayMicroseconds(5);
    digitalWrite(PING_PIN, LOW);

    //la même broche est utiliser pour l'émission et la réception du signal, donc une
    //fois le signal est envoyer on doit reconfigurer la broche en entrée pour
    //recevoir le signal.
    pinMode(PING_PIN, INPUT);
    duration = pulseIn(PING_PIN, HIGH);

    // convertir le temps en distance
    return duration / 29 / 2;
}
```

```
void moveForward()
{
    Serial.println(F("Marche avant..."));

    digitalWrite(BRAKE_A, LOW);
}
```

```
digitalWrite(BRAKE_B, LOW);
digitalWrite(DIR_A, HIGH);
digitalWrite(DIR_B, HIGH);

analogWrite(PWM_A, RMSpeed);
analogWrite(PWM_B, LMSpeed);

}

void moveBackward()
{
  Serial.println(F("Marche arriere..."));

  digitalWrite(BRAKE_A, LOW);
  digitalWrite(BRAKE_B, LOW);
  digitalWrite(DIR_A, LOW);
  digitalWrite(DIR_B, LOW);

  analogWrite(PWM_A, RMSpeed);
  analogWrite(PWM_B, LMSpeed);

}

void turnRight()
{
  Serial.println(F("Tourner a droite..."));

  digitalWrite(BRAKE_A, LOW);
  digitalWrite(BRAKE_B, LOW);
  digitalWrite(DIR_A, LOW);
  digitalWrite(DIR_B, HIGH);

  analogWrite(PWM_A, RMSpeed);
  analogWrite(PWM_B, LMSpeed);

  delay(turningDelay);
}

void turnLeft()
{
  Serial.println(F("Tourner a gauche..."));

  digitalWrite(BRAKE_A, LOW);
  digitalWrite(BRAKE_B, LOW);
  digitalWrite(DIR_A, HIGH);
  digitalWrite(DIR_B, LOW);

  analogWrite(PWM_A, RMSpeed);
  analogWrite(PWM_B, LMSpeed);

  delay(turningDelay);
}
```

```
void stopMotors()
{
  Serial.println(F("Arret..."));

  digitalWrite(BRAKE_A, HIGH);
  analogWrite(PWM_A, 0);
  digitalWrite(BRAKE_B, HIGH);
  analogWrite(PWM_B, 0);

  delay(500);
}
```

III.4.3. Programmation de l'Arduino :

Pour programmer l'Arduino on la branche à un Ordinateur via un câble USB (fiche A vers fiche B voir la Figure 63), puis sur cet ordinateur on a besoin d'installer l'environnement de développement Arduino IDE (Integrated Development Environment) [22] qui va nous permet de programmer la carte Arduino.



Figure 63: Câble USB (fiche A vers fiche B)

Note :

Les logiciels écrits avec l'IDE Arduino sont appelés des « Croquis » (ou Sketch en anglais) et ils ont l'extension «.ino».

La Figure 64 représente l'interface de l'environnement de développement Arduino.

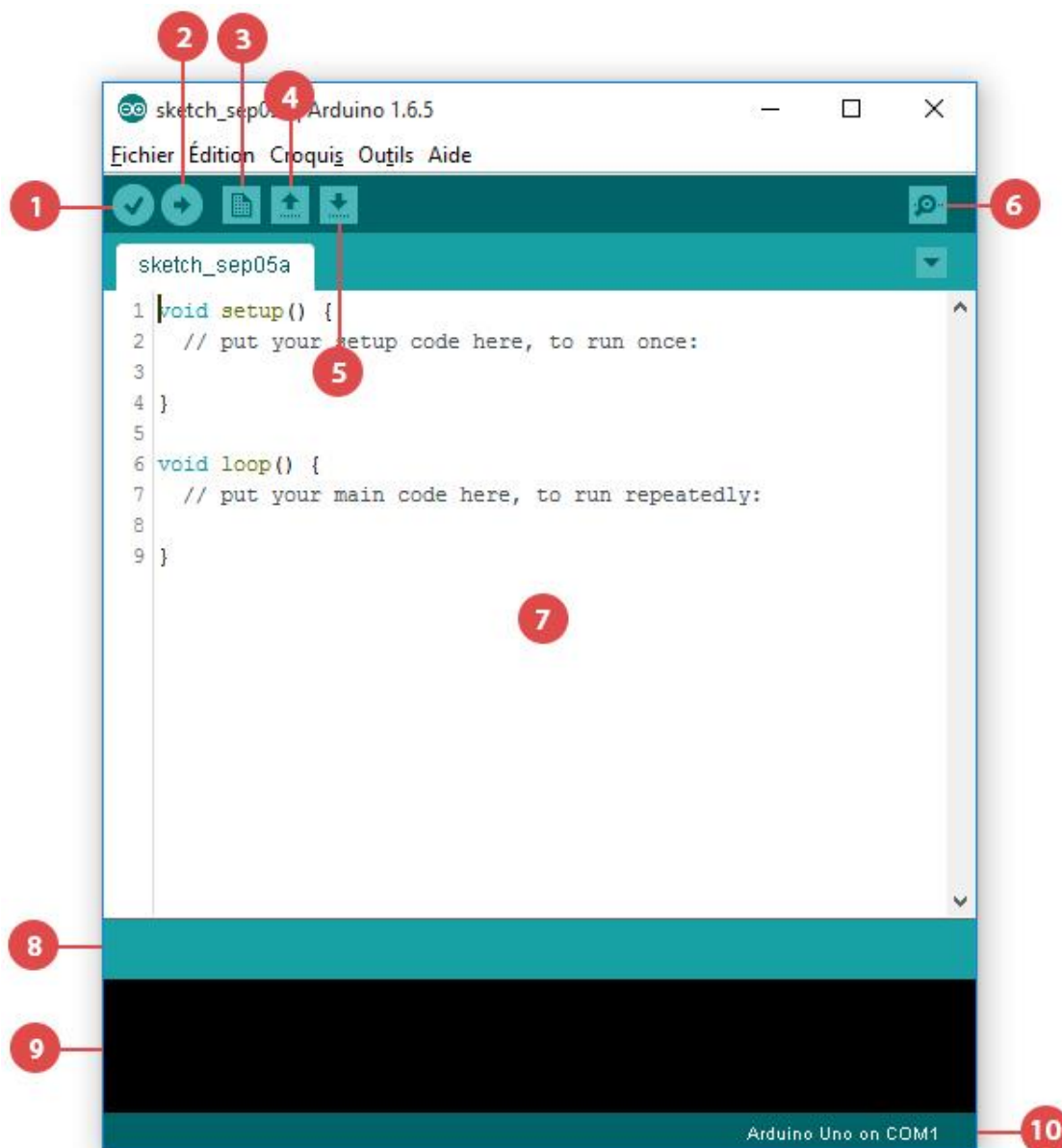


Figure 64: l'interface de l'environnement de développement Arduino IDE.

- 1 – **Vérifier** : ce bouton sert à compiler et vérifier s'il y a des erreurs de syntaxe.
- 2 – **Téléverser** : Compile le code si il n'est pas déjà compilé ensuite il l'envoie à la carte Arduino.
- 3 – **Nouveau** : ce bouton ouvre un nouveau croquis.
- 4 – **Ouvrir** : ce bouton permet d'ouvrir un croquis existant.
- 5 – **Enregistrer** : ce bouton permet d'enregistrer le croquis courant.
- 6 – **Moniteur Série** : ce bouton permet d'ouvrir le moniteur qui permet la communication série entre la carte Arduino et l'ordinateur sur laquelle on l'a branché.

- **7 – Editeur de texte** : cette région est la partie où se trouve le code source du croquis actuel.
- **8 – Région de message** : c'est ici où l'IDE nous informe sur le statut du téléversement.
- **9 – Console** : dans cette région l'IDE nous montre les messages d'erreurs complet, et voire même les différentes étapes de compilation en temps réel si choisie dans les préférences.
- **10 – Carte et port série** : montre quelle carte choisie et sur quel port elle se trouve.

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

- aller sur le menu **Outils**, ensuite cliquer sur **Type de carte** et puis sélectionner la carte désiré (qui est « Arduino UNO » dans notre cas).
- Ensuite il faut s'assurer qu'en est connecté sur le bon port série (surtout si on programme plusieurs cartes au même temps). Pour cela il suffit d'aller sur le menu **Outils** ensuite sur **Port** et choisir le bon port.
- Après il ne reste plus qu'à cliquer sur le bouton « **Téléverser** » et attendre un peu. Si le code source n'a pas d'erreurs la carte Arduino doit maintenant être programmée.

III.5. Tests et Résultats :

Une fois la réalisation matérielle et logicielle est faite, plusieurs calibrages ont été fait au robot dus aux imperfections que présente les différents organes du robot, et du déplacement du centre de gravité du robot au fur et à mesure qu'on rajoute des composants au châssis du robot.

On peut citer les exemples suivants :

- La vitesse de rotation du moteur droit est légèrement supérieure à celle du moteur gauche, car si les deux moteurs tournent avec la même vitesse le robot a tendance à tourner à gauche, la différence de vitesse entre les deux moteurs permet de corriger la trajectoire du robot.
- La distance seuil n'a été déterminée qu'une fois le robot est opérationnel, car il ne suffit pas de fixer la distance à une valeur donnée sans prendre en considération la vitesse de déplacement du robot et la fréquence avec laquelle le capteur ultrasonore effectue les mesures.
- Pour mesurer la distance du côté droit et du côté gauche, il a fallu introduire un délai de temps de 700ms pour permettre au servomoteur d'attendre la position désirée et prendre la mesure de distance avec le capteur ultrasonore.

La figure ci-dessous montre les différentes perspectives du robot une fois terminé :

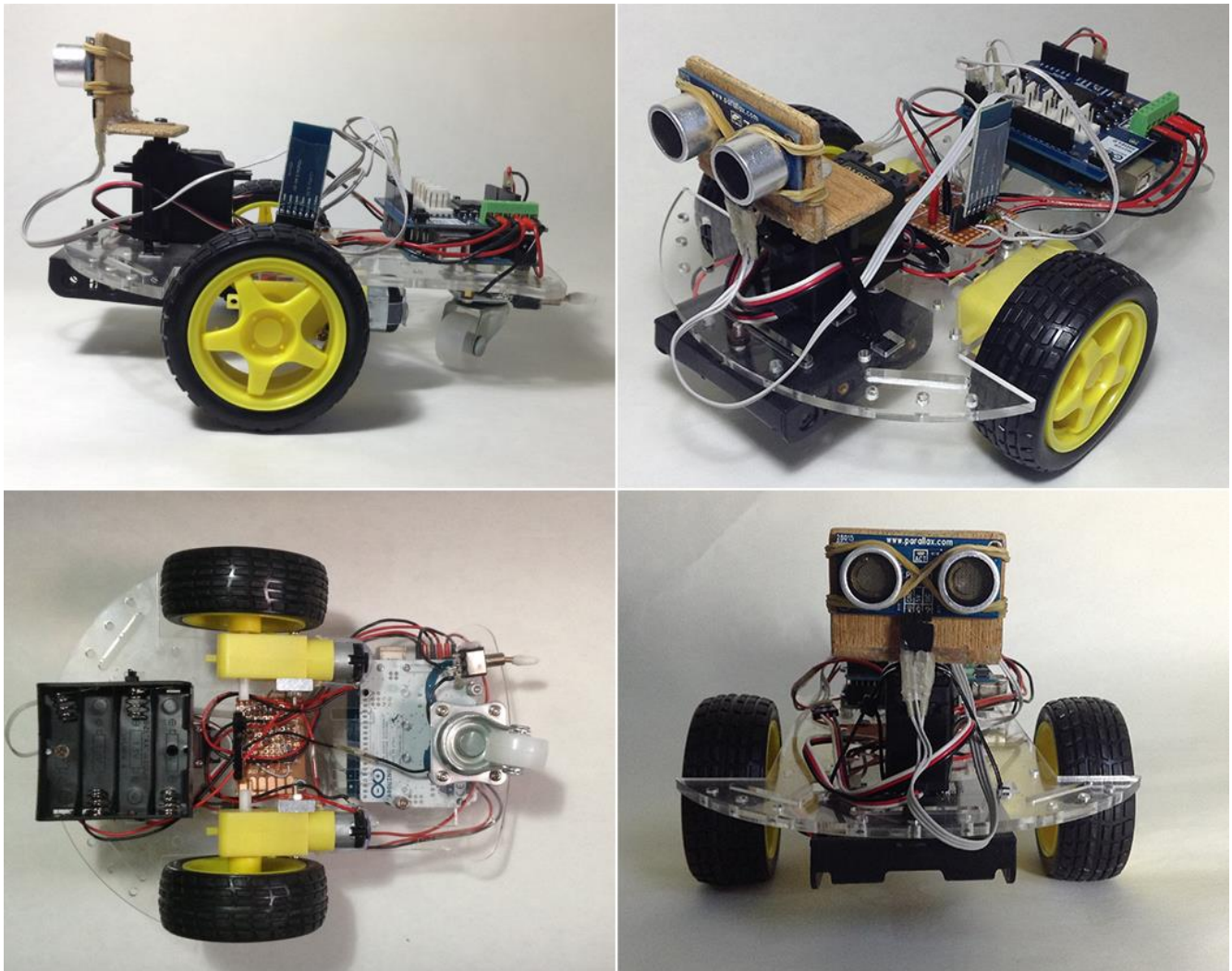


Figure 65: vue des différentes perspectives du robot.

Afin de contrôler la distance le robot plusieurs appareils peuvent être utilisés, on peut utiliser un téléphone portable dans lequel on installe une application qui permet d'envoyer des caractères via Bluetooth tout simplement (voire même créer une application java avec une simple interface graphique de 4 boutons), ou une tablette. Pour notre part on va utiliser un ordinateur portable équipé de la fonctionnalité de Bluetooth (intégrée ou avec un adaptateur Bluetooth), et le logiciel Tera Term qui est un terminal qui nous permet de se connecter au robot, et d'envoyer et recevoir des données via Bluetooth.

Lorsque on se connecte au module HC-05, deux port COM sont créé (COM4 et COM5 dans notre cas), l'un d'eux est dans la direction «Incoming » l'autre dans la direction « Outgoing » (voir la **Figure 66**), pour envoyer des commandes via Bluetooth il faut etre connecté au port dont la direction de transmission est « Outgoing ».

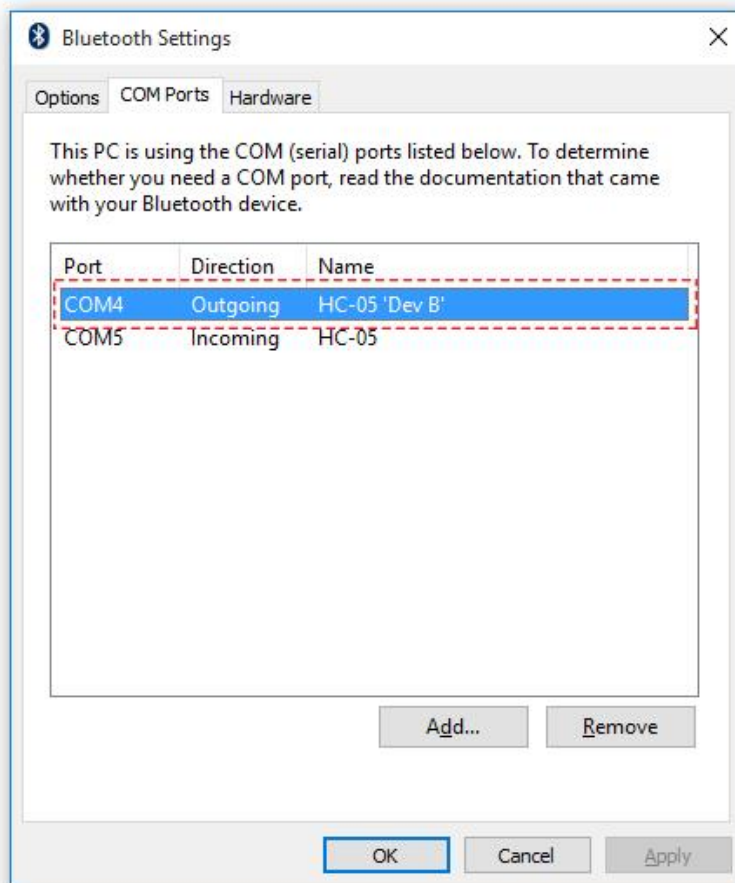


Figure 66: Paramètres Bluetooth.

Avec le logiciel Tera Term dès le lancement du logiciel il nous donne une fenêtre qui nous permet de se connecter au bon port, comme en peut le voir sur la figure suivante :

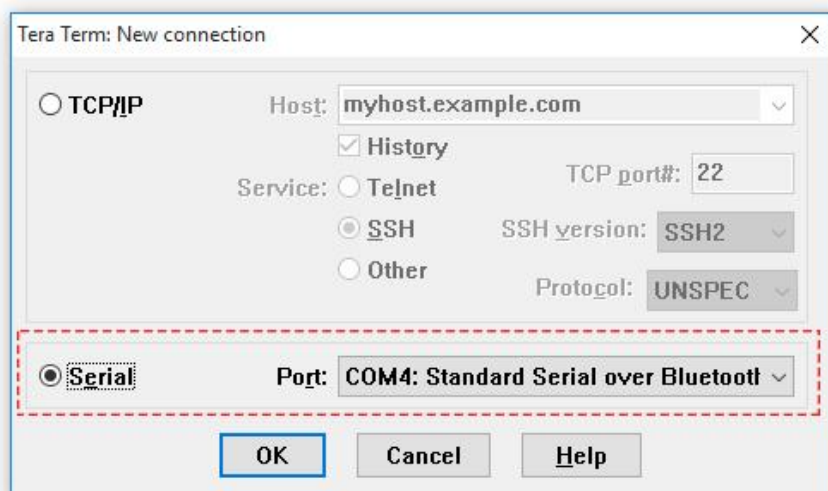


Figure 67: Fenêtre du choix de connexion du logiciel Tera Term.

Une fois connecté on reçoit le message «connected » (voir la Figure 68) sur le terminal, et après ça on est prêt à envoyer et recevoir des données via Bluetooth.

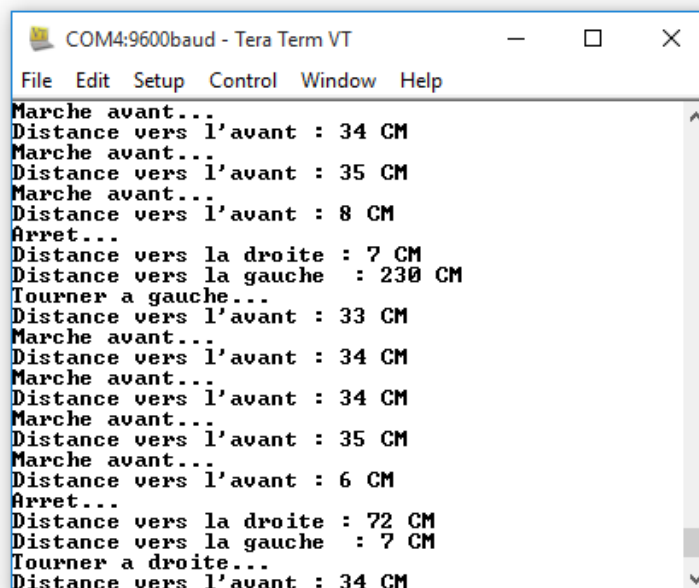


Figure 68: Fenêtre principale du logiciel Tera Term.

Comme on peut le voir sur la Figure 68 on reçoit continuellement des données du robot qui nous informe sur l'état du robot, ce qui nous offre un moyen de « feedback » (retour d'information) qui permet un débogage facile du robot.

III.6. Conclusion :

Dans ce chapitre on a fait la réalisation pratique après conception, d'un robot mobile autonome avec la possibilité d'être télécommander par voie Bluetooth SPP. Dans ce chapitre on a pu faire la réalisation d'un robot mobile de type tricycle, basé sur une carte Arduino UNO, dont le moyen de locomotion consiste en deux motoréducteurs (DG01D) commandés avec un shield moteur basé sur un pont H double (le L298). La perception est atteinte via un capteur ultrasonore (Parallax 28015) qui permet la mesure de distance, et un servomoteur (FUTABA S3010) qui joue le rôle du « cou » du robot en permettant au robot de mesurer la distance du côté gauche, et du côté droit sans tourner entièrement.

Le robot suit un certain algorithme (voir la **Figure 61**) pour calculer la trajectoire à suivre. Si une Défaillance est enregistrée ou un besoin de contrôle manuel est souhaité, le robot peut être télécommandé à distance via Bluetooth ce qui permet de garder l'aspect mobile du robot via le module Bluetooth HC-05.

Une fois la réalisation pratique est terminée, plusieurs calibrages et tests sont effectués afin d'assurer le bon fonctionnement du robot.

CONCLUSION GENERALE :

La conception d'un robot autonome et mobile est un défi. Parmi les nombreuses capacités devant être mises à contribution pour relever un tel défi, un robot autonome mobile doit avant tout être mobile ce qui nous impose d'avoir une source d'alimentation embarquée capable d'alimenter tous les organes du robot, un poids global pas trop grand, un châssis qui permet une mobilité facile est efficace dans les endroits étroits. L'autre aspect important est l'autonomie du robot, en effet un robot autonome doit être capable de prendre ses propres décisions et planifier lui-même sa trajectoire toute en évitant les obstacles rencontrés sur son chemin en suivant une certaine logique. Malgré l'aspect autonome du robot les imperfections des composants du robot peut présenter des pannes matérielles (dysfonctionnement d'un capteur ou d'un moteur, ...etc.), qui aboutit à une défaillance logicielle (car si un capteur par exemple ne nous signale pas la présence d'un obstacle le robot ne sera pas en mesure de l'éviter) .Par conséquent un moyen de communication (sans fils et à faible consommation) est requis, afin de pouvoir télécommander le robot à distance si on a besoin de corriger sa trajectoire, et avoir un feedback permanent.

Dans ce mémoire on a pu réalisé avec succès un robot autonome mobile 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 pour prendre les mesures de distance dans les différentes directions, et de deux motoréducteurs commandés avec un shield moteur basé sur le pont H double L298 comme moyen de locomotion. Ainsi que le module HC-05 qui permet d'avoir une liaison Bluetooth afin de pouvoir télécommander à distance le robot.

Les possibilités d'améliorations 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 labyrinthes (catégorie populaire dans les concours de robotique), 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 utiliser un autre moyen de téléguidage à plus grande portée que le Bluetooth. On peut aussi changer l'unité centrale du robot par une raspberry pie, ou une carte beaglebone (cartes de développement plus performantes que l'arduino

uno) avec une caméra comme moyen de perception (en faisant du traitement d'images) à la place du capteur ultrasonore.

Bibliographie :

- [1] W. Richard, L. Daniel et M. Robert , «Systems Engineering the Curiosity Rover: A Retrospective,» chez *System of Systems Engineering (SoSE), 2013 8th International Conference on*, Maui, HI, 2-6 June 2013.
- [2] R. Marcel, Qualification d'un télémètre à balayage laser pour la robotique, Grenoble: Mémoire d'ingénieur, Conservatoire National des Arts et Métires, 2000.
- [3] H. Joachim et R. Joachim, Localization of a mobile robot by matching 3D laser-range-images and predicted sensor images, October 24-26 éd., *Proceedings of the Intelligent Vehicles*, 1994, pp. 345-350.
- [4] A. Dubrawski et B. Siemiatkowska , A method for tracking pose of a mobile robot, In *proceedings of the 1998 IEEE International*, 1998.
- [5] M. Jallouli, E. Colle et F. Chavand, Dispositif de localisation par télémétrie, vol. 5, RAPA, 1992.
- [6] G. Garcia et P. Bonnifait, 6 DOF dynamic localization of a mobile robot, *Preprints of the fifth IFAC Symposium on Robot Control*, 1997, pp. 173-178.
- [7] J. José et 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, 3-D modeling of indoor scenes by fusion of noisy range and stereo data, vol. 2, *IEEE International Conference on Robotics and Automation*, 1989, pp. 681-687.
- [9] J. Il-Kyun, Cartographie et localisation simultanée en environnements 3D à partir de stéréovision, Toulouse: Thèse de Doctorat de l'Institut National Polytechnique de Toulouse, 2004.
- [10] B. M. Claus et S. A. Claus , Optimal landmark selection for triangulation of robot position, vol. 23, *Journal of Robotics and Autonomous Systems*, 1998, pp. 277-292.
- [11] A. H. Alois, A. M. Markus et B. Essam, mobile robot Localization in a structured environment with obstacles, vol. 3, *IEEE International Conference on Robotics and Automation*, 1992, pp. 2576-2581.
- [12] M. Sylvain, Système embarqué de localisation et de perception pour un robot mobile, Ecole Polytechnique de Montréal, Cours Séminaire ELE6904, 25-27 Janvier 2004, pp. 399-408.

- [13] B. Marhic, L. Delahoche, C. Pégard et P. Vasseur, A navigation system based on an omnidirectional vision sensor, Proceeding of IROS, 1997, pp. 718-724.
- [14] D. Cyril, Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels, CREA, Éd., Thèse de Doctorat de l'Université de, Février 2002.
- [15] R. Alessandro et C. Ricardo, A robot self-localization system based on omnidirectional color images, vol. 34, E. S. Publishers, Éd., Journal of Robotics and Autonomous Systems, 2000, pp. 23-38.
- [16] é. 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, 1 éd., vol. 2, IEEE Journal of Robotics and Automation , 1986, pp. 14-23.
- [18] B. Fry et C. Reas, «Processing.org,» [En ligne]. Available: <https://www.processing.org/>. [Accès le 28 juillet 2015].
- [19] Arduino.cc, «Arduino - SoftwareSerial,» [En ligne]. Available: <https://www.arduino.cc/en/Reference/SoftwareSerial>. [Accès le 27 juillet 2015].
- [20] Arduino.cc, «Arduino - Wire,» [En ligne]. Available: <https://www.arduino.cc/en/Reference/Wire>. [Accès le 27 juillet 2015].
- [21] Arduino.cc, «Arduino - SPI,» [En ligne]. Available: <https://www.arduino.cc/en/Reference/SPI>. [Accès le 27 juillet 2015].
- [22] Arduino.cc, «Arduino - Software,» [En ligne]. Available: <https://www.arduino.cc/en/Main/Software>. [Accès le 9 9 2015].