

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

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'informatique
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etude De MASTER PROFESSIONNEL

Domaine : Science Et Technologie

Filière : Génie électrique

Spécialité : Automatique et Informatique Industrielles

Présenté par

Mr.Sofiane MEZIOUD

Mr.Hamza MEDANI

dirigé par

Mr. MELLAH.R

Thème

Conception et réalisation d'un robot mobile à base de carte arduino

Mémoire soutenu publiquement le 20 Septembre 2017 devant le jury composé de :

M r. MAIDIA

Professeur, UMMTO, Président

M r. MELLAH .R

Professeur, UMMTO, Encadreur

M r. ALI-BEY.M

MCB, UMMTO, Co-encadreur

M r. MOULOUDJ.K

Ingénieur, école-technique Tizi ouzou, Co-encadreur

Remerciements

*Avant tout, nous remercions le bon **Dieu** le tout puissant de nous avoir donné le courage, la volonté et la patience durant toutes les années d'études et que grâce à lui ce travail a pu être réalisé.*

*Nos vifs remerciements vont en premier lieu, à nos chers **parents et familles** de nous avoir aidé, encouragés et soutenus tout au long de ces années et qui continuent de croire en nous en dépit de tout.*

*Comme nous tenons à exprimer notre profonde gratitude à notre promoteur **Mr. Mellah .R** qui a cru en nous, qui nous a soutenus et orienter tout au long de notre travail et pour tous les moyens mis à notre disposition pour l'élaboration de ce mémoire.*

Qu'il trouve ici l'expression de notre reconnaissance.

Nous souhaitons également exprimer notre profonde gratitude et nos vifs remerciements à

***Mr. MOULOUDJ .K** formateur et enseignant à l'école technique de Tizi- Ouzou, qui n'a jamais cessé de nous suivre et orienter tout au long de notre travail, ainsi que pour ses conseils judicieux. On a pu bénéficier à la fois de ses compétences scientifiques, et de sa grande disponibilité, tant pour résoudre les difficultés rencontrées lors de notre réalisation, de répondre à nos questions.*

*Nous remercions d'avance, les **membres de jury** qui nous font l'honneur de juger notre travail.*

Nous espérons être à la hauteur de leurs attentes.

*Nous sommes aussi très reconnaissantes à tous les **enseignants** qui ont veillé au bon déroulement de notre formation tout au long de notre cursus.*

Qu'ils trouvent ici l'expression de notre respect et remerciements les plus sincères.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que ce modeste travail voit le jour.

Liste des figures

Chapitre I : Généralités Sur La Robotique

Figure 1.1 : Architecture d'un robot mobile.....	4
Figure 1.2 : navigation avec détection des obstacles.....	6
Figure 1.3 : chaîne de détection de l'environnement.....	7
Figure 1.4 : Les principaux types de roues des robots mobiles.....	7
Figure1.5 : Robot de type uni-cycle.....	8
Figure1.6 : Robot de type tricycle.....	9
Figure1.7 : Robot de type voiture.....	9
Figure1.8 : Robot de type omnidirectionnel.....	10
Figure 1.9 : image de dessous pour le châssis de robot.....	10
Figure 1.10 : Structure électrique et électronique du robot.....	11
Figure 1.11: repérage du robot.....	12

Chapitre II : Les Composants Et Les Langages De Programmation

Figure 2.1 : système de contrôle.....	15
Figure2.2:Descriptionde la carteArduinoUnoR3.....	17
Figure 2.3 : capteur ultrasonique HC-SR04.....	18
Figure 2.4 :Schémade principede fonctionnementduHC-SR04.	19
Figure 2.5 : Vue du Module Bluetooth HC-06.....	20
Figure 2.6 : Module Xbee.....	21
Figure 2.7 : différents types de modules XBee	22
Figure 2.8 : carte de puissance	23
Figure 2.9: un joystick.....	24
Figure 2.10 : un moteur à CC.....	25
Figure 2.11 : schéma d'un moteur CC.....	26
Figure 2.12 : un servomoteur.....	26
Figure 2.13 : fils d'alimentation.....	27
Figure 2.14 : fenêtre de logiciel arduino.....	28

Liste des figures

Figure 2.15 : les boutons de commande.....	29
Figure 2.16: fenêtre principale de processing.....	30
Figure 2.17: Concept d'App Inventer 2.....	31

Chapitre III : Montage Et Teste Des Composants

Figure : 3.1 : Schéma de conception du circuit.....	32
Figure 3.2 : exemple de câblage d'un module Bluetooth.....	34
Figure 3.3 : Organigramme du fonctionnement de module Bluetooth HC-06.....	35
Figure 3.4 : exemple de câblage d'un capteur ultrasonique.....	36
Figure 3.5 : Organigramme du capteur ultrason.....	37
Figure 3.6: communication des modules XBee.	38
Figure 3.7: Types de shields Xbee.....	38
Figure 3.8: Montage Xbee arduino.....	39
Figure 3.9 : Montage Xbee avec Adaptateur USB.....	39
Figure 3.10 : mode d'adressage des Xbee.....	40
Figure 3.11 Configuration des commandes AT.....	41
Figure 3.12 : schéma de L293D.....	42
Figure 3.13: La carte de puissance.....	44
Figure 3.14: exemple de câblage des moteurs avec L293D et Arduino.....	45
Figure 3.15: organigramme de la commande des moteurs.....	46
Figure 3.16: exemple de câblage de servomoteur avec l'arduino.....	47
Figure 3.17: la variation de la tension en fonction de temps.....	48
Figure 3.18 : branchement de joystick avec l'arduino.....	49
Figure 3.19 : organigramme pour le joystick.....	51
Figure 3.20: Ecran de contrôle de la commande manuelle.....	52

Liste des figures

Chapitre IV : Réalisation Pratique

Figure 4.1 : les différentes parties de robot	54
Figure 4.2 : les manettes.....	55
Figure 4.3 : Organigramme finale pour le robot avec évitement d'obstacle.	56
Figure 4.4 : les angles de navigation.....	57
Figure 4.5 : marche avant de robot.....	59
Figure 4.6 : marche à droite de robot.....	60
Figure 4.7 : marche à gauche de robot.....	61

Liste Des Tableaux :

Tableau 3.1: La logique de commande des broches de L293D.....	44
Tableau 3.2: la table de vérité de L293D.....	44
Tableau 3.3: tableau illustre les neuf positions d'un joystick	50

Abréviations :

PWM : (Pulse Width Modulation)

ICSP : est un bus informatique il permet de relier facilement un microprocesseur et différents circuits Inter-Integrated Circuit (In-Circuit Serial Programming)

E/S TOR : entrées sorties toute ou rien

UART : un émetteur-récepteur asynchrone universel (Universal Asynchronous Receiver Transmitter)

RX : réception

TX : transmission

TTL : le temps pendant lequel une information doit être conservée (*Time to Live*)

UHF : Ultra haute fréquence

VCC : l'alimentation a cc

GND : la masse

802.15.4 : protocole de communication défini par l'IEEE. Il est destiné aux réseaux sans fil de la famille des LR WPAN (Low Rate Wireless Personal Area Network)

IEEE : Institut des ingénieurs électriciens et électroniciens (L'*Institute of Electrical and Electronics Engineers*).

WPANs : Wireless Personal Area Networks.

Wifi : fidélité sans fil (*wireless* fidelity)

IDE : Environnement de développement

PCB : Un circuit imprimé (*printed circuit board*)

MIT : Massachusetts Institute of Technology.

APK : un format de collection de fichiers (*package*) pour le système d'exploitation Android.

SOMMAIRE

I.1. Introduction général	1
---------------------------------	---

Chapitre I : Généralités Sur La Robotique

I.1. Introduction	2
I.2. Définitions	2
I.3. Types de robots	2
I.4. Les robots mobiles	3
I.4.1. Cahier des charges	3
I.4.2. Définition	3
I.4.3. Classification des robots mobiles	3
I.4.4. L'architecture des robots mobiles	3
I.4.4.1. La structure mécanique et la motricité.....	4
I.4.4.2. La motricité et l'énergie	5
I.4.4.3. Les organes de sécurité	5
I.4.4.4. Traitement des informations et gestion des tâches	5
I.4.4.4.1. La navigation	6
I.4.4.4.2. La modélisation de l'environnement	6
I.4.5. Architecture globale d'un robot mobile à roue	7
I.4.5.1. Disposition des roues	7
I.4.5.2. Les robots à roues	8
I.4.5.2.1. Robot uni-cycle	8
I.4.5.2.2. Robot tricycle	8
I.4.5.2.3. Robot voiture	9
I.4.5.2.4. Robot omnidirectionnel	9
I.5. Le robot mobile uni-cycle	10

I.5.1. Conception du robot	10
I.6. Modélisation	12
I.6.1. Modèle cinématique	12
I.6.2. Modèle dynamique	13
I.7. Conclusion	14

Chapitre II : Les Composants Et Les Langages De Programmation

II.1. Introduction	15
II.2. Structure du système	15
II.2.1. Le module d'acquisition de traitement de données.....	16
II.2.2. Module de commande	16
II.3. Structure du module d'acquisition et de traitement de données	16
II. 3.1.Présentation de la carte Arduino-uno	16
II. 3.2. Présentation de capteur ultrasonique	18
II.3.3. Présentation de module Bluetooth HC-06	20
II.3.4.Le module XBee	20
II.4. Module de commande.....	23
II.4.1. Présentation de la carte de puissance(L293)	23
II.4.2. Joystick	24
II.4.3. Moteur DC	25
II.4.4. Le servomoteur	26
II.5. Choix des langages de programmations	27
II.5.1. C Arduino	27
II.5.2. Circuits.ioet Autodesk	29
II.5.3. App inventeur 2.....	30
II.6. Conclusion	31

Chapitre III : Montage Et Teste Des Composants

III.1.Introduction	32
III.2.Réalisation	32
III.2.1.Réalisationmatérielle	32
III.2.1.1. Module Bluetooth HC-06	33
III. 2.1.2. Le module ultrasonique HC-SR04.....	36
III. 2.1.3. Le module Xbee	38
III. 2.1.3. 1. Les shieldes Xbee	38
III. 2.1.3. 2. Montages	39
III. 2.1.3.3. Configuration Des Xbee	40
III. 2.1.3.4.Adressage	40
III. 2.1.3.5.Configuration Avec IDE arduino	40
III. 2.1.4. Partie commande	41
III. 2.1.4.1. Le Circuit L293D	41
III. 2.1.4.1.A. Caractéristiques Techniques Du L293D	42
III. 2.1.4.1.B. Le Câblage Du Circuit L293D	42
III. 2.1.4.1.C. La table de vérité du circuit	44
III. 2.1.4.1.D. Carte De Puissance	44
III. 2.1.4.2. Commande Des Moteurs.....	45
III. 2.1.4.2.A. Association moteur-réducteur	45
III. 2.1.4.2.B. Connexion des moteurs avec la carte arduion	45
III. 2.1.4.3. Le servomoteur	47
III. 2.1.4.3.A. Connexion d'un servo-moteur à une carte Arduino	47
III. 2.1.4.3.B. Commande de servo-moteur	48
III. 2.1.4.4. Joystick.....	48
III. 2.1.4.4.A. Commander avec un joystick.....	48
III. 2.1.4.4.B. Branchement de joystick	48
III. 2.1.4.4.C. Calibrage du joystick	50
III.2.2.Réalisationlogiciel	52

III.2.2.1. Application androïde	52
III.3. Conclusion	53

Chapitre IV : Réalisation Pratique

IV.1. Introduction	54
IV .2. Réalisation du robot	54
IV.3. Organigramme Finale du robot	56
IV .4.Mode de navigation	57
IV .4.1. Sélectionner le chemin	57
IV .4.2. Evitement d'Obstacles automatiquement	57
IV.5.Méthode de contrôle du robot	58
IV.5.1-Mode automatique	58
IV.5.1-Mode manuelle	58
IV.6. Les directions du robot dans l'espace	59
IV.6.A.Marche avant	59
IV.6.B.Marche à droite	60
IV.6.C.Marche à gauche	61
IV.7.Conclusion	62
IV.7.Conclusion Générale	63

Introduction :

La Robotique fait partie des sciences des objets et des systèmes artificiels. Elle peut être vue comme la science de la perception et du mouvement et de leur intégration en une machine physique, mécanique et informatique.

Un robot est donc un système matériel possédant des capacités de perception, d'action, de décision et de communication, parfois capable d'améliorer ses propres performances par apprentissage automatique ou supervisé par des humains[28].

La robotique comporte deux grands pôles d'intérêt: la robotique de manipulation (robotique industrielle) et la robotique mobile. Un des problèmes majeurs de la robotique mobile est la planification de mouvement. Autour de ce problème de planification de mouvement de nombreuses études ont été réalisées dans le but de développer des méthodes générales pour guider les robots.

L'objectif de notre travail est l'utilisation de la carte Arduino-uno, pour réaliser un robot mobile qui va être commandé à distance à travers des interfaces de communication xbee ou une application androïde (Bluetooth), ou bien commandé dans un espace libre de manière autonome, toute en assurant l'évitement d'obstacles qui rencontre dans son parcours de navigation.[44]

Ce mémoire est organisé en quatre chapitres à travers lesquels, nous décrivons le travail effectué pour la conception et la réalisation de notre robot.

Dans le premier chapitre nous présentons le monde de la robotique mobile et les spécificités du cahier de charges sur lequel nous nous sommes basés pour la réalisation de notre projet.

Le second chapitre décrit la phase de conception du système. On y décrit les principaux composants constituant notre robot.

Le troisième chapitre présente les étapes de réalisation du robot. On y découvre les schémas électriques détaillés de différentes parties du robot, ses modes de fonctionnement ainsi que la mise en place de la partie logicielle et les résultats obtenus.

Le quatrième chapitre fait l'objet de la description appropriée à la réalisation pratique permettant au robot naviguer conformément au cahier des charges, prédéfini.

Nous terminons notre travail par une conclusion générale et quelques perspectives.

I.1. Introduction :

La robotique peut être définie comme étant l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles. Elle se subdivise en deux types : les robots industriels et les robots mobiles. Les robots industriels sont généralement fixes, ils sont utilisés dans des nombreuses applications industrielles: l'assemblage mécanique, la soudure, la peinture... Quant aux robots mobiles ne sont pas fixes, ils sont classifiés selon la locomotion en robots marcheurs, à roues, à chenilles... comme ils peuvent être classifiés selon le domaine d'application en robots militaires, de laboratoire, industriels et de services.

Les robots mobiles présentent un cas particulier en robotique. Leur intérêt réside dans leur mobilité, destinés à remplir des tâches pénibles (exemple : transport de charges lourdes) et ils travaillent même en ambiance hostile (nucléaire, marine, spatiale, lutte contre l'incendie, surveillance...). L'aspect particulier de la mobilité impose une complexité technologique (capteurs, motricité, énergie).

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 même dans un environnement peu connu ou totalement inconnu. [44]

I.2. Définitions [45]::

Parmi les définitions appropriées à un robot mobile, disponibles dans la littérature. Ainsi, nous avons retenues ces définitions:

- Un robot est un système mécanique poly-articulé, entraîné par des actionneurs et commandé par un ordinateur qui est destiné à effectuer une grande variété de tâches.
- Un robot est appareil automatique qui peut effectuer des fonctions ressemblent à celles normalement effectuées par des êtres humains.
- Un robot est appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe ou modifiable.

I.3. Types de robots :

Généralement, les robots sont classés selon leur type, comme suit :

- Robots mobiles, robots explorateurs
- Robots manipulateurs, robots industriels
- Les humanoïdes

I.4. Les robots mobiles :

Notre travail, porte sur la réalisation d'un robot mobile commandé à distance par une manette, conformément à ce cahier des charges

I.4.1. Cahier des charges :

Deux cartes arduino uno pour traité des informations acquises des autres modules.

Une carte de puissance(L293D) pour contrôler les deux moteurs de robot.

Deux moteurs DC pour assurer marche (avant-arrière-gauche-droite).

Deux capteurs ultrason (HC-SR04) pour détectée des obstacles.

Un servomoteur qui va jouer le rôle de radar

Module Bluetooth (HC-06) pour une commande manuelle de robot avec une application androïde

Module xbee pour une commande manuelle de robot avec un joystick.

Les batteries (11V) pour assurer l'alimentation de robot.

I.4.2. Définition :

Un robot mobile est un véhicule doté de moyens de locomotion qui lui permettent de se déplacer. Suivant son degré d'autonomie il peut être doté de moyens de perception et de raisonnement. [47]

I.4.3. Classification des robots mobiles[47] :

La classification des robots mobiles se fait suivant plusieurs critères (degré d'autonomie système de locomotion, énergie utilisée ...).

La classification la plus intéressante, et la plus utilisée est selon leur degré d'autonomie.

- Véhicule télécommandé par un operateur qui lui impose chaque tâche élémentaire à réaliser.
- Véhicule télécommandé au sens de la tâche à réaliser. Le vehicule contrôle automatiquement ses actions.
- Véhicule semi- autonome réalisant sans l'aide de l'operateur des tâches prédéfinies.
- Véhicule autonome qui réalise des tâches semi- définies.

Les principaux problèmes particuliers liés à la conception de tels robots sont:

1. La conception mécanique liée à la mobilité.
2. La détermination de la position, de l'orientation et la détermination du chemin optimal.

I.4.4. L'architecture des robots mobiles[47] :

La figure 1.1 illustre l'architecture des robots mobiles qui est constituée généralement de quatre .éléments de base à savoir :

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

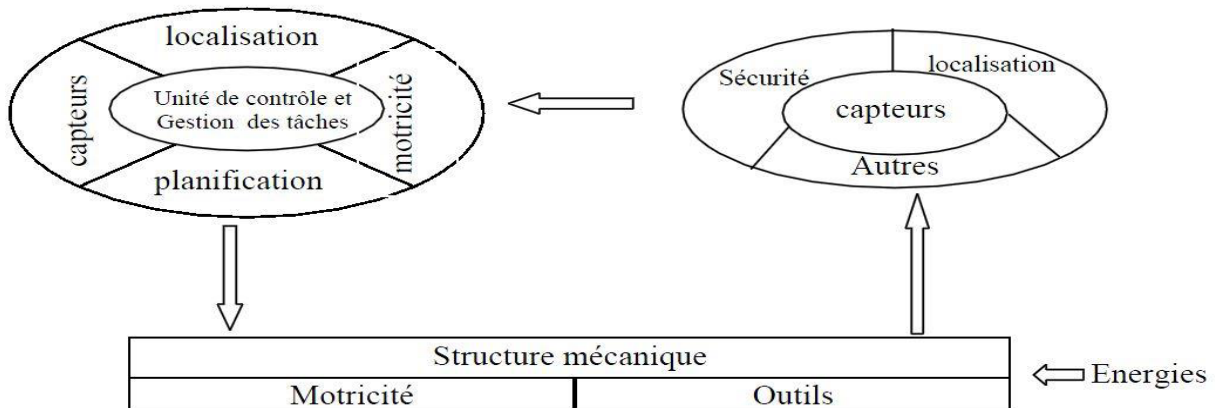


Figure 1.1 : Architecture d'un robot mobile.

I.4.4.1. La structure mécanique et la motricité :

On peut dénombrer quatre types de structures mécaniques assurant la motricité.

I.4.4.1.1. Les robots mobiles à roues :

La mobilité par roues est la structure mécanique la plus communément appliquée. Cette technique assure selon l'agencement et les dimensions des roues un déplacement dans toutes les directions avec une accélération et une vitesse importante. Le franchissement d'obstacles ou l'escalade de marches d'escalier est possible dans une certaine mesure. Toutes les configurations (nombre, agencement, fonction) des roues sont appliquées. [47]

I.4.4.1.2. Les mobiles à chenilles :

L'utilisation des chenilles présente l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence (présence de boue, herbe...). [47]

I.4.4.1.3. Les 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. [47]

I.4.4.1.4. Les robots rampants :

La reptation est une solution de locomotion pour un environnement de type tunnel qui conduit à réaliser des structures filiformes.

Le système est composé d'un ensemble de modules ayant chacun plusieurs mobilités. Les techniques utilisées découlent des méthodes de locomotion des animaux. [47]

I.4.4.2. La motricité et l'énergie :

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

L'énergie électrique la plus fréquemment employée offre l'avantage d'une commande aisée. Par contre le transport et la génération présentent des difficultés. Ainsi, plusieurs méthodes sont employées:

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

L'énergie thermique est essentiellement employée par des véhicules de forte puissance comme énergie de base pour la traction ou pour activer un compresseur hydraulique. [47]

I.4.4.3. Les organes de sécurité :

Un robot, selon la tâche qui lui est confiée, peut être amené à travailler au voisinage du personnel. A ce titre, il est obligatoire qu'il soit doté d'organes garantissant la sécurité. Des capteurs sont disponibles tout autour du mobile afin de détecter un obstacle sur un domaine le plus étendu possible. Deux types de capteurs sont employés : les capteurs proximétriques assurant la détection avant collision (ultra-son, hyper fréquence, infrarouge...) et les capteurs de contact détectant une collision ou un choc avec l'environnement (contact électrique sur pare-chocs, résistance variable, fibre optique...). Ce sont des dispositifs redondants par rapport aux capteurs précédents.

I.4.4.4. Traitement des informations et gestion des tâches :

L'ensemble de traitement des informations et gestion des tâches constitue le module information central qui établit les commandes permettant au mobile de réaliser un déplacement et d'activer les divers organes en accord avec l'objectif. Nous nous limiterons au problème de génération de plan qui consiste à établir la manière dont le robot se déplace par rapport à des connaissances à priori (statiques) ou obtenues en cours d'évolution (dynamiques). [47]

La génération de plus repose sur trois concepts :

- La stratégie de navigation
- La modélisation de l'espace
- Evitement d'obstacles

I.4.4.1. La navigation :

La navigation est une étape très importante en robotique mobile. Elle permet une large autonomie à un robot mobile. Le système de navigation comporte plusieurs modules qui peuvent être traités différemment et parmi lesquels on distingue celui de la localisation et celui de l'évitement d'obstacles. La détection et l'évitement des obstacles sont des étapes fondamentales de l'évolution d'un robot en territoire inconnu. On dispose à cet effet de plusieurs types de capteurs: caméras; un programme d'analyse des images étant alors nécessaire, capteurs laser, capteurs infrarouge et capteurs à ultrasons. [47]

On utilise en général un capteur à ultrasons qui permet d'avoir des informations à propos de la présence d'un obstacle sur le chemin d'évolution. Une fois les obstacles repérés, le robot peut effectuer plusieurs actions, par exemple : cartographier le site sur lequel il évolue, vérifier si sa distance à l'obstacle est supérieure ou non à une distance limite, et dans le cas contraire, éviter l'obstacle. La figure 1.2 illustre un exemple de navigation avec détection des obstacles.

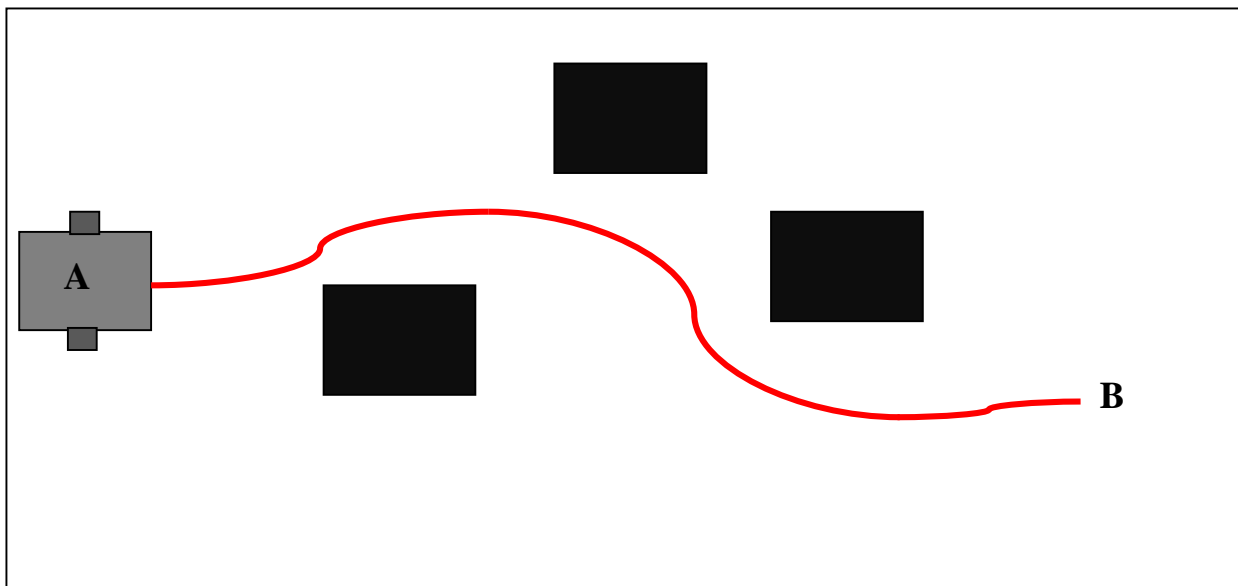


Figure 1.2 : navigation avec détection des obstacles

I.4.4.2. La modélisation de l'environnement :

La connaissance du milieu dans lequel évolue le robot mobile n'est établie en général qu'après avoir effectué une campagne de mesures appropriées à l'ensemble des éléments

constituant l'environnement. Cette procédure est difficile à réaliser, mais elle peut être évitée si le robot construit lui-même son modèle d'environnement de manière dynamique. Par contre, la planification de trajectoire n'est pas utilisable tant que le robot ne dispose pas d'un modèle de l'espace d'évolution ce qui handicape très fortement l'utilisation du robot. A partir de cette base d'informations et d'une loi évaluant les erreurs de représentation, le planificateur peut générer des sous-trajectoires faisables dans certaines parties et par la suite modifier les sous-trajectoires dans d'autres parties à l'aide des informations locales issues des mesures effectuées par les capteurs d'environnement. Lors de l'exécution d'une trajectoire, le robot acquiert des informations qui vont permettre de reconstituer le plus fidèlement possible le type de l'environnement de manière récursive à l'aide d'un algorithme approprié. [47] La figure 1.3, représente un modèle typique d'environnement :

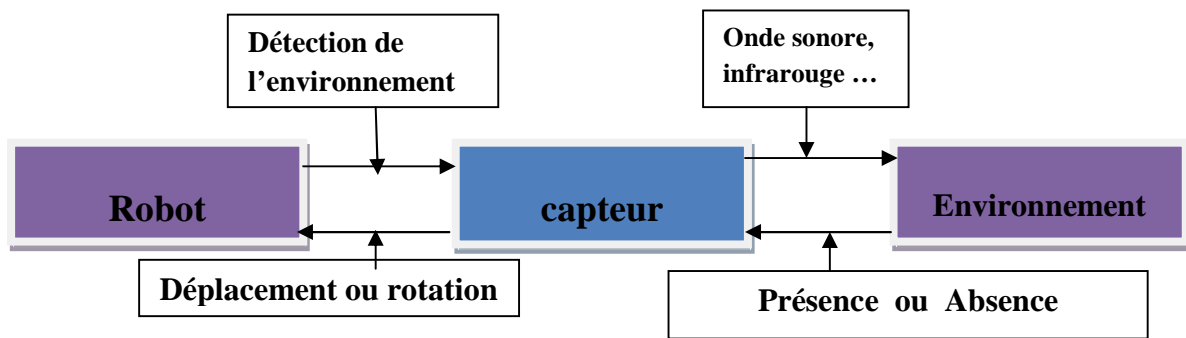


Figure 1.3 : chaîne de détection de l'environnement

I.4.5. Architecture globale d'un robot mobile à roue :

I.4.5.1. Disposition des roues :

C'est la combinaison du choix des roues et de leur disposition qui confère à un robot son mode de locomotion propre. Sur les robots mobiles, on rencontre principalement trois types de roues

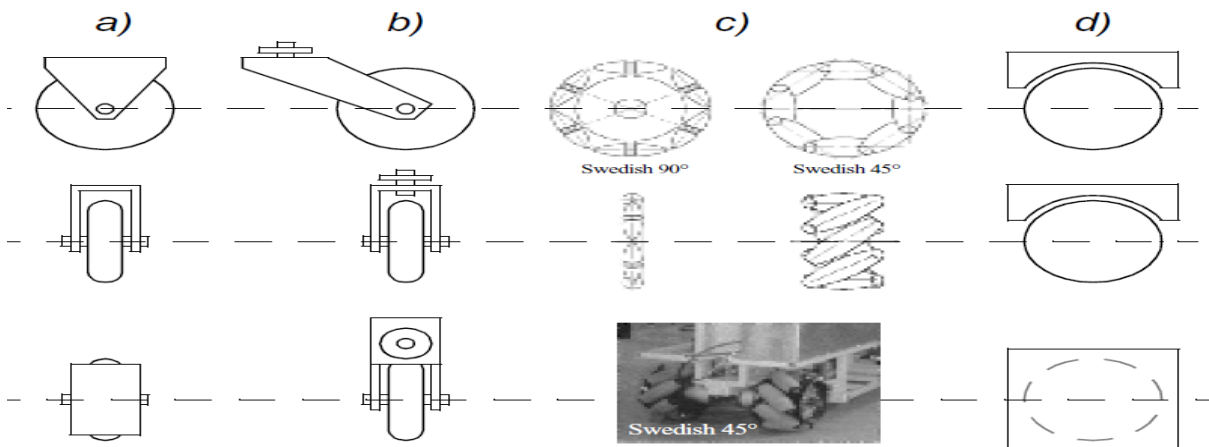


Figure 1.4 : Les principaux types de roues des robots mobiles

Les quatre types de roues de base [38] :

- A. Roue standard : Deux degrés de liberté; Rotation autour de l'essieu (motorisé) et du point de contact.
- B. Roue de roulette: Deux degrés de liberté; Rotation autour d'un joint de direction décalé.
- C. Roue suédoise: Trois degrés de liberté; Rotation autour de l'essieu (motorisé), autour des rouleaux et autour du point de contact.
- D. Roulette à billes ou sphérique: La réalisation techniquement difficile.

I.4.5.2. Les robots à roues :

Il existe 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.

I.4.5.2.1. Robot uni-cycle :

Un robot de type uni-cycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

C'est un robot non-holonome, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites, comme le montre la figure suivante. [46]

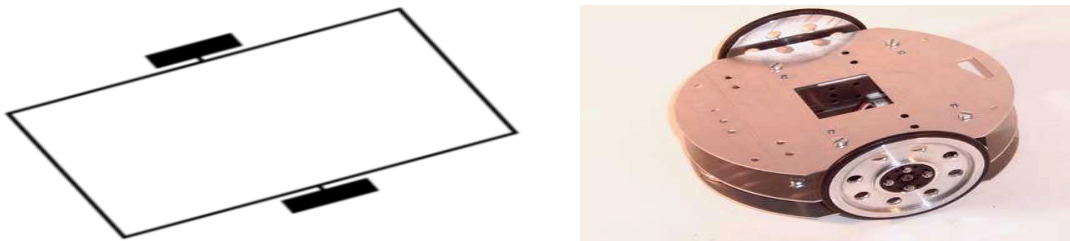


Figure1.5 : Robot de type uni-cycle

I.4.5.2.2. Robot tricycle :

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

C'est un robot non-holonome. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable. [46]

La figure 1.6 représente un prototype d'un robot tricycle.

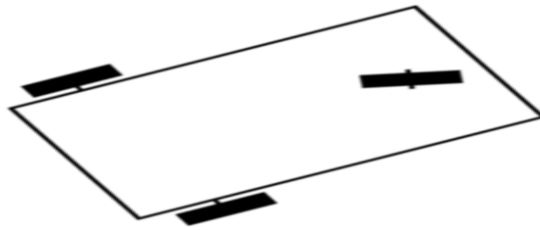


Figure 1.6: Robot de type tricycle

I.4.5.2.3. Robot voiture :

Un robot de type voiture illustré par la figure 1.7 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.

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. [46]

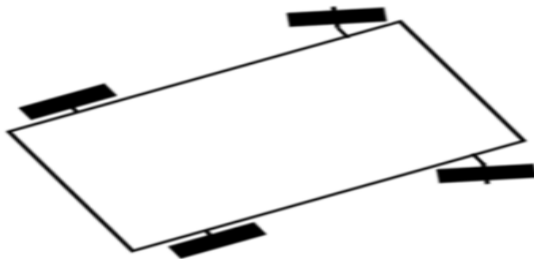


Figure1.7: Robot de type voiture

I.4.5.2.4. Robot omnidirectionnel [46]:

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

Ce type de robot présente l'avantage, le fait qu'il est holonome ce qui lui permet le déplacement dans toutes les directions. Néanmoins, ceci se fait au dépend d'une complexité mécanique bien plus grande. La figure 1.8 représente un prototype de robot omnidirectionnel.

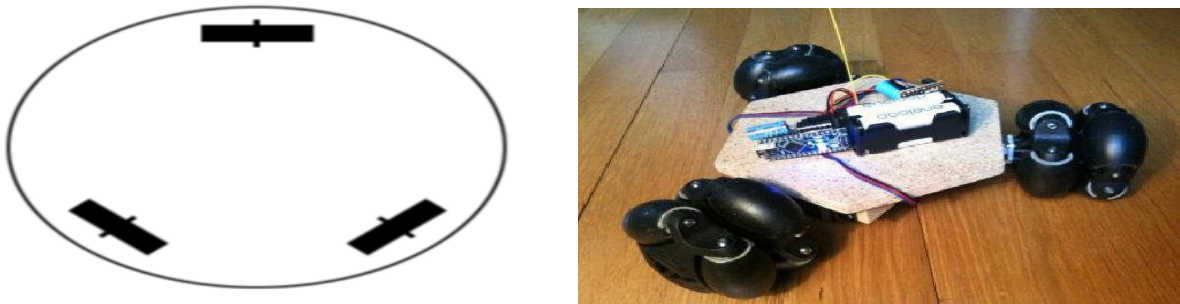


Figure 1.8 : Robot de type omnidirectionnel

Note : En robotique, une plateforme est dite holonome lorsque que le nombre de degrés de libertés contrôlables est égal au nombre total de degrés de liberté.

I.5. Le robot mobile uni-cycle :

I.5.1. Conception du robot :

La conception du robot mobile est tributaire du choix de la structure mécanique adéquate entraînée par une partie électrique qui gère à son tour par un logiciel informatique.

I.5.1.1. Conception mécanique :

De nombreuses configurations sont possibles pour un robot mobile muni de roues. Par conséquent, le robot uni-cycle, il est actionné par deux moteurs indépendants dont chacun est couplé avec une roue et par une bille libre assurant la stabilité.

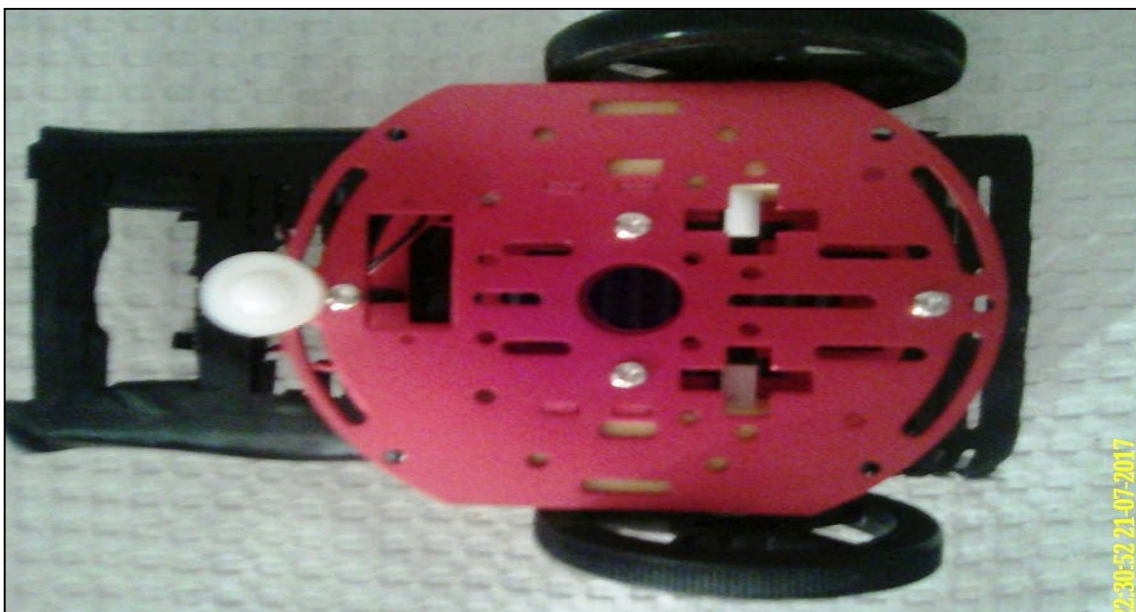


Figure 1.9: image de dessous pour le châssis de robot.

I.5.1.2. Conception électrique et électronique :

La conception électrique et électronique doit débiter avec le choix de l'architecture. Le choix de l'architecture du système électrique et électronique constitue en fait l'identification de toutes les composantes qui formeront le système. Nous pouvons diviser cette architecture en deux sections, soit l'alimentation et l'électronique de contrôle. La figure 1.10 donne le schéma synoptique de la structure de commande de notre robot mobile à base de la carte ARDUINO UNO.

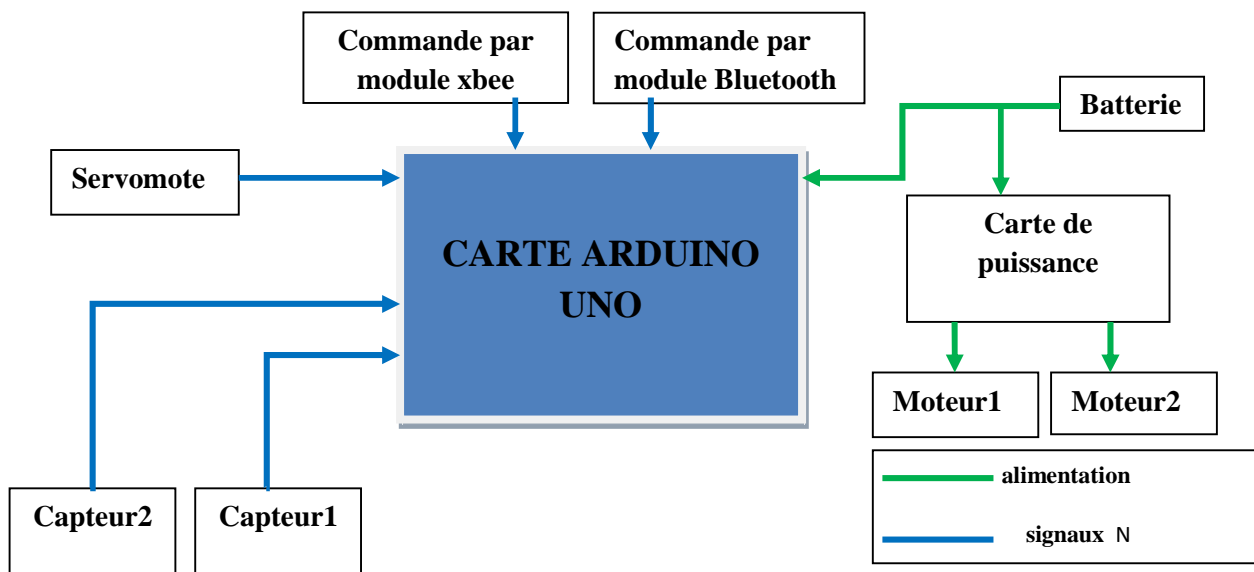


Figure 1.10 : Structure électrique et électronique du robot

I.5.1.2.1. Le contrôle électronique :

Le contrôle correspond en quelque sorte au système nerveux du robot. Elle doit être constituée dans un premier temps d'une unité de traitement informatique (microcontrôleur ou ordinateur embarqué) qui permet d'implanter les différents algorithmes désirés. Ensuite, un système permettant l'asservissement des moteurs doit être utilisé.

Dans le cas où nous désirons également utiliser différents capteurs sur le robot, il est nécessaire d'envisager une interface entre ces capteurs et l'unité de calcul. Généralement, des pattes d'entrées/sorties analogiques et/ou numériques peuvent être utilisées.

I.5.1.3. Conception informatique :

Comme, la structure de commande de notre robot est gravitée autour d'une carte arduino UNO muni d'un microcontrôleur ATMEGA328P. Ainsi, les caractéristiques désirées de la

structure logicielle du robot sont la modularité, les possibilités d'évolution, l'efficacité et la robustesse.

I.6. Modélisation :

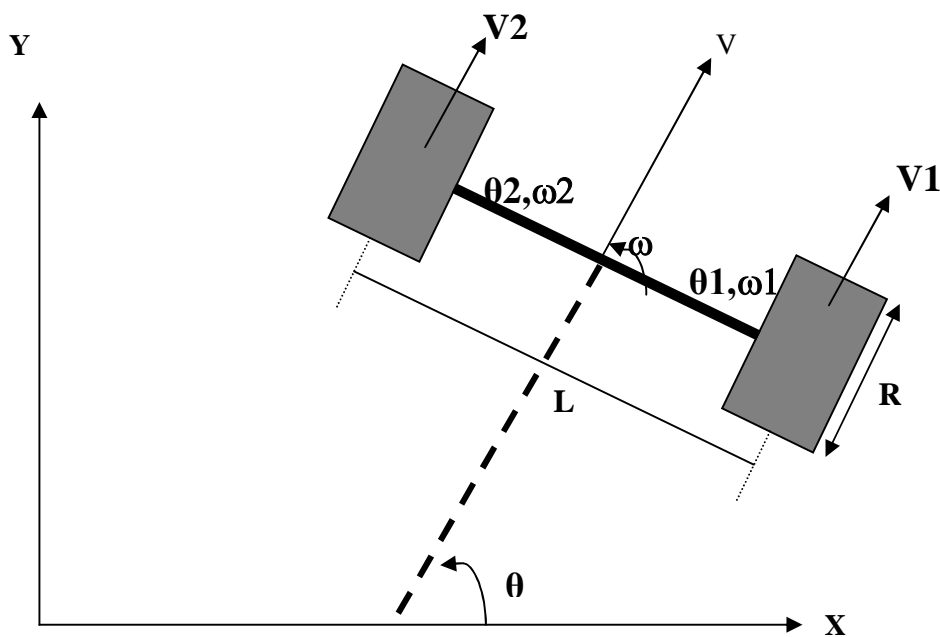


Figure 1.11: repérage du robot

I.6.1. Modèle cinématique :

La modélisation cinématique est l'étude du mouvement d'un system mécanique sans considération des forces qui influent sur son mouvement. À ce stade on s'intéresse qu'aux vecteurs de vitesse. [40]

Les équations de vitesse, rotation, et de translation [40]. :

$$\omega_1 = \frac{d\theta_1}{dt} \quad (\text{I.1})$$

$$\omega_2 = \frac{d\theta_2}{dt} \quad (\text{I.2})$$

$$V_1 = \omega_1 \cdot \frac{R}{2} \quad (\text{I.3})$$

$$V_2 = \omega_2 \cdot \frac{R}{2} \quad (\text{I.4})$$

$$V = \frac{V1+V2}{2} \quad (\text{I.5})$$

$$\omega = \frac{V1-V2}{L} \quad (\text{I.6})$$

$$\theta = \int \omega dt \quad (\text{I.7})$$

$$x = V \cdot \cos(\theta) \quad (\text{I.8})$$

$$y = V \cdot \sin(\theta) \quad (\text{I.9})$$

I.6.2. Modèle dynamique :

La dynamique est l'étude du mouvement d'un système mécanique en tenant compte les différentes forces qui affectent son mouvement. Ce modèle permet de modéliser le mouvement en translation et en rotation. [40].

I.6.2.1. Le Mouvement de Translation [40]:

Le mouvement de translation est décrit par cette relation :

$$F = dV \cdot m \quad (\text{I.10})$$

Avec :

dv : l'accélération du robot .

m : la masse du robot.

F : la force exercée sur le centre de gravité du robot.

I.6.2.2. Le Mouvement de Rotation [40]:

Le formalisme mathématique approprié au mouvement de rotation est décrit comme suit :

$$N = I \cdot d\omega \quad (\text{I.11})$$

N: les couples qui produisent la rotation autour le centre de gravité.

dω: l'accélération angulaire du robot.

I = $\frac{1}{2} m \left(\frac{L}{2}\right)^2$: le moment d'inertie de rotation du robot.

I.7. Conclusion :

Pour répondre à ce cahier des charges, le système de contrôle du robot sera divisé principalement en deux modules :

- 1- Le module d'acquisition et de traitement de donnée.
- 2- Les modules de contrôle.

Avant de montrer les détails de notre réalisation, nous présentons dans le chapitre suivant les composants et les langages de programmation que l'on utilisera pour la conception de notre robot mobile.

II.1. Introduction :

Afin de mettre notre robot en marche, il est nécessaire d'étudier la structure de notre robot, pour pouvoir traiter, contrôler les différents composants (moteurs, carte de puissance et joystick...). Donc dans ce chapitre nous allons présenter la conception du système et le fonctionnement de chaque composant qu'il contient.

II.2. Structure du système :

Le robot que nous avons conçu est constitué essentiellement de deux modules essentiels dont chacun accomplit une ou plusieurs fonctions qui lui sont propres. Le premier module représente l'acquisition de données et de mesures. Quant au second module gère la commande adéquate au mouvement souhaité du robot. On commence par le premier module qui est réalisé à base d'une carte ArduinoUno, du moment que c'est lui qui s'occupe en premier lieu de l'acquisition de données et de mesure des différents capteurs, puis on passe au second module. La figure 2.1 représente le schéma synoptique du système de contrôle.

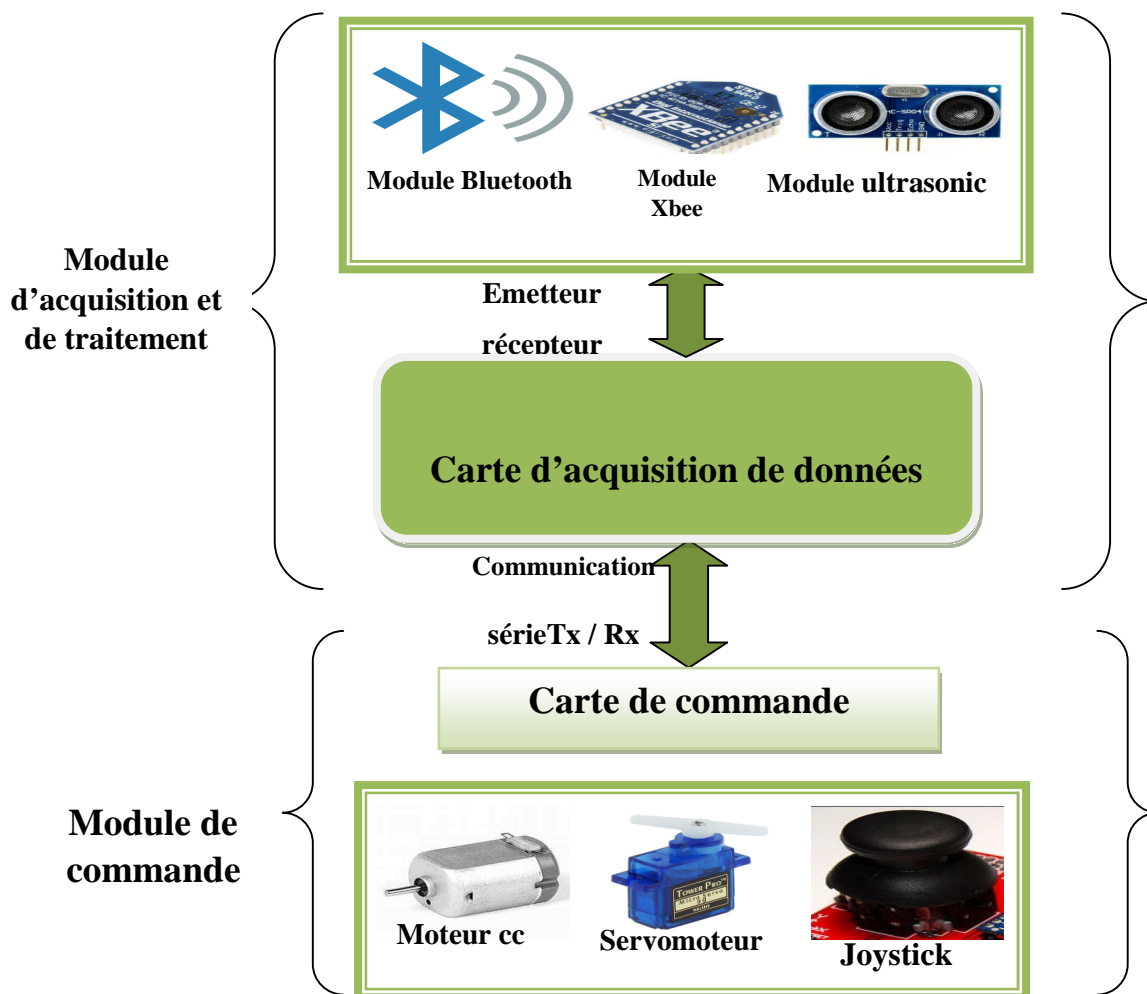


Figure 2.1 : système de contrôle

II.2.1. Le module d'acquisition de traitement de données :

C'est la partie hardware du système à base de la carte ArduinoUno, qui prend en charge la réception et l'émission des différentes informations entre les deux modules : c'est lui qui gère et maintient le fonctionnement du circuit ainsi que tous les capteurs y sont connectés.

II.2.2. Module de commande :

Il assure la commande des moteurs à travers une carte de puissance. Il a pour rôle aussi de communiquer avec carte arduino pour recevoir les instructions de fonctionnement.

II.3. Structure du module d'acquisition et de traitement de données :

Pour la réalisation de cette partie du projet, nous avons opté pour les composants suivants :

- Une carte Arduino-uno.
- Deux capteurs ultrasoniques.
- Module Bluetooth HC-06.
- Module xbee.

II. 3.1. Présentation de la carte Arduino-uno :

Le modèle de la carte unode la société ARDUINO dont le schéma synoptique décrit par la figure 2.1, est une carte électronique à base d'un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur ATmega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C. Cette carte possède 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme étant des sorties PWM (Pulse Width Modulation), 6 entrées analogiques avec un convertisseur Analogique/Numérique de 10 bits de résolution, 1 résonateur céramique (quartz) de 16 Mhz, 1 connecteur ICSP (In-Circuit Serial Programming) qui permet la programmation du microcontrôleur sur le circuit sans avoir à l'enlever, 1 connecteur jack pour une alimentation extérieure, un bouton de reset pour mettre le processus à zéro.

L'avantage de cette carte c'est qu'elle n'a pas besoin de piloteur pour faire la conversion FTDI USB/Série, elle a juste un petit microcontrôleur ATmega 16U2 programmé comme convertisseur USB/Série.

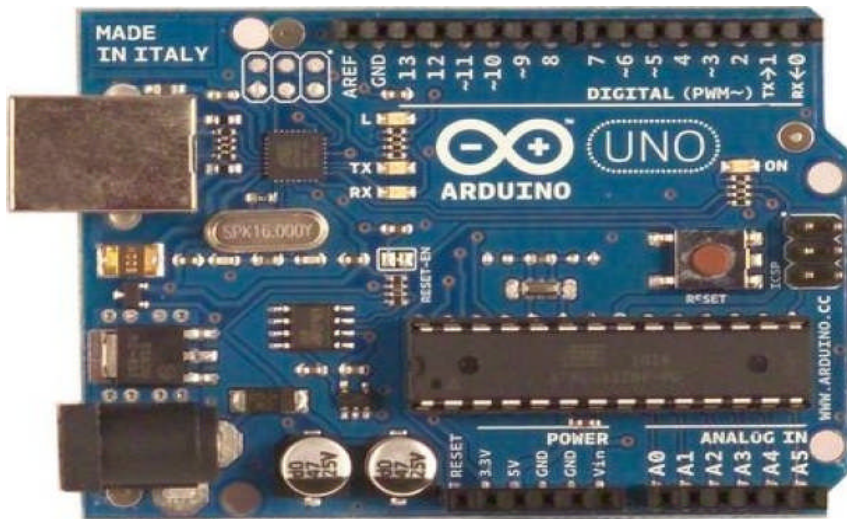


Figure 2.2: Description de la carte Arduino Uno

L'intérêt principal des cartes Arduino est leur facilité de mise en œuvre. Arduino fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus I2C, exploitation de servomoteurs ...etc. [48]

II. 3.1.1. Alimentation de carte arduino :

La carte Arduino Uno peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte. L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus).

II. 3.1.2. Arduino et la communication avec l'extérieur :

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs.

L'ATmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter) ou émetteur-récepteur universel asynchrone, pour la communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX).

II. 3.2. Présentation de capteur ultrasonique :

Le capteur HC-SR04 utilise les ondes sonores pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter. [49]

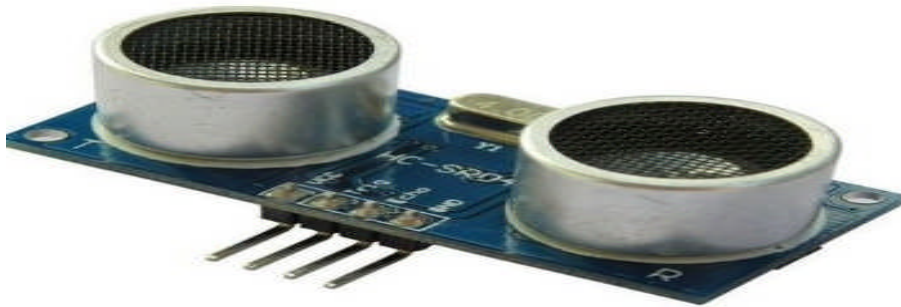


Figure 2.3: capteur ultrasonique HC-SR04

Le choix du capteur ultrasonique est un module pour arduino HC-SR04 qui est composé de deux capteurs ultrason : l'un pour l'émission des ondes ultrasons, l'autre pour la réception de ces mêmes ondes réfléchies par un obstacle

Caractéristiques :

- Alimentation = 5v CC.
- Dimensions : 45 mm x 20 mm x 15 mm
- Plage de mesure : 2 cm à 400 cm
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : 15 °
- Largeur d'impulsion sur l'entrée de déclenchement : 10 μ s (Trigger Input Pulse width)

II. 3.2.1. Principe de Fonctionnement :

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

Distance de l'objet :

La distance parcourue par un son se calcule en multipliant la vitesse du son, environ 340 m/s (ou 34 000 cm/1 000 000 μ s) par le temps de propagation, soit :

$$d = v \cdot t \text{ (distance = vitesse \cdot temps)}$$

Le HC-SR04 donne une durée d'impulsion en dizaines de μ s. Il faut donc multiplier la valeur obtenue par 10 μ s pour obtenir le temps t. On sait aussi que le son fait un aller-retour. La distance vaut donc la moitié.

$$d = 34'000 \text{ cm}/1'000'000 \mu\text{s} \cdot 10\mu\text{s} \cdot \text{valeur} / 2$$

en simplifiant :

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

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

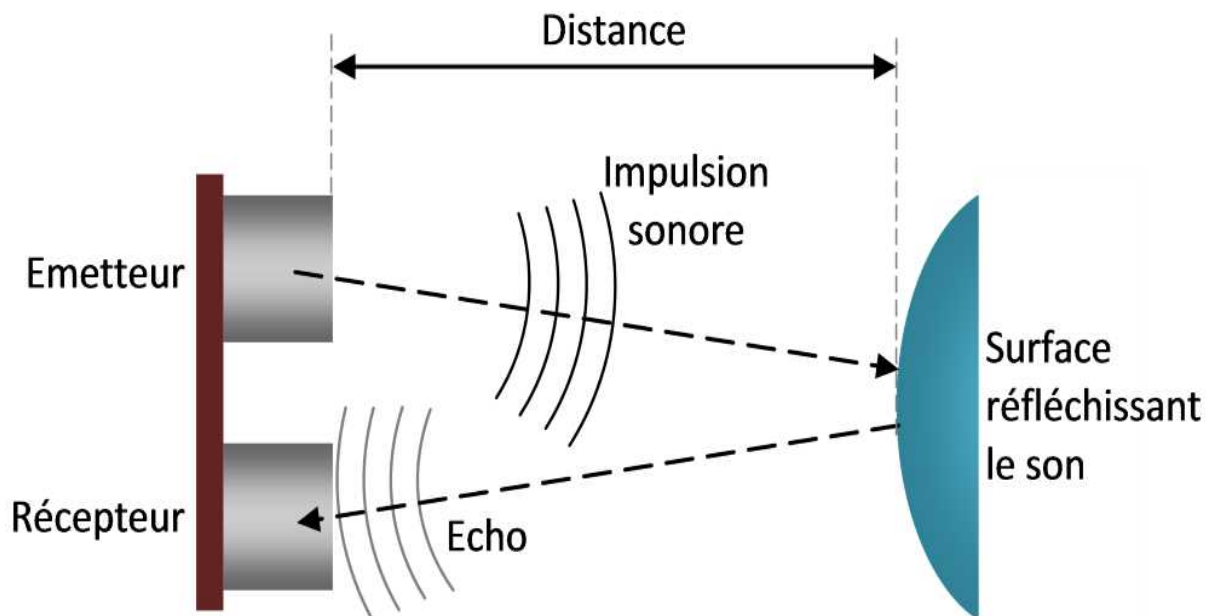


Figure2.4: Schéma de principe de fonctionnement du HC-SR04.

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

La distance est calculée, en utilisant la formule suivante :

$$\text{distance} = (\text{durée de l'impulsion (en } \mu\text{s)}) / 58$$

II.3.3. Présentation de module Bluetooth HC-06:

Bluetooth est un standard de communication permettant l'échange bidirectionnel de données à très courte distance en utilisant des ondes radio UHF sur une bande de fréquence de 2,4 GHz. Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires. Elle peut remplacer par exemple les câbles.

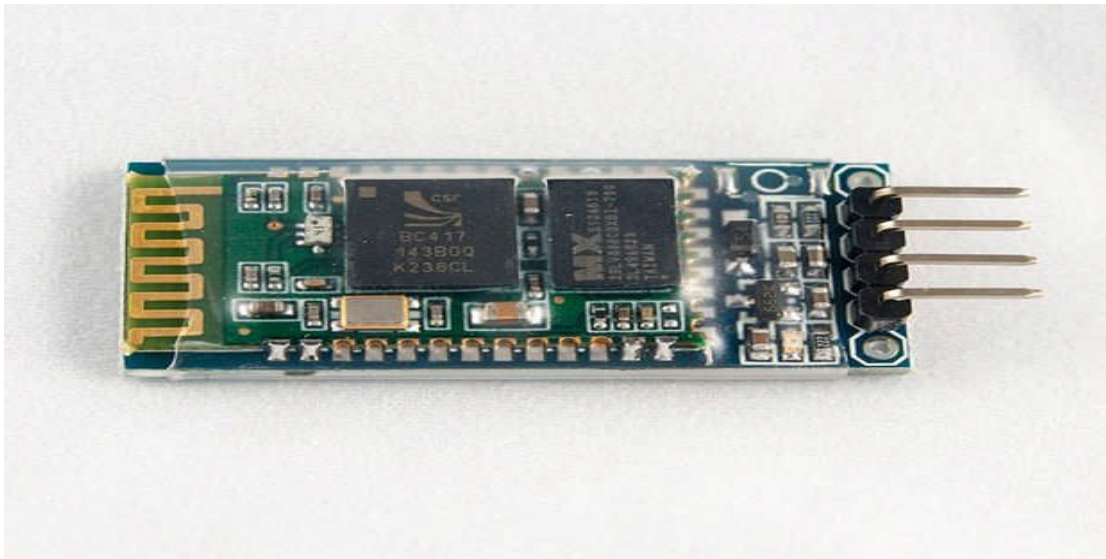


Figure 2.5 : Vue du Module Bluetooth HC-06

Le HC-06 agit comme un port série à travers lequel on peut envoyer et recevoir des données. Donc en utilisant un terminal série ou une application Bluetooth personnalisée sur l'ordinateur ou téléphone, on peut contrôler et surveiller notre robot.

Nous trouvons sur ce petit module, un ensemble de broches, VCC et GND pour l'alimentation 5V, Rx/Tx pour la communication.

II.3.4. Le module XBee :

Les XBee sont des modules de communication sans fil très populaires fabriqués par l'entreprise Digi International. Ils ont été certifiés par la communauté industrielle ZigBee Alliance. La certification Zigbee se base sur le standard IEEE 802.15.4 qui définit les fonctionnalités et spécifications des réseaux sans fil à dimension personnelle (Wireless Personal Area Networks: WPANs).

Les modules XBee utilisant les protocoles 802.15.4, permettant de réaliser différents montages, d'une liaison série RS232.

Selon les modèles, ils utilisent la bande des 2,4 GHz (comme le Wifi ou le Bluetooth). Selon leur puissance, ils émettent à une distance comprise entre 30 et 1000 mètres

Le débit peut atteindre 250 kbps, mais si on les utilise pour réaliser une liaison série sans fil, les débits standards sont compris entre 9600 bps à 38400 bps. La vitesse et la bande passante ont des effets sur les erreurs de transmission et ne sont pas possibles à obtenir dans tous les environnements.

Le module XBee est alimenté sous une tension comprise entre 2.8 et 3.4 V. Pour une alimentation de 3.3 V, la consommation est de 45 mA en émission et elle vaut 50 mA en réception.[43]

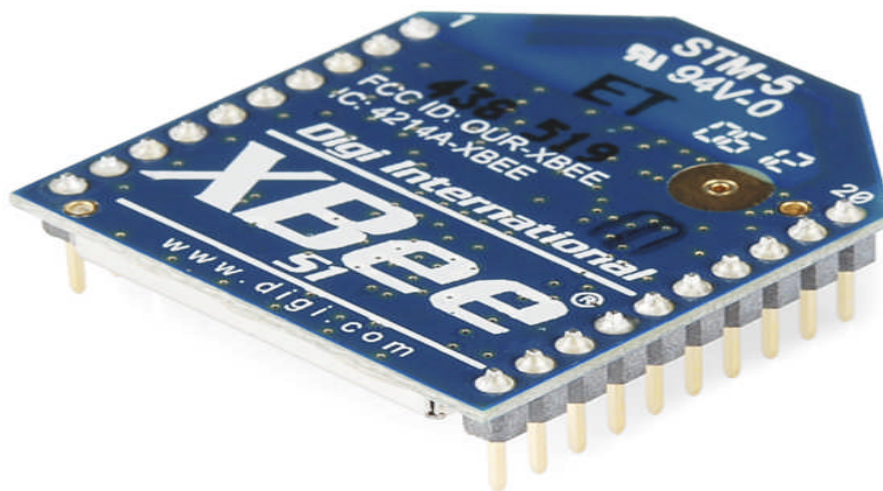


Figure 2.6 :Module Xbee

II. 3.4.1. Les modules Xbee :

Il existe 3 versions différentes de modules XBee :

- Version avec antenne Chip intégrée (XBee1)
- Version avec connecteur U.FL pour antenne externe (XBee2).
- Version avec antenne filaire intégrée (XBee3).

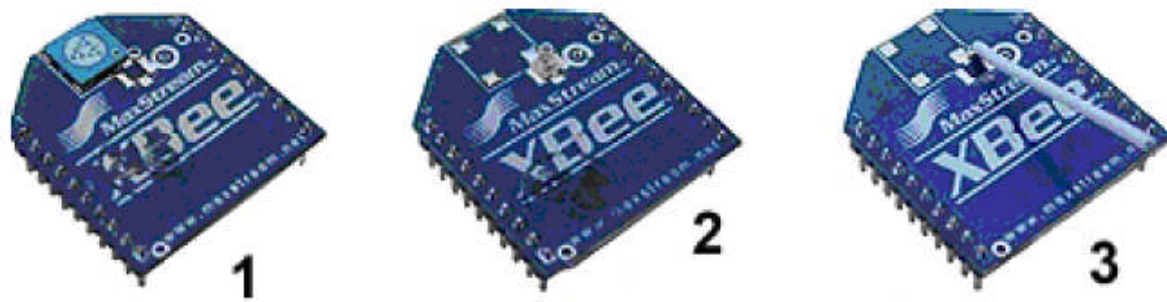


Figure 2.7 :différentstypes de modules XBee

Note:Ces modules ne sont pas compatible entre eux (on ne peut pas faire la connexion entre série1 et série2), ils ont des entrée et des sorties différentes.

II.3.4.2.Les principales caractéristiques du XBee :

- fréquence porteuse : 2.4Ghz
- portées variées : assez faible pour les XBee 1 et 2 (10 - 100m), grande pour le XBee Pro (1000m)
- faible débit : 250kbps
- faible consommation : 3.3V à 50mA
- entrées/sorties : 6 10-bit ADC input pins, 8 digital IO pins
- sécurité : communication fiable avec une clé de chiffrement de 128-bits
- simplicité d'utilisation : communication via le port série
- topologies de réseaux variées : maillé, point à point, point à multipoint

II.3.4.3. Modes de configuration :

Le XBee possède trois modes : Transparent, Command Et Api.

Le Mode Transparent : est le mode par défaut à la mise en marche du module, celui qui reçoit et envoie les données. [43]

Le Mode Command : permet de configurer le module, ses entrées, ses sorties, son adresse, l'adresse de destination de ses messages. A l'aide des commandes AT. [43]

Le Mode API :permet de commander le module distant. Evidement, le mode API est incompatible avec le mode transparent. Les commandes AT existent encore, mais elles sont intégrées dans des commandes API. [43]

II.3.4.4. Les Commande AT [42]:

ATRE : Restaure les paramètres par défaut du module.

ATDL : Modifie ou lit les 32 bits LSB de l'adressage destinataire.

ATMY : Modifie ou lit les 16 bits de l'adressage source.

ATIDxxxx : Modifie ou lit l'adresse du Pan ID. Il faut que cette valeur soit la même pour que les modules puissent communiquer entre eux.

ATCN : Pour quitter le mode commande.

II.4. Module de commande:

Pour la réalisation de cette partie du projet, nous avons opté pour les composants suivants :

- carte de puissance L293.
- Joystick.
- 2 Moteur CC.
- Servomoteur.

II.4.1. Présentation de la carte de puissance(L293) :

Le composant L293D est une double interface pour charge inductive. Il est constitué d'une partie commande et d'une partie puissance à structure de type pont en H. Il est largement utilisé pour la commande de moteurs CC ou pas à pas.



Figure 2.8 :carte de puissance

II.4.1.1. Caractéristiques principales :

Vcc, alimentation commande 4,5 à 5,5v.

Vs, alimentation puissance 5 à 36v.

Courant maximal 600 mA

II.4.1.2. Principe de fonctionnement :

Le L293 permet :

- de contrôler un moteur dans un sens.
- de contrôler un moteur dans l'autre sens.
- de contrôler un moteur dans les 2 sens.

II.4.2. Joystick :

Le joystick est composé de deux potentiomètres disposés à angle droit et d'un bouton poussoir qui nous permettent de régler le mouvement en 2-D (x,y). Les potentiomètres agissent comme des capteurs qui nous fournissent une tension variable en fonction de la rotation de l'appareil autour de son arbre. [34]

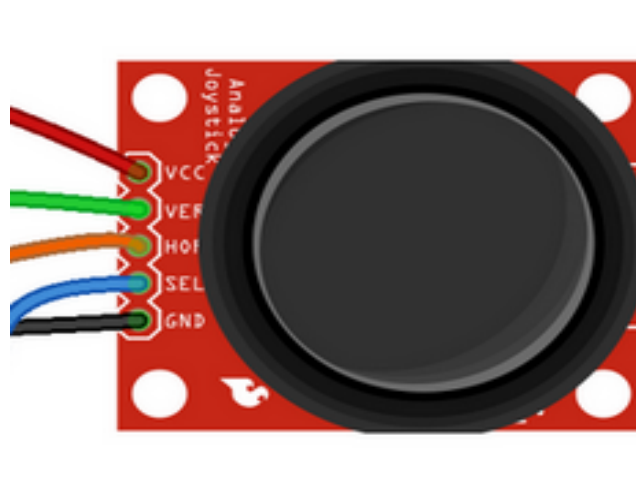


Figure2.9:joystick

VCC: à l'alimentation positive (5V ou 3.3V), **GND:** la masse.

VERT: à un PIN d'entrée analogique de l'Arduino où il indiquera la tension sur l'axe vertical. Lorsque le joystick est centré, il indique approximativement la moitié de la tension d'entrée. L'Arduino convertit en une valeur comprise entre 0 et 1023.

HORIZ: idem pour l'axe horizontal.

SEL: pour le bouton poussoir, lors de l'appui, cette sortie est reliée à la masse (GND).

II.4.3. Moteur DC :

Les moteurs à courant continu transforment l'énergie électrique en énergie mécanique de rotation. Mais ils peuvent également servir de générateur d'électricité en convertissant une énergie mécanique de rotation en énergie électrique.

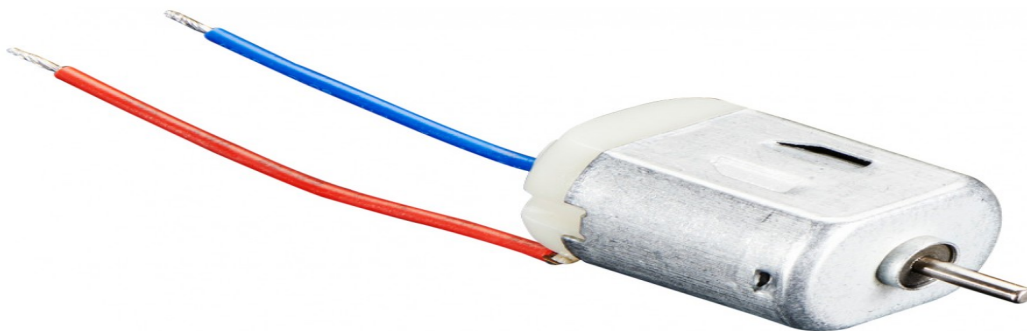


Figure 2.10 : moteur CC

II.4.3.1. Principe de fonctionnement du moteur à courant continu :

Le moteur à courant continu donné par la figure 2.11 est composé de deux parties principales : le rotor (partie qui tourne) et le stator (partie qui ne tourne pas, statique). Le rotor est composé de fils de cuivre enroulés (bobines) sur un support lui-même monté sur un axe. En alimentant ces enroulements par un courant continu, un champ magnétique sera généré de telle manière à faire tourner l'axe du moteur. Ce dernier représente l'arbre de sortie du moteur. C'est lui qui va transmettre le mouvement à l'ensemble mécanique (pignons, chaîne, actionneur...) qui lui est associé en aval.

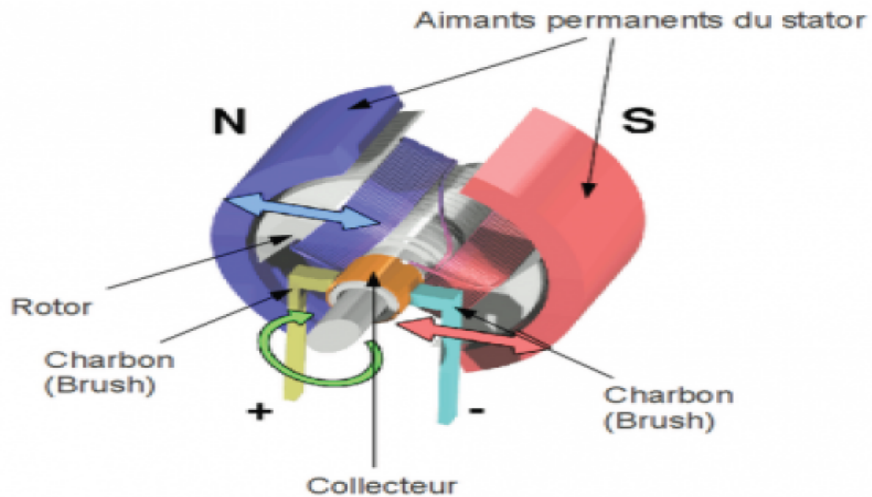


Figure 2.11 :schéma d'un moteur CC

II.4.4. Le servomoteur :

Les servomoteurs permettent de déplacer un bras, jusqu'à une certaine position (ou angle de rotation), puis à maintenir solidement cette position. Le terme même de servomoteur signifie qu'il s'agit d'un moteur asservi, obéissant à une commande externe. [34]

La figure 2.12 donne un prototype d'un servomoteur qui est un ensemble complet de mécanique et d'électronique, qui contient :

- un moteur à courant continu (CC) souvent de petite taille.
- une carte électronique d'asservissement.
- un réducteur de vitesse.
- un potentiomètre, pour contrôler la position de l'axe du moteur.
- un axe dépassant hors du boîtier avec différents bras ou roues de fixation.



Figure 2.12 : servomoteur

Le servomoteur a besoin de trois fils de connexion pour fonctionner. La figure 2.13 montre la disposition de ces trois fils, dont deux fils servent à son alimentation, et le dernier étant celui qui reçoit le signal de commande [34]:

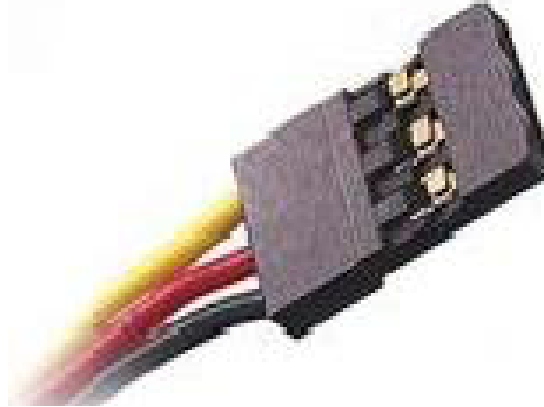


Figure 2.13 :fils d'alimentation

II.4.4.1 Fonctionnement du servomoteur :

Un servomoteur se pilote par l'intermédiaire d'un câble à trois fils. Ce câble permet à la fois de l'alimenter et de lui transmettre des consignes de position par le fil de signal.

Contrairement à un moteur CC, qui peut être piloté par des variations de tension ou par allumage/extinction, le servomoteur réagit en fonction d'une impulsion de durée variable. C'est la durée de ce signal qui détermine la rotation de l'axe donc la position de l'objet fixé dessus.

II.5. Choix des langages de programmations :

II.5.1. C Arduino :

L'IDE est une application écrite en Java inspirée du langage Processing. Il permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte arduino.

Arduino Software (IDE) contient un éditeur de texte pour écrire un code, une zone de message, une console de texte, une barre d'outils avec des boutons pour des fonctions communes et une série de menus. Il se connecte au matériel Arduino et Genuino pour télécharger des programmes et communiquer avec eux.

II.5.1.1. L'interface :

L'interface dispose plusieurs fenêtres :

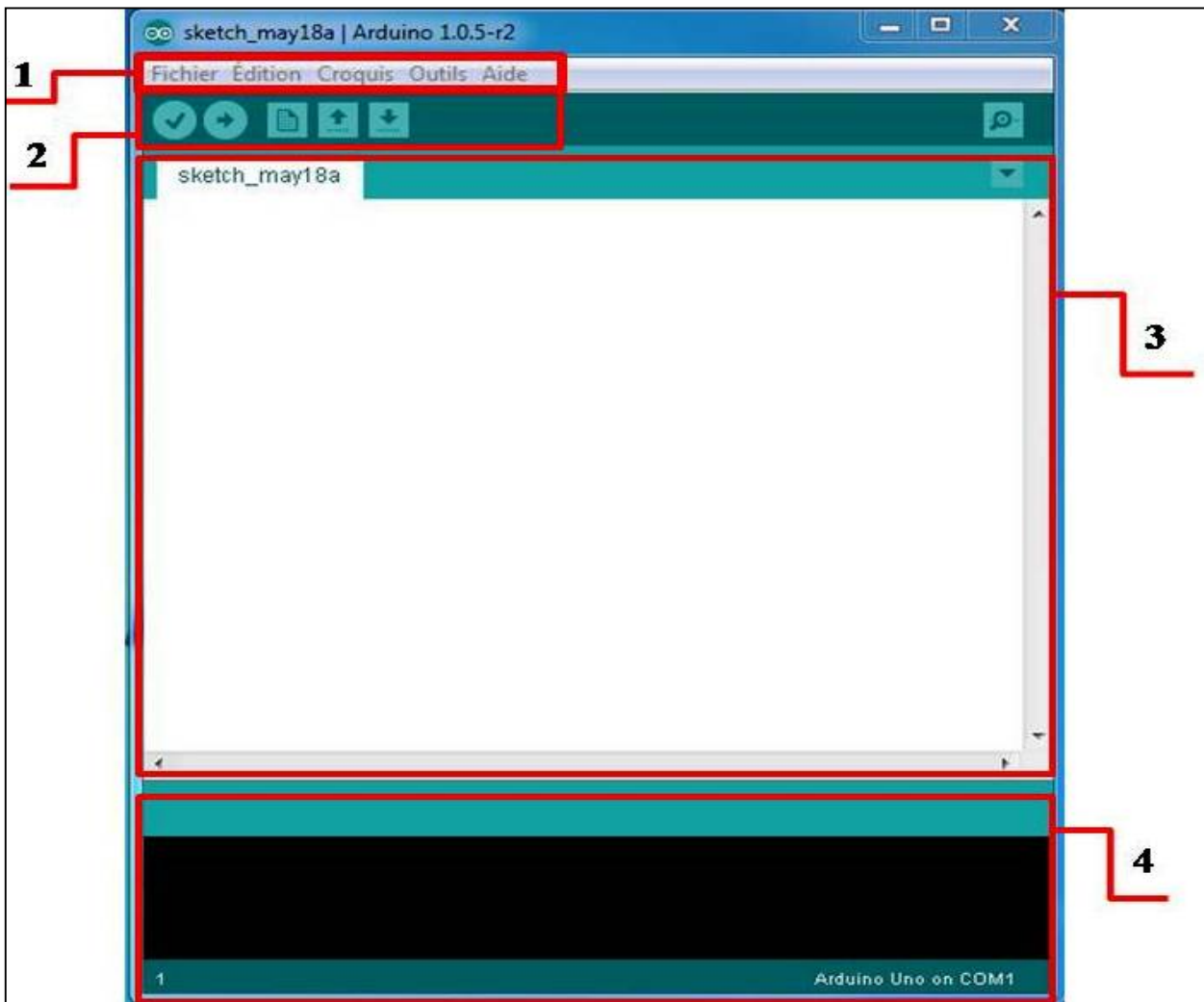


Figure 2.14 :fenêtre de logiciel arduino

1. options de configuration du logiciel.
2. boutons pour la programmation des cartes.
3. programme à créer et 4. débogueur (affichage des erreurs de programmation) .

II.5.1.2. Les boutons :

1. permet de vérifier et compiler le programme, il actionne un module qui cherche les erreurs dans le programme.
2. Compiler et envoyer le programme vers la carte.

3. Créer un nouveau fichier.

4. Charger un programme existant.5. Sauvegarder le programme en cours.6. Liaison série.

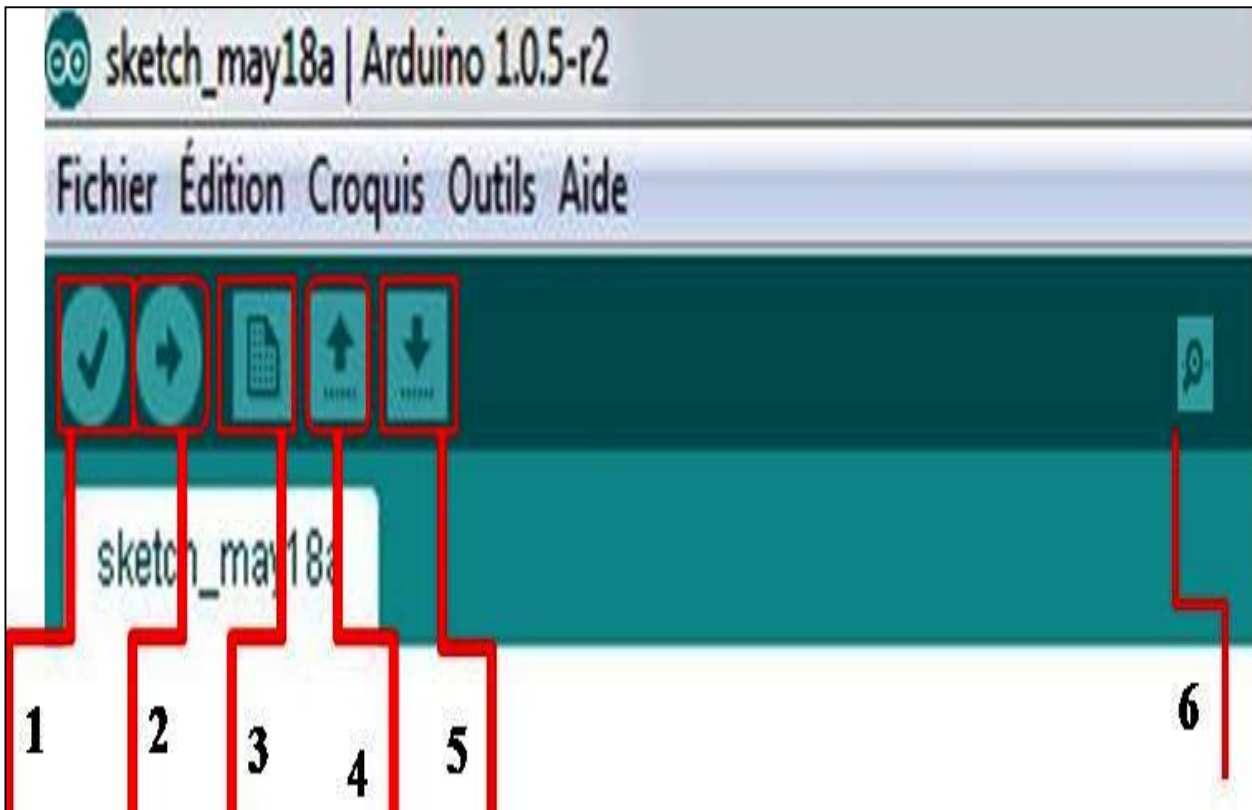


Figure 2.15 : les boutons de commande

II.5.2. Circuits.io et Autodesk:

Autodesk et Circuits.io viennent de développer un outils de conception électronique en ligne totalement gratuit. Bien évidemment, il ne va vous permettre de réaliser de grands circuits complexes, mais il est une bonne base pour concevoir des shields Arduino très simplement. [5]

Le principal avantage de cette solution est la possibilité de concevoir un montage un peu comme avec Fritzing. Il est ensuite possible de simuler son montage en y intégrant même du code arduino pour ensuite finaliser en routant son PCB.

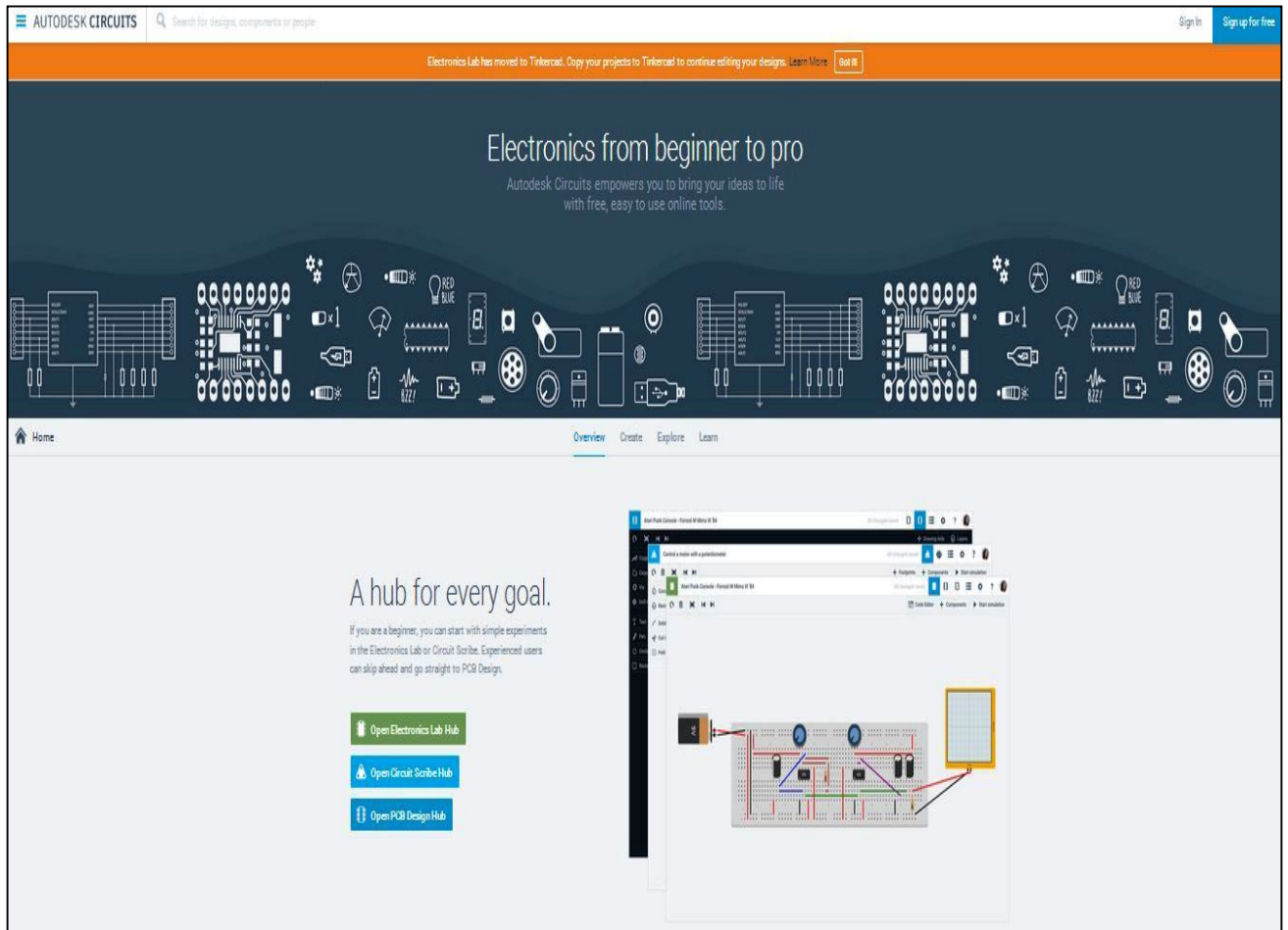


Figure 2.16: fenêtre principale de autodesk (circuit.io)

II.5.3.App inventeur 2:

C'est une application développée par Google. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT).

Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont plus familiers avec les langages de programmation. Elle est basée sur une interface graphique similaire, grâce à son interface entièrement graphique et à l'absence totale de ligne de code. Elle est particulièrement adaptée à l'initiation des enfants à la programmation, et ce des écoles primaires.

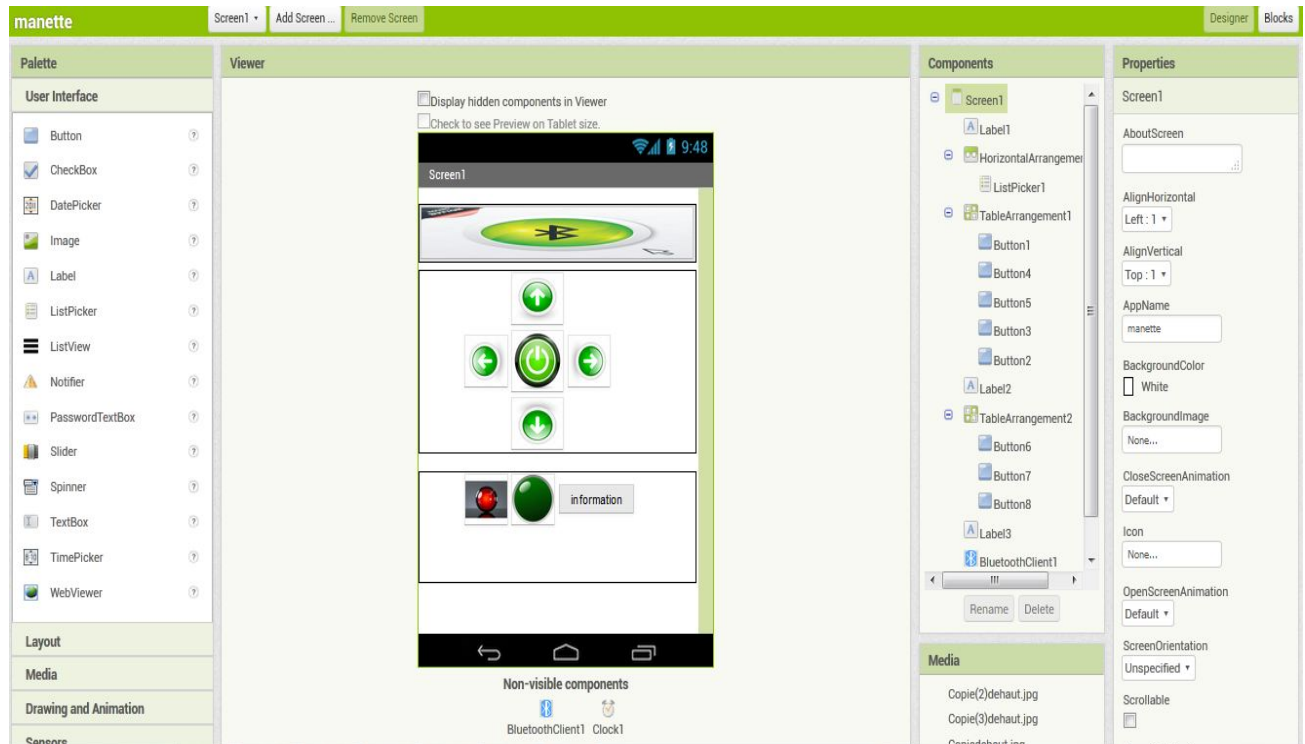


Figure 2.17: Concept d'App Inventer 2

II.6. Conclusion :

Pour que le robot navigue facilement dans son environnement. Nous avons besoin de connaître l'architecture matérielle et logiciel de notre projet. Ainsi les caractéristiques et le mode de fonctionnement des composants et les langages de programmations qui constituent notre projet.

III.1. Introduction :

Après avoir conçu le robot mobile relativement la réalisation pratique conformément au cahier des charges souhaitées, suivie de tests de fonctionnement du système font l'objet de ce chapitre. Ainsi, nous présentons ce chapitre sous la forme en deux parties à savoir la réalisation matérielle gérée par un programme à travers une interface xbee, qui représente la deuxième partie de ce chapitre.

III.2. Réalisation :

III.2.1. Réalisation matérielle :

La réalisation matérielle est faite en premier lieu, de telle manière que chaque partie est réalisée et testée séparément. Cependant, les montages sont d'abord construits sur des « *breadboard* » ou Lab d'essai. Par la suite, une fois qu'ils sont expérimentés et adoptés séparément, nous les avons regroupés et branchés sur la carte Arduino Uno dans le châssis de robots. La figure3.1 donne le schéma synoptique relatif au circuit électronique du système de contrôle.

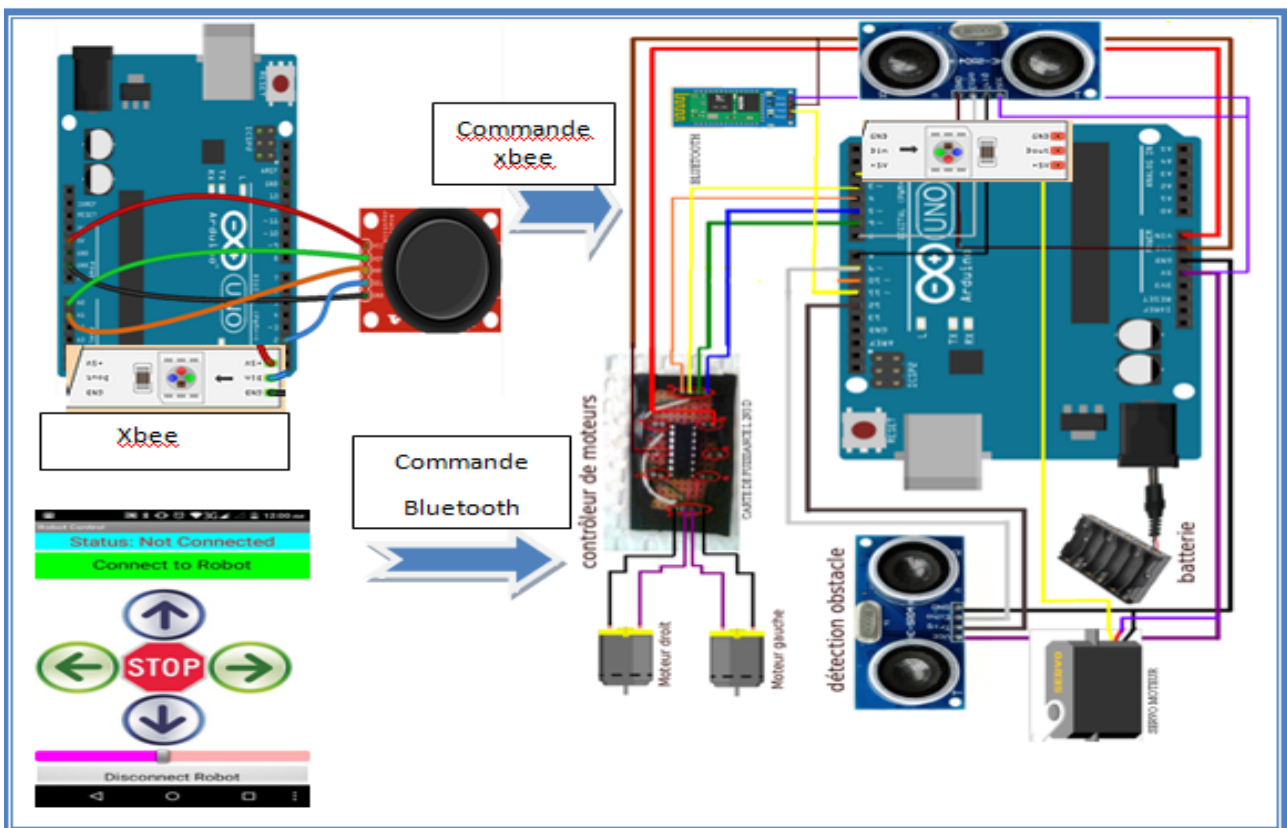


Figure 3.1: Schéma de conception du circuit

A travers ce circuit électronique, on constate que le système de contrôle est constitué essentiellement de deux parties dont la première partie qui représente la partie commande qui est composée de deux cartes Arduino UNO et des interfaces de communication à savoir : Un module Bluetooth HC-06, un module xbee, deux capteurs ultrasoniques et un joystick. Quant à la deuxième partie qui représente la partie puissance, elle est composée d'une carte de puissance L293D pour commander deux moteurs DC, un servomoteur.

Pour assurer le fonctionnement conformément au cahier des charges, la première carte arduino à base de microcontrôleur Atmega 328 cadencé par un oscillateur externe de fréquence 16 Mhz est configurée comme suit :

- Les broches Tx et Rx sont réservées pour l'XBee.
- Les broches A5 et 4 sont réservées pour commander le moteur droite de robot en avant ou en arrière.
- Les broches 5 et 6 sont réservées pour commander le moteur gauche de robot en avant ou en arrière.
- La broche 3 pour PWM pour commander la vitesse des moteurs.
- Les broches 8 et 9 sont réservées au capteur ultrason radar.
- Les broches 2 et 12 sont réservées au deuxièmes capteur ultrason.
- La broche A0 est réservées pour servomoteur.
- La broche 11 est configurée en entrées(Rx), et elle est réservée au module Bluetooth.

Quant à la deuxième carte Arduino Uno :

- Les entrées/sorties numériques ont été configurées.
- Les broches 0 et 1(Tx, Rx) sont réservées au module Xbee.
- Les broches analogiques A0 et A1 sont réservées au joystick.
- Les broches 3 et 4 sont réservées au deux leds témoin.
- La broche 2 est réservée au bouton.

III.2.1.1. Module Bluetooth HC-06 :

Le module Bluetooth HC-06 qui est à faible consommation d'énergie et facile à mettre en œuvre va nous assurer une communication unidirectionnelle avec la carte Arduino-Uno et sans fil grâce à une application installée sur téléphone portable Androïde de telle manière à configurer deux modes de contrôles comme le montre la figure suivante :

- Commande manuelle.
- Commande automatique.

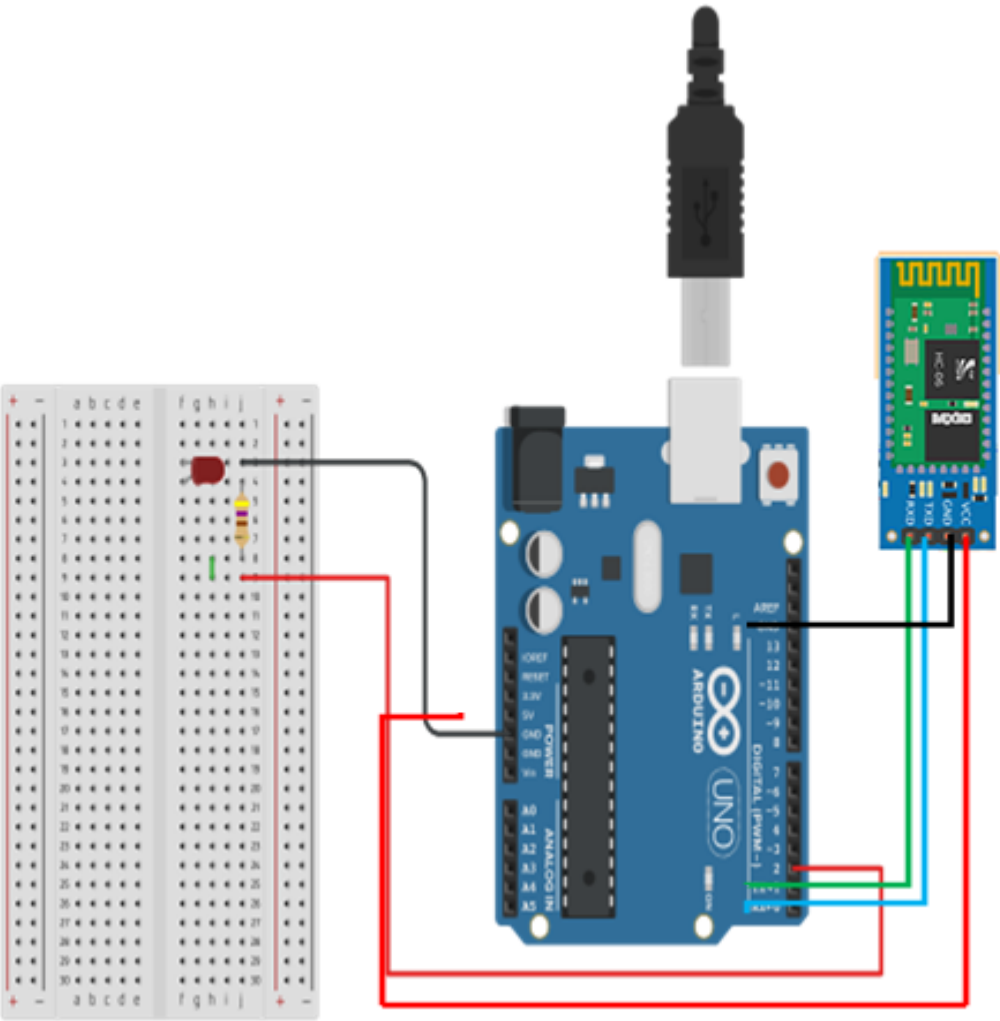


Figure 3.2 : exemple de câblage d'un module Bluetooth

En outre, l'organigramme de fonctionnement du module Bluetooth HC-06 est donné comme suit :

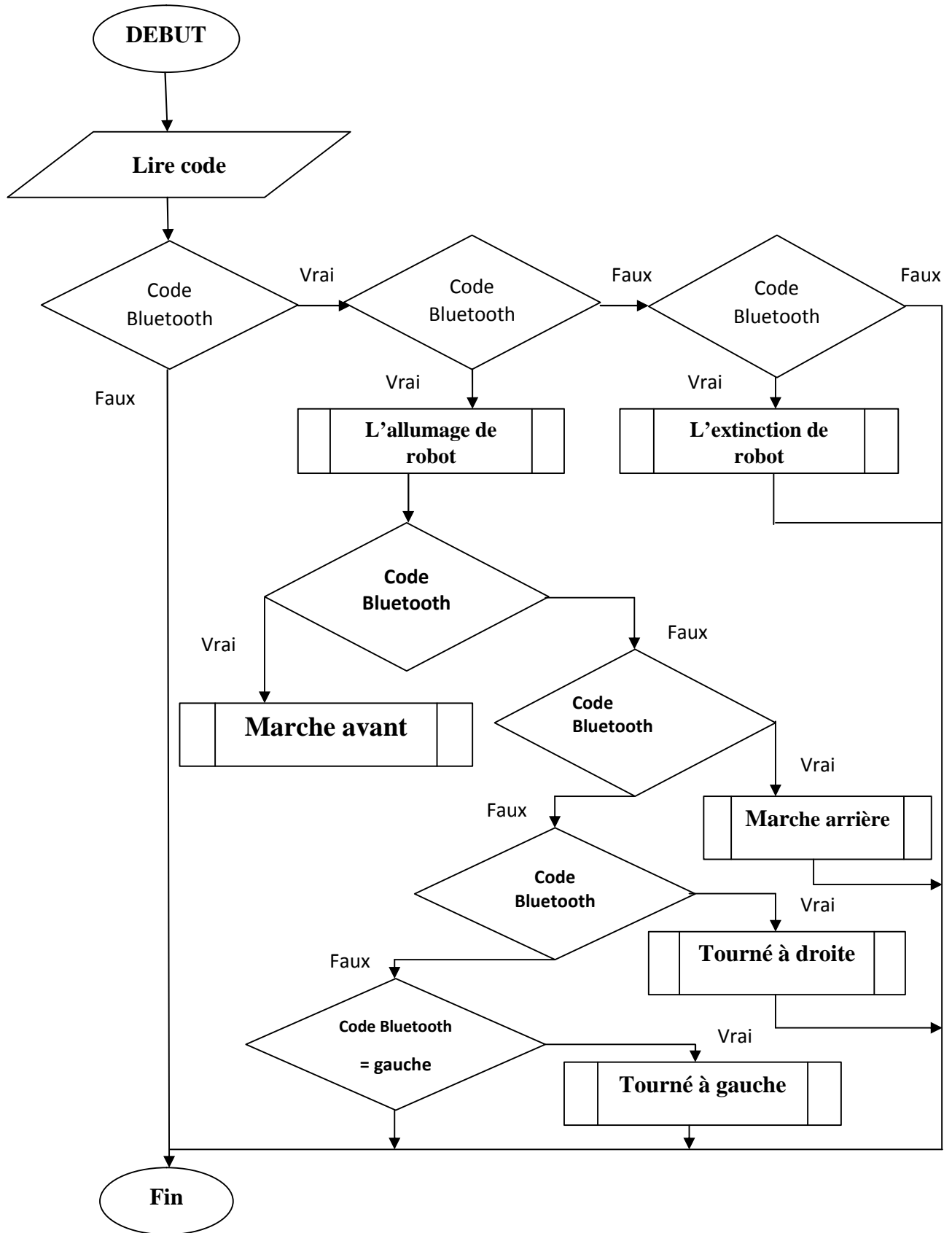


Figure 3.3 : Organigramme du fonctionnement de module Bluetooth HC-06

III. 2.1.2. Le module ultrasonique HC-SR04 :

Le principe de fonctionnement du premier capteur ultrason HC-SR04 est simple, il est expliqué en détail dans le chapitre précédant : on envoie une impulsion d'une durée de $10\mu\text{s}$ sur une broche de l'Arduino qui correspond à la patte *Trig* du capteur et on reçoit une impulsion sur une broche de l'Arduino qui correspond à la patte *Echo* du capteur, d'une durée proportionnelle à la distance qui sépare le capteur de l'obstacle. Le schéma de brochage est réalisé à l'aide du site TINKERCAD ou bien le site 123D CIRCUITS, [21] comme le montre la figure 3.4:

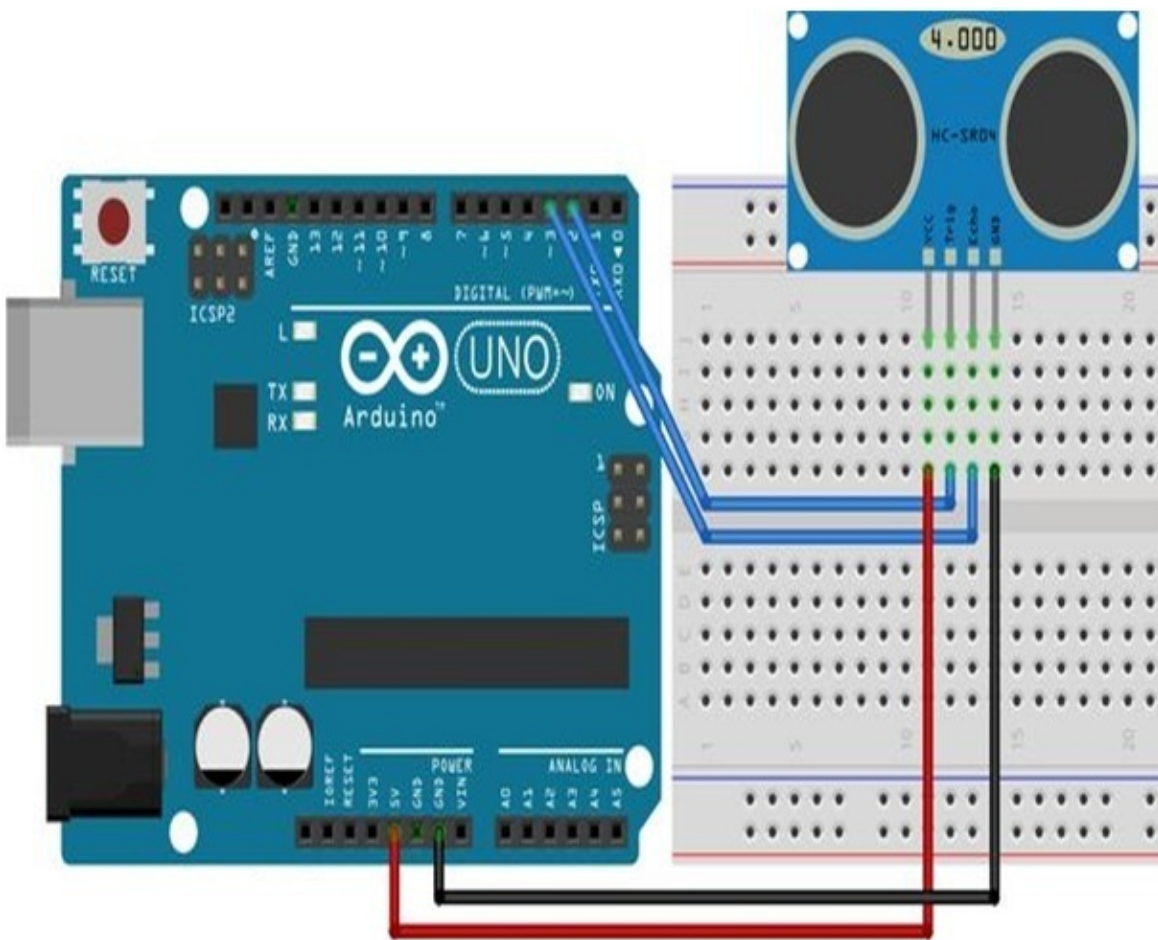


Figure 3.4 : exemple de câblage d'un capteur ultrasonique

Le deuxième capteur ultrasonique HC-SR04 est utilisé pour détecter les trous et les vides et pour protéger le robot de ne pas se détruire.

L'organigramme de fonctionnement du capteur ultrason est le suivant :

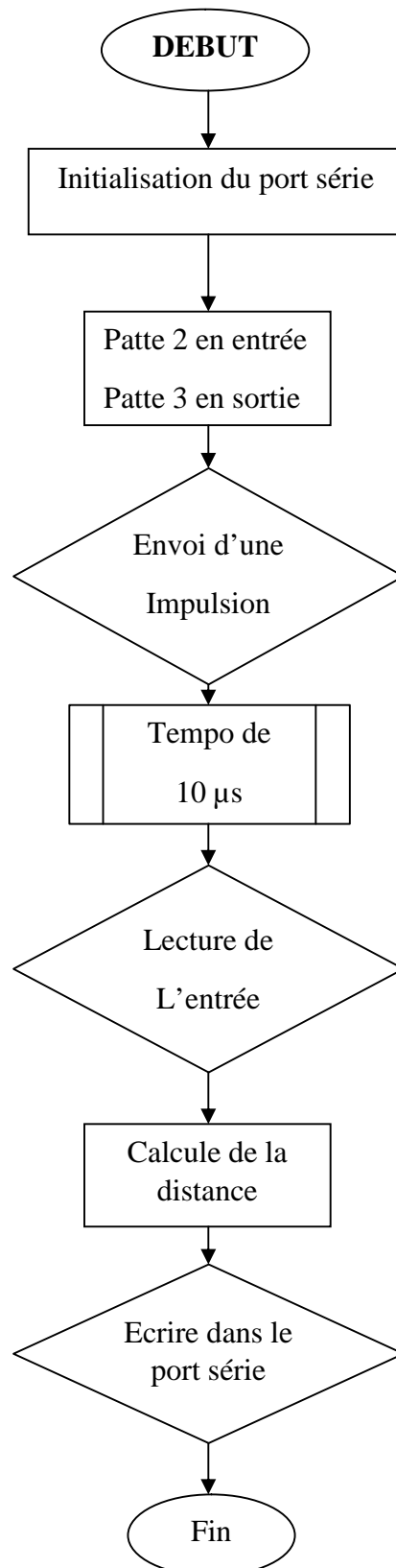


Figure 3.5 : Organigramme du capteur ultrason

III. 2.1.3. Le module Xbee :

Les modules XBee sont des circuits de communication sans-fil utilisant les protocoles 802.15.4 et Zigbee, permettant de réaliser différents montages, d'une liaison série RS232 classique à un réseau maillé (mesh). [42]

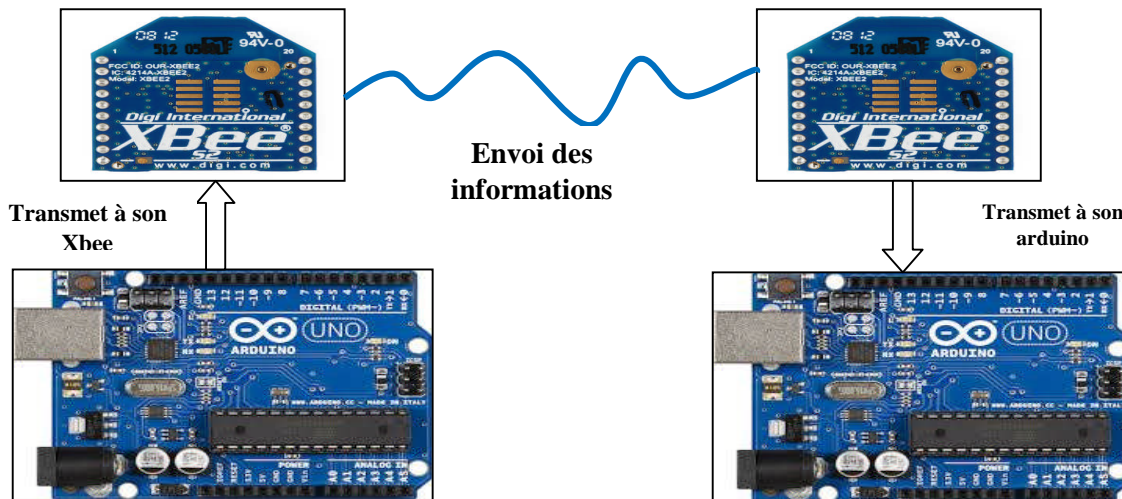


Figure 3.6: communication des modules XBee.

III. 2.1.3. 1. Les shields Xbee :

Ce sont des cartes d'extension dédiée aux modules XBee qui ajoute des leds, un convertisseur 3,3 volts et des cavaliers de configuration permettant de conserver la liaison USB-série.

shields Xbee : permet de connecter un XBee avec la carte arduino.

Adaptateur USB Xbee : permet de connecter un XBee avec l'ordinateur. Ce circuit dispose des leds indiquant la mise sous tension, la transmission / réception avec un autre module.

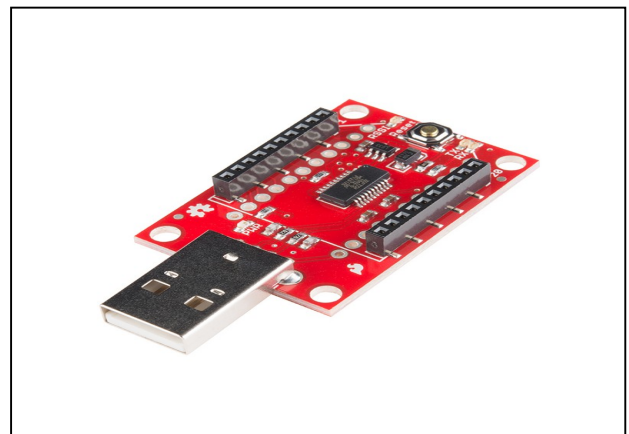
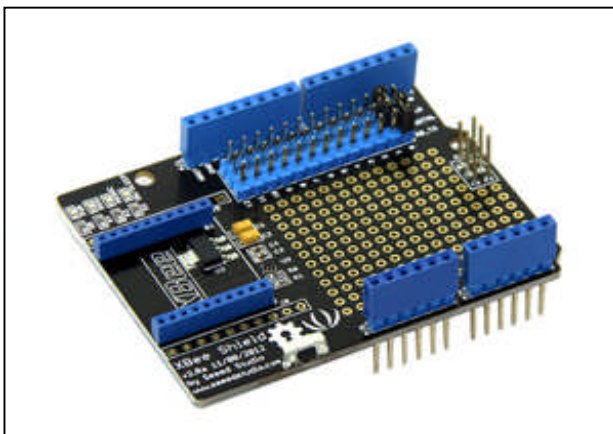


Figure 3.7: types de shields Xbee

III. 2.1.3. 2. Montages :**XBee Avec Arduino (Shieldes Xbee) :**

Tx vers **Tx** de la carte arduino.

Rx vers **Rx** de la carte arduino.

GND vers la **GND** de la carte.

Vcc vers **3,3V** de l'arduino

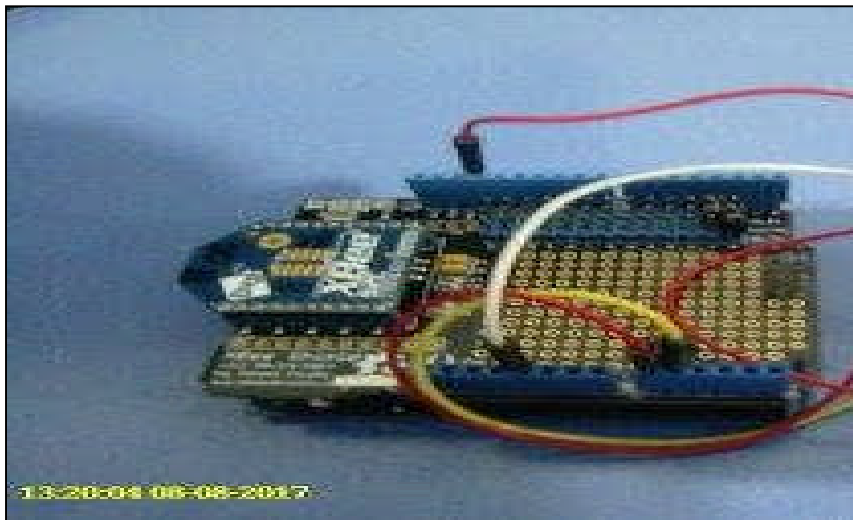


Figure 3.8: Montage Xbee arduino

XBee Avec Ordinateur (Adaptateur USB Xbee) :

Figure 3.9 : Montage Xbee avec Adaptateur USB

III. 2.1.3.3. Configuration Des Xbee :

La configuration des paramètres de la liaison RS232 (Vitesse, Start et stop) se fait avec une commande AT.[42]

III. 2.1.3.4. Adressage :

Pour tout XBee, il faut nécessairement définir l'adresse du réseau ATID, son adresse personnelle ATMY et si besoin, l'adresse de destination des paquets ATDL.[42]

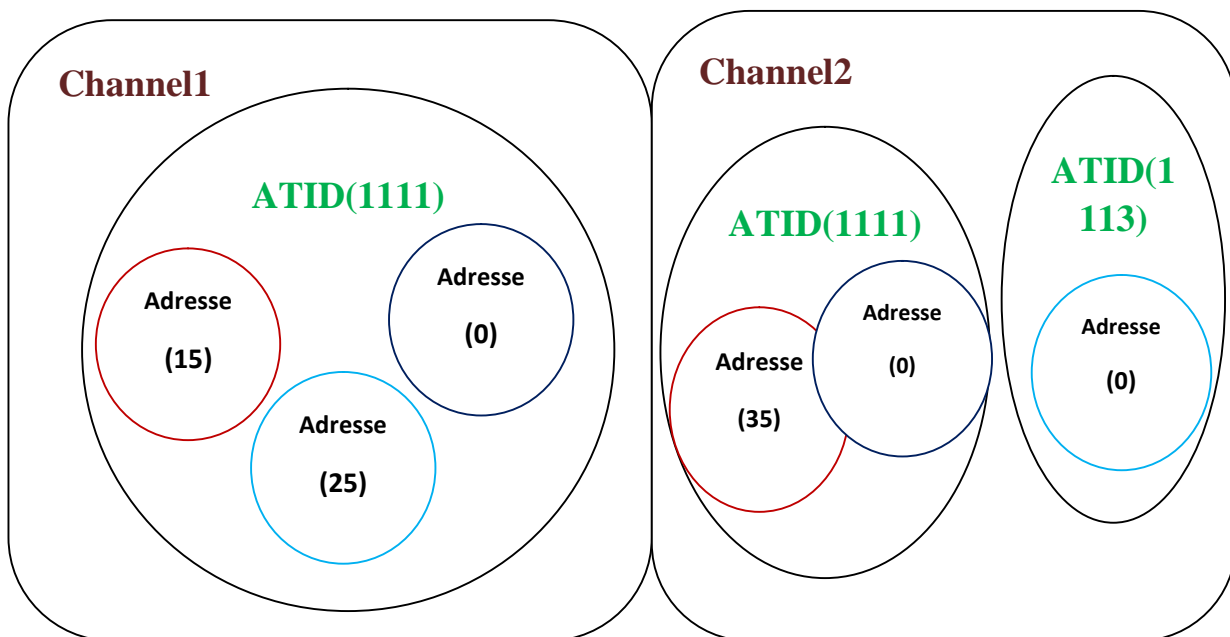


Figure 3.10 : mode d'adressage des Xbee

III. 2.1.3.5. Configuration Avec IDE arduino :

Pour pouvoir envoyer des commandes au module et recevoir ses retours, nous avons utilisé le moniteur série de IDE arduino. [25]

Si on envoi : +++ on va recevoir OK sur le moniteur. Le XBee nous signale qu'il passe en mode COMMAND et qu'il est prêt à recevoir les messages de configuration.

Si nous attendons plus de 10 secondes sans appuyer sur une touche, le XBee revient en mode TRANSPARENT. Nous devons alors retaper +++ pour revenir en mode COMMAND.[25]

Après nous allons envoyer les autres commandes l'une après l'autre :

(ATRE), (ATDL), (ATMY), (ATID), (ATCN).

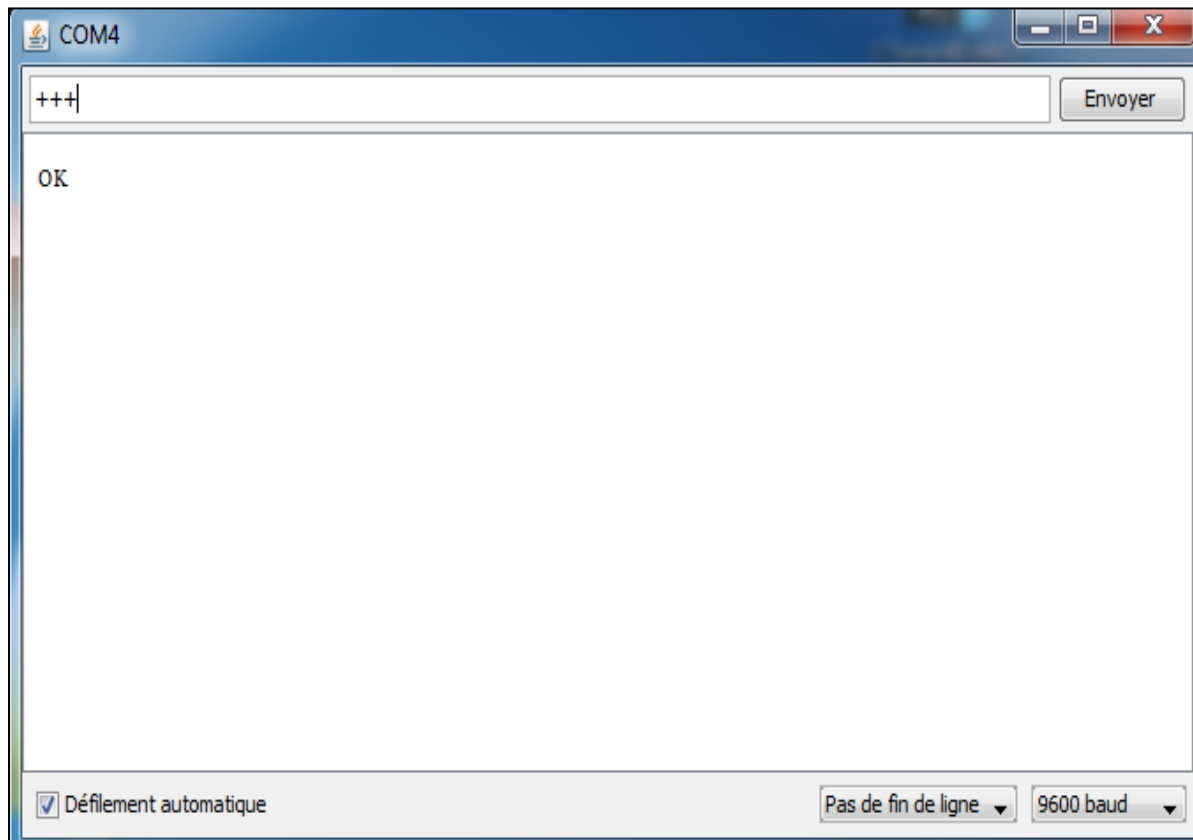


Figure 3.11: Configuration des commandes AT

III. 2.1.4. Partie commande :

Afin de voir une idée générale sur cette partie, nous allons étudier et voir en détail le branchement des composants qui constituent cette partie.

III. 2.1.4.1. Le Circuit L293D :

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens). En raccordant les sorties de façon appropriées, il est possible de constituer deux pont-H. Il est ainsi possible de commander deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre. Il intègre aussi les diodes de roue libre (le "D" derrière le L293) qui protègent le circuit.

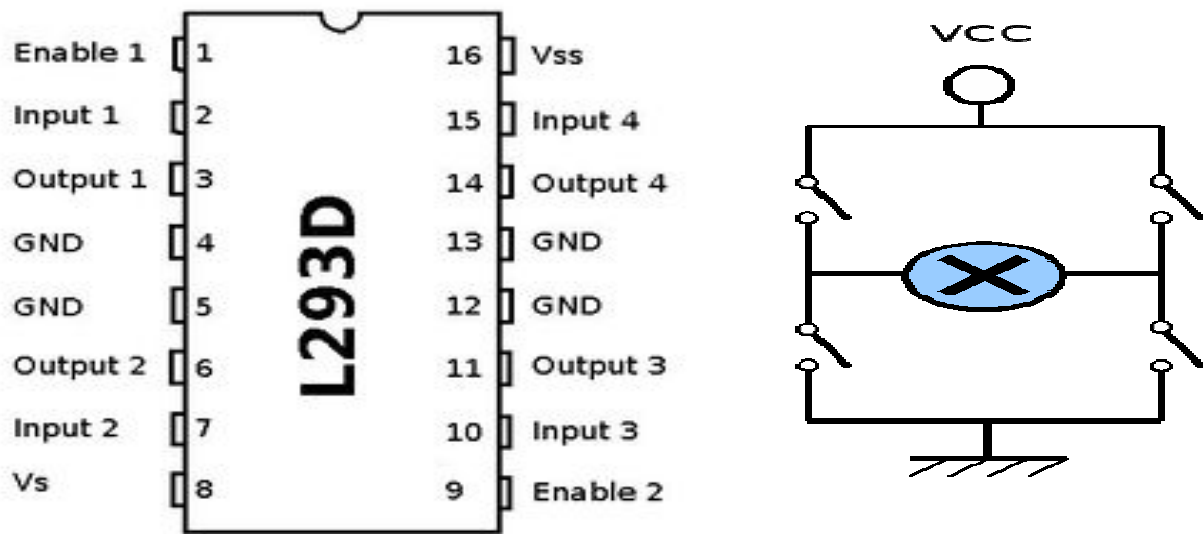


Figure 3.12 : schéma de L293D

III. 2.1.4.1.A. Caractéristiques Techniques Du L293D :

Les caractéristiques techniques du composant L293D:

- Nbre de pont-H: 2
- Courant Max Régime continu: 600mA.
- Courant de pointe Max : 1200mA.
- VS Max Alim moteur: 36v
- VSS Max Alim logique: 4,5v-5,5v.
- Nbre de Broches: 16 .
- Perte de tension: 1.3v.

III. 2.1.4.1.B. Le Câblage Du Circuit L293D :

Ci-dessous la configuration des broches du L293D et la table de la logique de commande.

Broche	Nom	Description
1	Enable1	Permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 et OUTPUT2. ENABLE1 commande l'activation/désactivation du premier Pont-H.

		<ul style="list-style-type: none"> • Si ENABLE1=GND, le pont-H est déconnecté et le moteur ne fonctionne pas. • Si ENABLE1=VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas, en fonction des tensions appliquée sur INPUT1 & INPUT2.
2	Input1	<p>Avec Input 2, sont les broches de commande du Pont-H Output1/Output2.</p> <p>Se raccorde à Arduino, permet de commander le sens du courant entre Output 1 et Output 2.</p>
3	Output1	Avec Output 2, seront les broches à raccorder à la charge (le moteur).
4	GND	<p>Doit être raccorder à la masse (GND) de la source d'alimentation de puissance VS (ex: la borne négative de l'accumulateur +9.2v) et à la masse de la source d'alimentation de la logique "VSS" (donc GND Arduino).</p> <p>Si vous n'avez qu'une source d'alimentation pour le tout, c'est forcément plus simple.</p>
5	GND	
6	Output2	Avec Output 1, seront les broches à raccorder à la charge (le moteur).
7	Input2	<p>Avec Input 1, sont les broches de commande du Pont-H Output1/Output2.</p> <p>Se raccorde à Arduino, permet de commander le sens du courant entre Output 1 et Output 2.</p>
8	Vs	Alimentation de puissance des moteurs (souvent +9.2v).
9	Enable2	Commande l'activation du second pont-H constitué de Output3/Output4.
10	Input3	A utiliser conjointement avec Input 4 pour commander le pont-H Output3/Output4.
11	Output3	Constitue une des deux sorties du second pont-H (Output3/Output4)
14	Otput4	Constitue une des deux sorties du second pont-H (Output3/Output4)
15	Input4	A utiliser conjointement avec Input 3 pour commander le pont-H Output3/Output4.

16	Vss	Alimentation de la logique de commande (5V). A raccorder à la borne +5V d'Arduino .
----	-----	--

Tableau 3.1: La logique de commande des broches de L293D.

III. 2.1.4.1.C. La table de vérité du circuit :

En	Input1(3)	Input2(4)	Fonction
HIGH	LOW	HIGH	Tourne à droite
HIGH	HIGH	LOW	Tourne à gauche
HIGH	LOW	LOW	Arrêt rapide
HIGH	HIGH	HIGH	Arrêt rapide

Tableau 3.2: la table de vérité de L293D.

III. 2.1.4.1.D. Carte De Puissance:

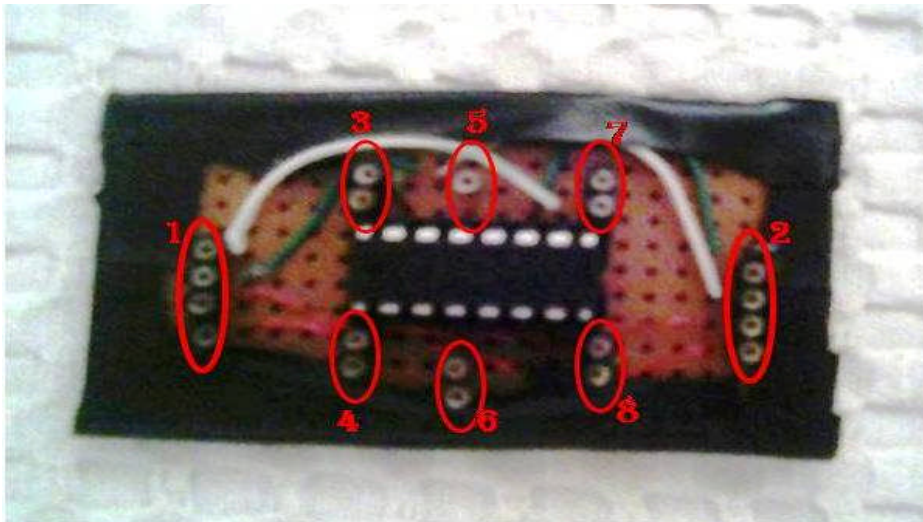


Figure 3.13: La carte de puissance.

1 : Les Entrées Des Moteurs.

2 : Les Entrées Pour L'arduino.

3 et 4 : VSS et Enable 2.

5 et 6 : la GND.

7et 8 : Enable 1 et VS

III. 2.1.4.2. Commande Des Moteurs :

Les moteurs utilisés sont des moteurs à courant continu, ces moteurs sont commandés par deux méthodes :

1- La commande de la puissance (et donc de la vitesse) par la modulation d'amplitude du signal (PWM ou Pulse Width Modulation). [34]

La commande de vitesse est réalisée en envoyant une valeur comprise entre 0 et 255 sur les pattes EN du circuit (pattes 1 pour les moteurs 1 et 9 pour le moteur 2).

2- La commande du sens de rotation par un pont en H. [34]

Pour commander les moteurs, nous allons utiliser la commande du sens par un pont H (le circuit intégré L293D) que nous avons déjà vu précédemment.

III. 2.1.4.2.A. Association moteur-réducteur :

Les moteurs à courant continu sont construits pour fonctionner en permanence dans une plage de vitesse proche de leur vitesse à vide. Cette plage de vitesse est généralement trop élevée pour la majorité des applications. Pour réduire cette vitesse, nous ajoutons au moteur un réducteur l'ensemble s'appelle motoréducteur.

III. 2.1.4.2.B. Connexion des moteurs avec la carte arduino :

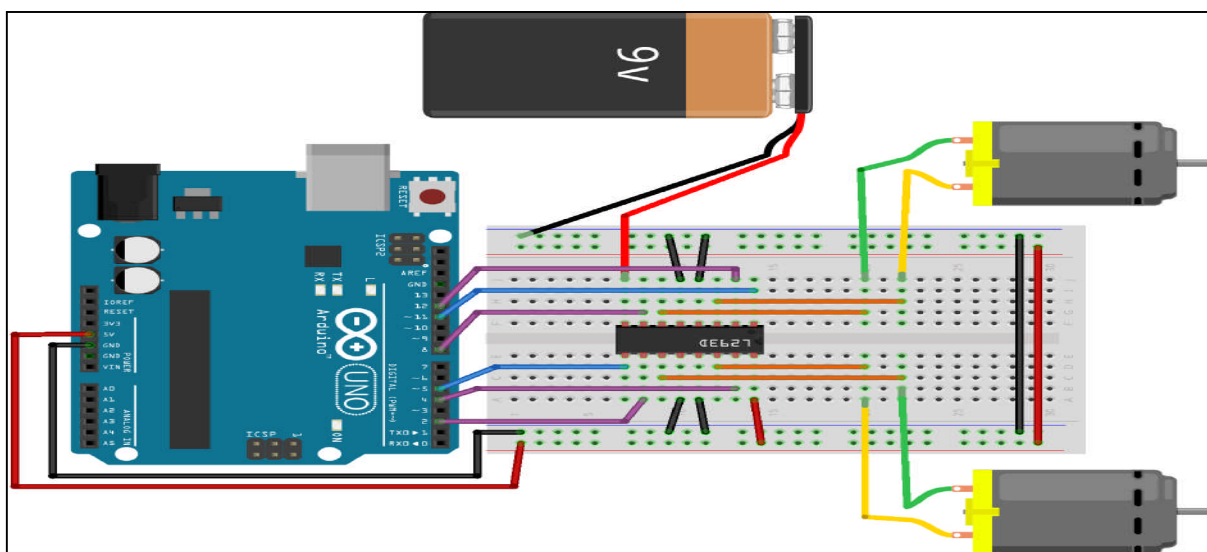


Figure 3.14: exemple de câblage des moteurs avec L293D et Arduino

Organigramme

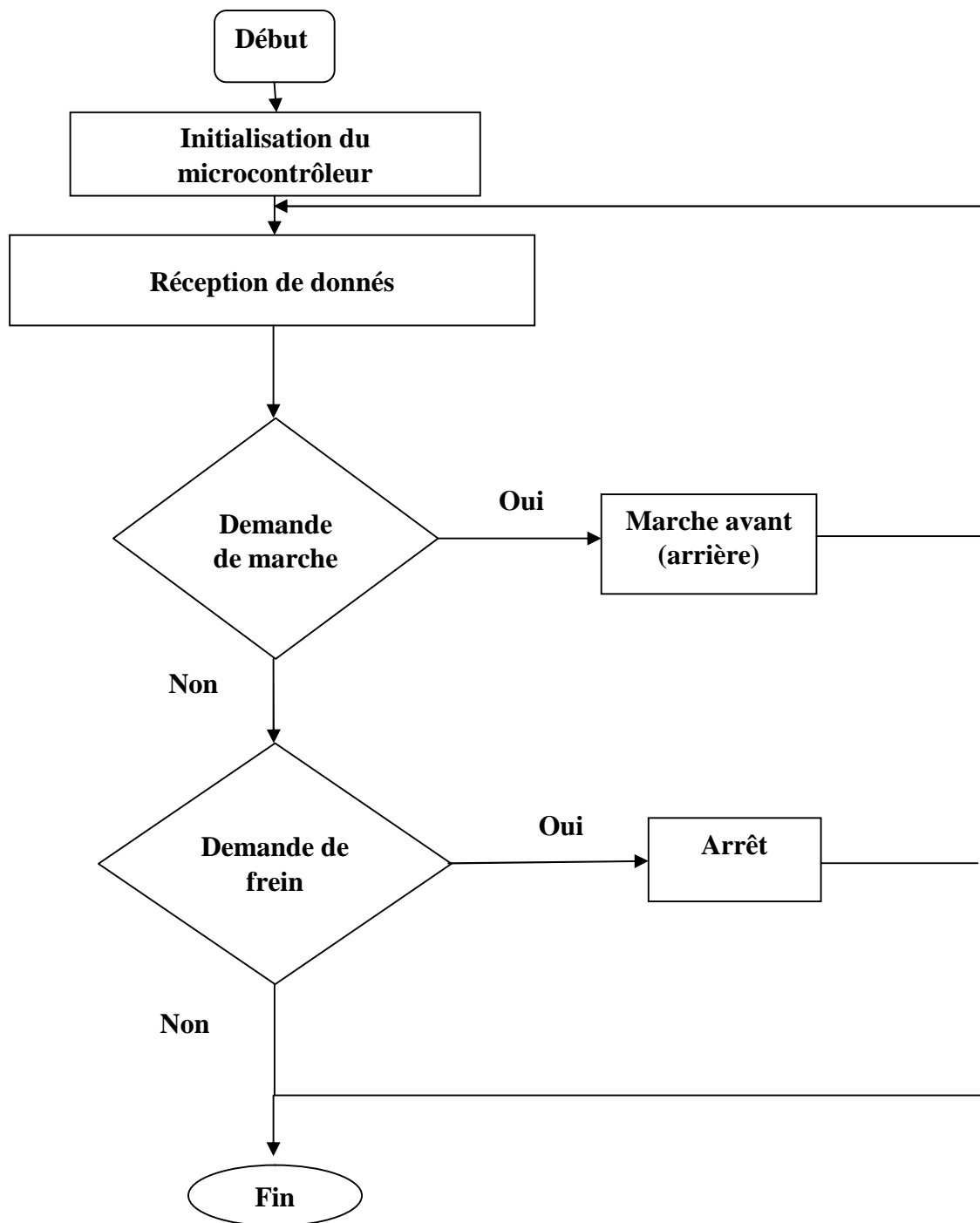


Figure 3.15: organigramme de la commande des moteurs.

III. 2.1.4.3. Le servomoteur :

Rappel : servo-moteur.

Le servo-moteur entraîne un train d'engrenage qui lui permet de réduire la puissance de rotation. L'axe qui sort du servo-moteur tourne donc bien moins vite que celui du moteur qui est à l'intérieur, en revanche son couple est plus important.

Donc ce système est puissant pour commander un déplacement angulaire dans les deux sens allant de 0 à 180°. [35]

III. 2.1.4.3.A. Connexion d'un servo-moteur à une carte Arduino

Le câblage est relativement simple comme le montre la figure ci dessous: fil rouge du servomoteur sur la broche 5V de la carte Arduino, fil noir sur la broche GND et fil blanc (ou jaune en fonction des constructeurs) sur la broche 8 de la carte Arduino.

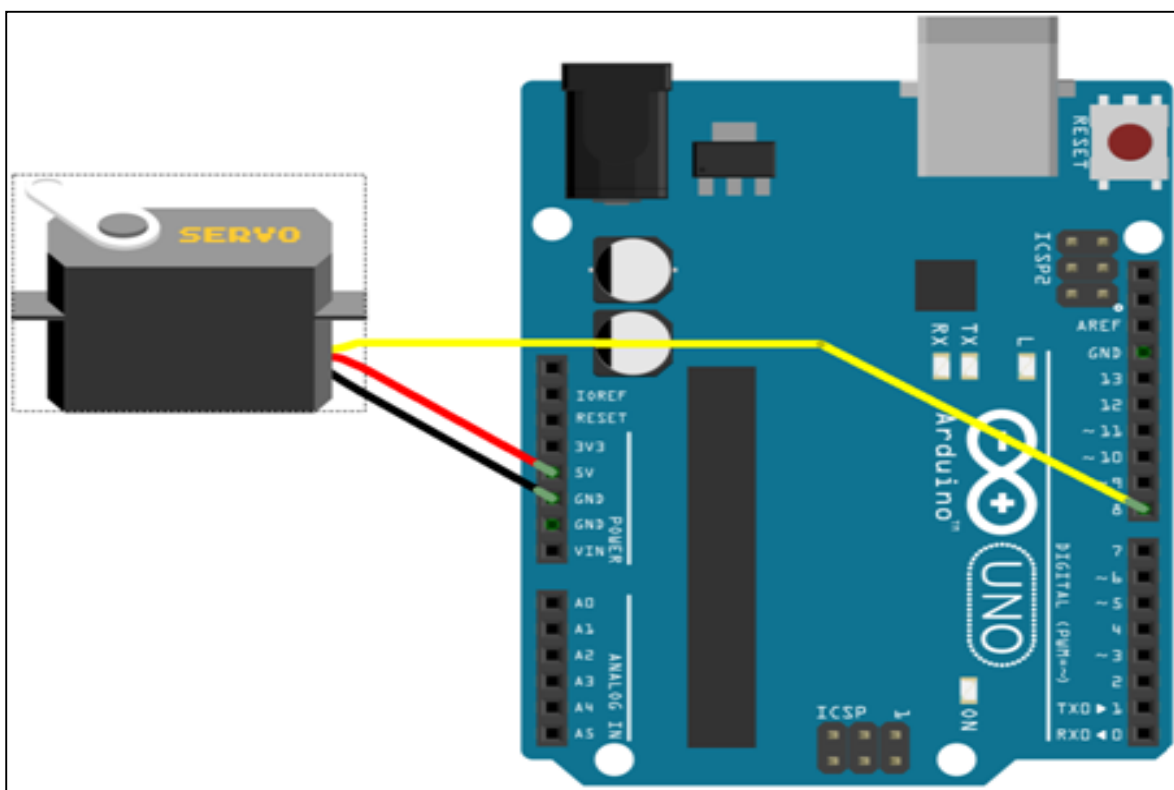


Figure 3.16: exemple de câblage de servomoteur avec l'arduino.

III. 2.1.4.3.B. Commande de servo-moteur :

Les servo-moteur sont commandé par des trains d'impulsions :

La valeur de la tension est soit +5V (HIGH de l'Arduino) soit 0V (LOW de l'Arduino),

Les impulsions sont envoyées régulièrement.

Le servo-moteur est piloté grâce à la variation de la durée de l'impulsion (x).

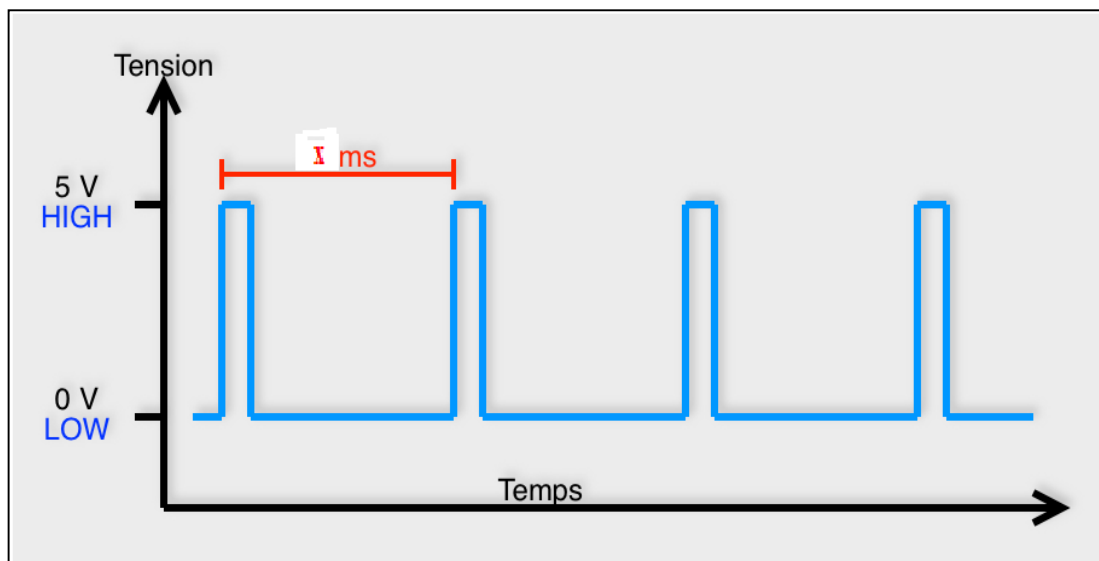


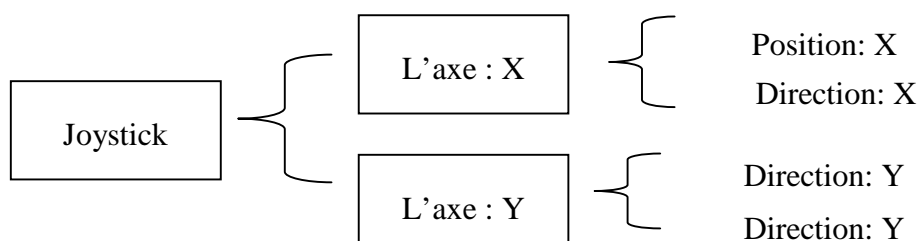
Figure 3.17: la variation de la tension en fonction de temps

III. 2.1.4.4. Joystick :

III. 2.1.4.4.A. Commander avec un joystick :

Pour diriger le robot manuel on utilise un joystick, qui possède deux potentiomètres qui nous indiquent la direction et la position sur les deux axes (X, Y) :

III. 2.1.4.4.B. Branchement de joystick :



Le cablage :

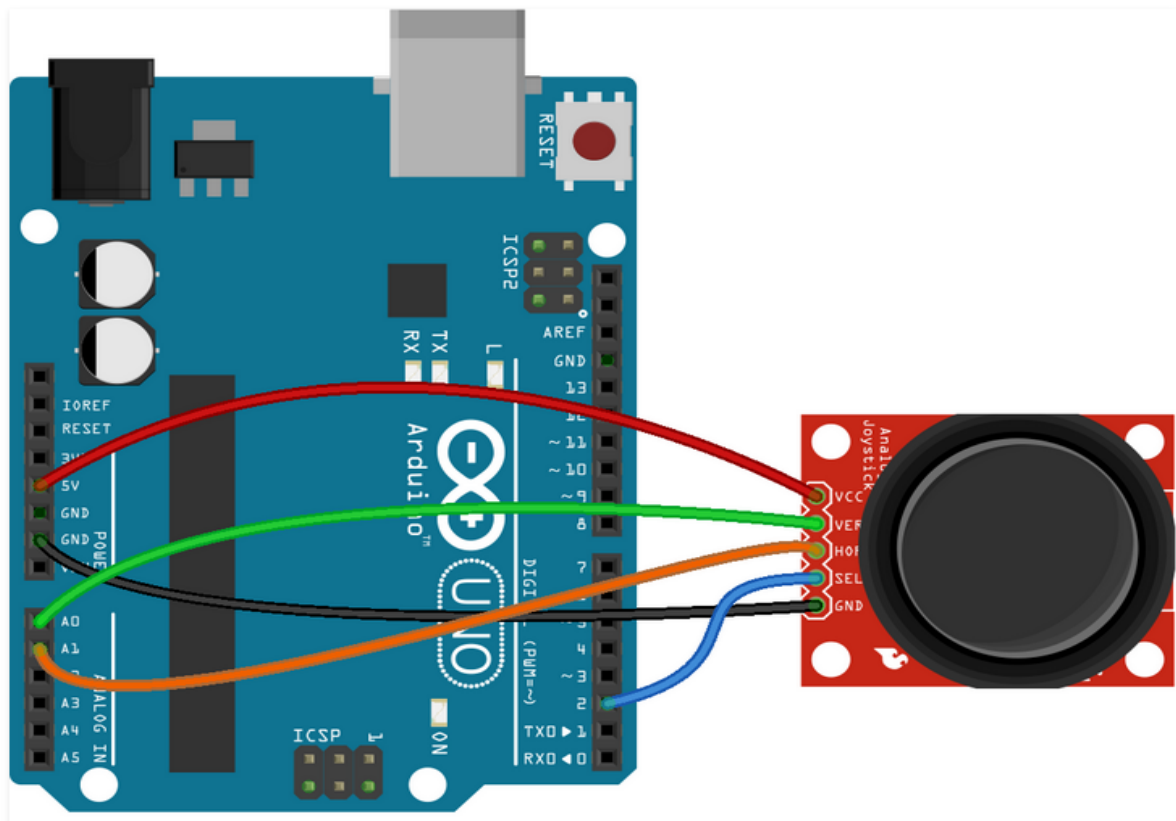


Figure 3.18 : branchement de joystick avec l'arduino

1 -VCC: à l'alimentation positive (5V ou 3.3V).

2-VERT: à un PIN d'entrée analogique de l'Arduino (A0) où il indiquera la tension sur l'axe vertical. Lorsque le joystick est centré, il indique approximativement la moitié de la tension d'entrée. L'Arduino convertit en une valeur comprise entre 0 et 1023.

3- HORIZ: à un PIN d'entrée analogique de l'Arduino (A1) où il indiquera la tension sur l'axe horizontal.

4- SEL: à un PIN d'entrée numérique de l'Arduino (2) pour le bouton poussoir, lors de l'appui, cette sortie est reliée à la masse (GND).

5- GND: la masse.

Chaque potentiomètre ayant des valeurs entre 0-1023, donc on va avoir 1024 pour chacun soit 1024x1024 pour les deux. On constate qu'un petit appui fait varier les valeurs. Dans ce cas, les moteurs se mettraient en mouvement car une valeur différente du point de repos aurait été mesurée. En prenant en compte que les directions du joystick, on peut réduire la lecture de 1024x1024 à 9 situations. [34]

III. 2.1.4.4.C. Calibrage du joystick :

Le calibrage du joystick revient à définir un intervalle sur les X et les Y qui correspond au point neutre afin d'éviter les actions intempestives.

Pour calibrer le joystick, il faut encadrer le point neutre soit 512 (0,0) par des seuils mini et max qui encadrent un intervalle au sein duquel le joystick est réputé dans sa position neutre. [34]

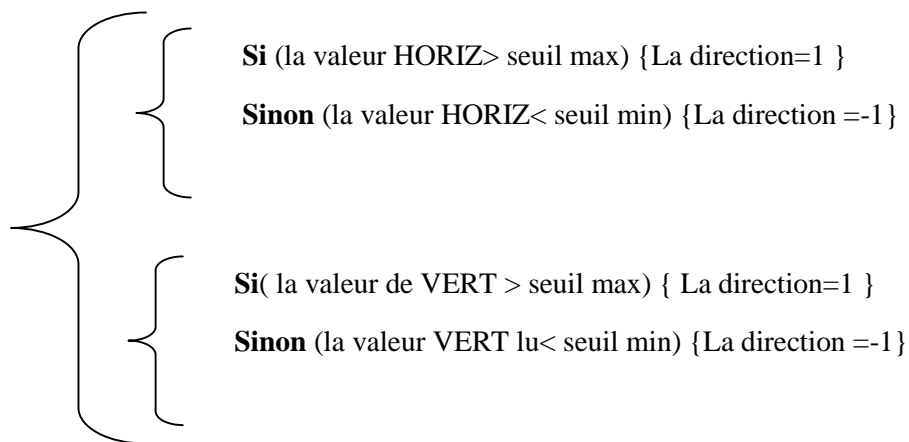
Parmi toutes ces valeurs, on peut définir 8 directions avec une position neutre (l'état repos de joystick) :

X	1	0	-1
Y			
1	(1,1)	(0,1)	(-1,1)
0	(1,0)	(0,0)	(-1,0)
-1	(1,-1)	(0,-1)	(-1,-1)

Tableau 3.3: tableau illustre les neuf positions d'un joystick

Les quatre principales directions sont teste comme se suit :

A chaque fois que les valeurs des deux potentiomètres HORIZ et VERT sont lues, on remet à 0 les variable de direction (arrêt des moteurs) et on teste :



III.2.2. Réalisation logiciel:

III.2.2.1. Application androïde :

L'application (Robot commande.APK) est une application que l'on a développée sur le site APP inventer2. Grâce à celle-ci qui est installée sur notre téléphone Android, on peut contrôler le robot en deux modes : [1]

- Manuelle.
- Automatique.



Figure 3. 20: Ecran de contrôle de la commande manuelle

III.3. Conclusion :

Dans ce chapitre nous avons décrit et détaillé le mode de fonctionnement des différents composants qui constituent la carte de commande à distance d'un robot mobile en utilisant le module Xbee pour assurer la communication à distance entre la manette de commande joystick et le robot mobile. Par la suite, nous avons configuré tous les composants du système conformément au cahier des charges en optimisant les différentes étapes de câblage de tous les composants du système de navigation et de commande pour notre robot.

IV.1. Introduction :

Ce chapitre n'est qu'une suite des chapitres précédents, son objectif est la réalisation de robot.

Tous les composants discutés aux deuxièmes et troisièmes chapitres seront utilisés dans ce qui suit.

IV .2. Réalisation de robot :

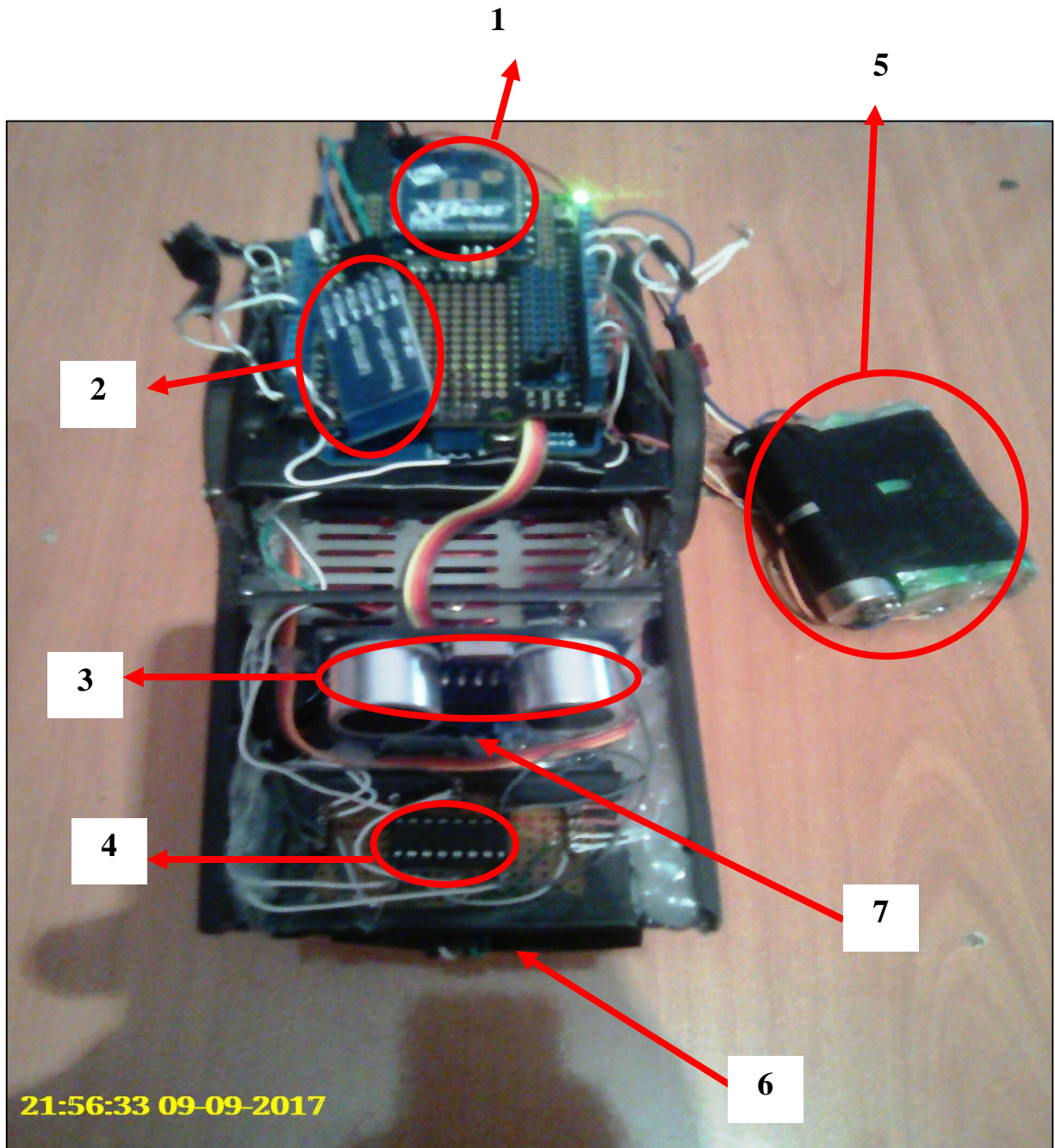


Figure4.1 : les différent partie de robot

Chapitre IV Réalisation Pratique

Le robot est constitué de :

1. module xbee pour la communication entre le robot et la manette
2. module bluetooth pour la communication entre l'application manette robot et le robot
3. Le capteur ultrasonique son rôle est détection des obstacles
4. une carte de puissance à base de circuit L293D qui est le driver des moteurs
5. une batterie de 11V pour assurer l'énergie au robot
6. Le capteur ultrasonique son rôle est de protéger le robot de tomber dans les trous

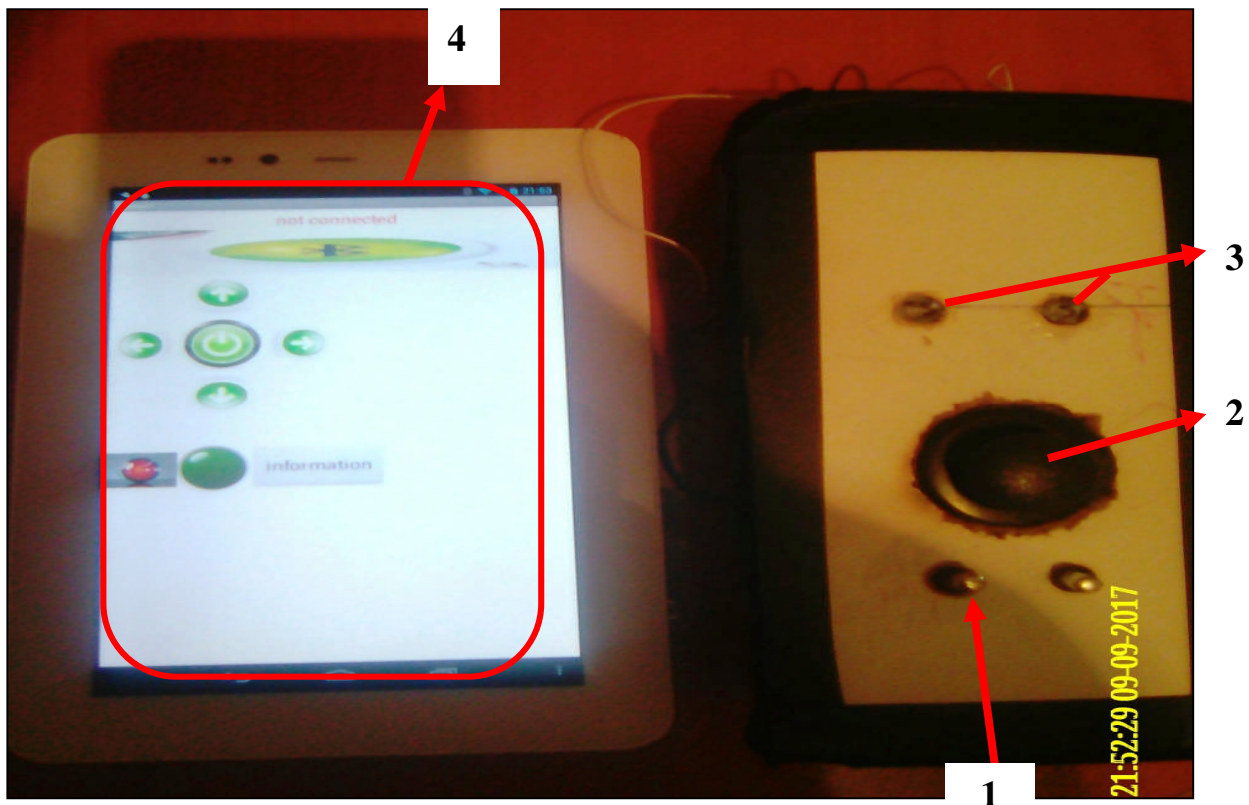


Figure 4.2 : les manettes

1. un bouton pour changer le mode (manuelle ou automatique)
2. un Joystick pour déterminer la direction de robot
3. les leds témoins verte (mode automatique) rouge (mode manuelle)
4. L'application robot manette

IV.3. Organigramme Finale du robot :

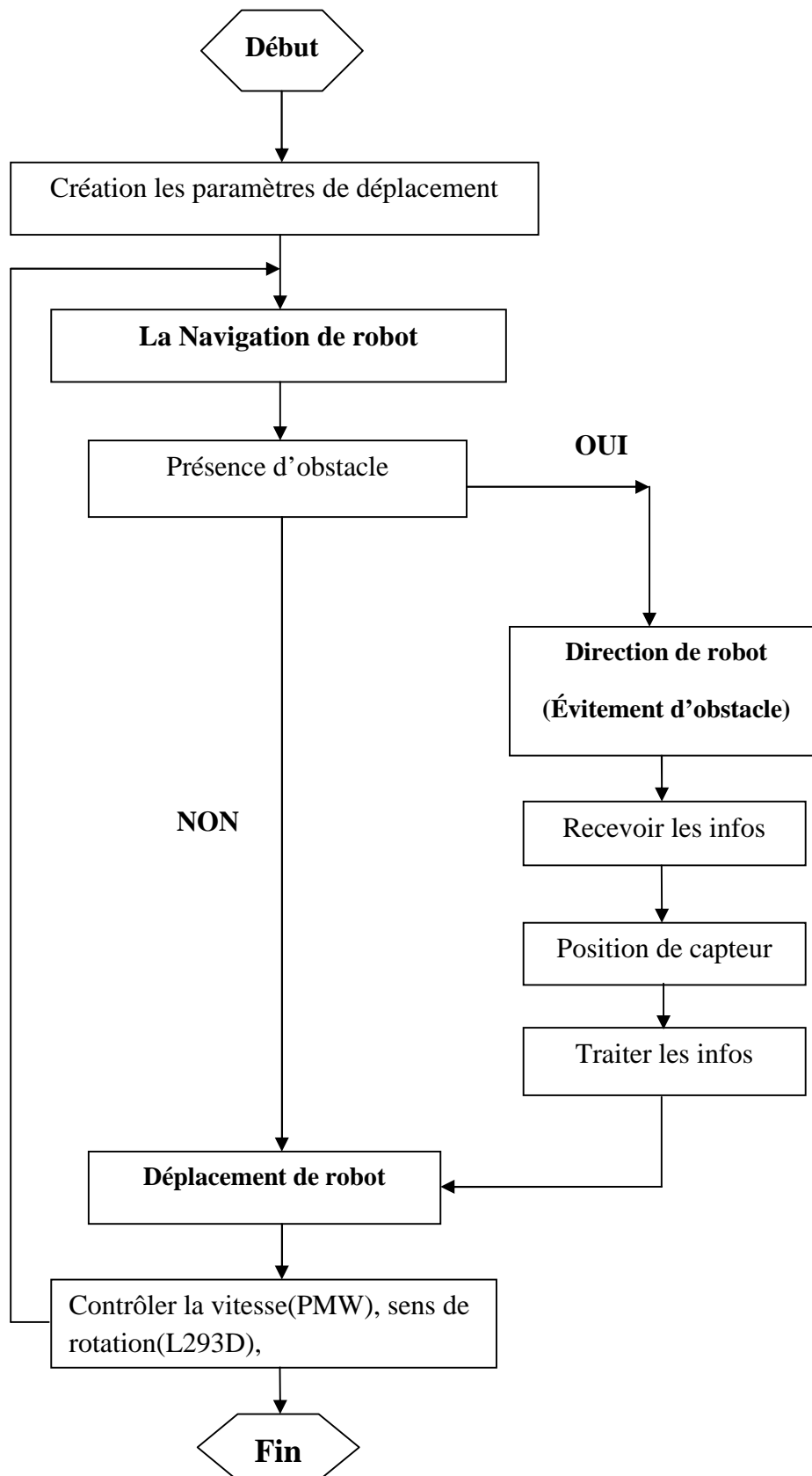


Figure 4.3 : Organigramme finale pour le robot avec évitement d'obstacle.

IV .4. Mode de navigation :

IV .4.1. Sélectionner le chemin :

Le robot se déplace aléatoirement dans l'espace depuis sa position initiale. Le but de ce comportement est la recherche d'un site d'abri tout en évitant les obstacles. L'environnement pour le robot est totalement inconnu.

IV .4.2. Evitement d'Obstacles automatiquement :

L'évitement d'obstacles est un comportement de base présent dans quasiment tous les robots mobiles. Il est indispensable pour permettre au robot de fonctionner dans son environnement. Il permet d'éviter les obstacles apparus dans le champ de vision du robot, pour cela le robot doit déterminer un autre chemin. Le comportement d'évitement d'obstacle est déterminé par un capteur US selon l'environnement dans lequel il évolue et selon sa position par rapport aux obstacles rencontrés.

Les directions autour du robot (entre l'angle 30° et 150°) sont représentées par une distance.

Ces distances sont des unités utilisées pour traiter les rotations des moteurs. (La plus grande distance + son angle par rapport au servo-moteur) nous permettront de déterminer dans quel sens les moteurs doivent se tourner.

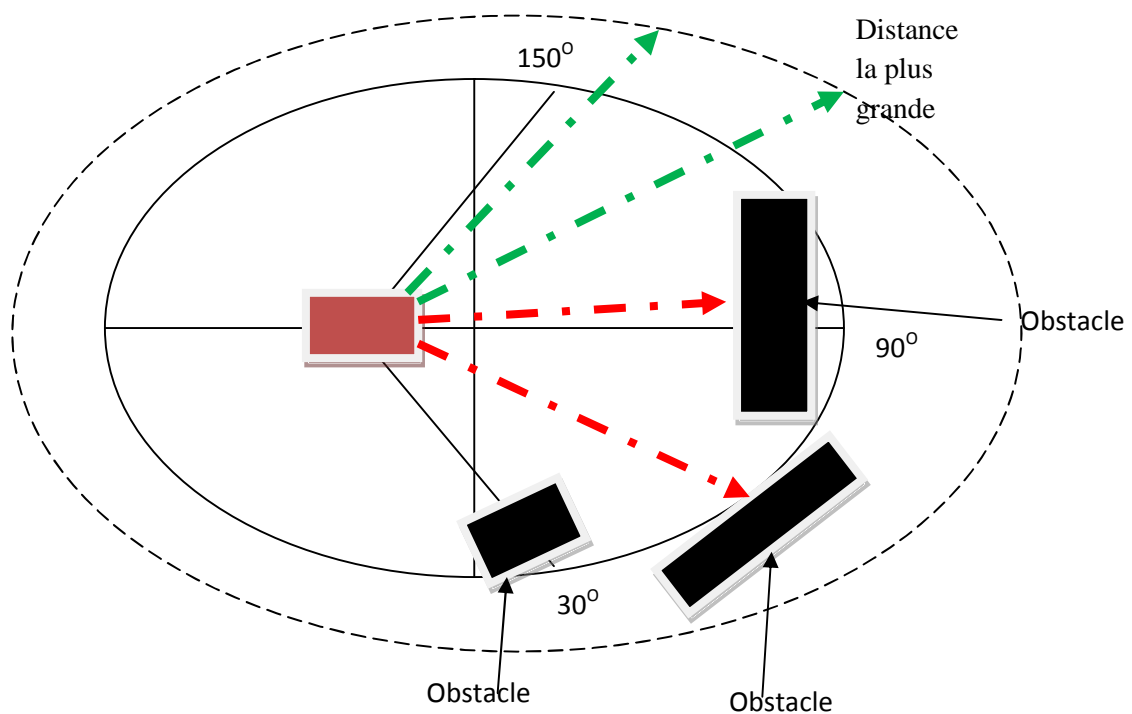


Figure 4.4 : les angles de navigation

Chapitre IV Réalisation Pratique

Si le robot ne reçoit aucune information de ses capteurs au cours de son déplacement, il considère que le chemin est libre c'est-à-dire aucune présence des obstacles, donc dans cette situation le robot se déplace vers la cible jusqu'à ce qu'un obstacle soit détecté par l'un des capteurs dans l'un des champs de perception précédemment mentionnés.

A ce point, la direction (l'angle de braquage permet la rotation en place dans les deux sens) est modifiée par le contrôleur flou d'évitement d'obstacle de telle sorte que le robot freine d'autant plus qu'il est loin de l'obstacle. Le robot doit réagir en "temps réel" c'est à dire exécuter une action réflexe qui est de s'arrêter brusquement et d'exécuter une trajectoire de contournement. Le chemin de contournement n'est pas issu d'une planification mais d'une méthode de contournement gérée par le contrôleur d'évitement d'obstacles. Une trajectoire est représentée comme un ensemble de points de passage reliant le point de départ au point d'arrivée.

IV.5. Méthode de contrôle du robot:

Le robot à l'état initial est au repos, Dès que l'opérateur donne l'ordre de démarrage par l'une des commandes (xbee, Bluetooth) soit :

- Mode automatique.
- Mode manuelle.

IV.5.1-Mode automatique :

Il suffit de choisir ce mode par une action sur le bouton de la manette ou bien la commande par Bluetooth à travers une application android que nous avons réalisée

Le fonctionnement de robots dans ce cas est simple il va marcher d'une manière autonome.

IV.5.1-Mode manuelle :

Dans le mode manuelle il suffit d'activer le Bluetooth du téléphone et de le relier au module Bluetooth du HC-06. Après que cette liaison soit effectuée, il faut activer la connexion Bluetooth de l'application de la page de commande manuelle. Il suffit de cliquer sur les boutons voulus pour contrôler le Robot. L'application va générer des codes qu'elle va envoyer à notre carte qui elle va contrôler le robot via la liaison Bluetooth.

Chapitre IV Réalisation Pratique

Ou bien avec une manette la connexion xbee elle est configuré une fois puis on peut manipuler le robot avec un joystick, la manette va générer des codes qu'elle va envoyer à notre carte qui elle va contrôler le robot via la liaison xbee.

IV.6. Les directions du robot dans l'espace :

A-Marche avant : les deux moteurs ont la même vitesse.

```
void TraitementManuel(){  
if((compar("AV")) || (BTHAV)){  
digitalWrite(MGAV,HIGH);  
digitalWrite(MGAR,LOW);  
  
digitalWrite(MDAV,HIGH);  
digitalWrite(MDAR,LOW);  
}  
void TraitementAuto (){  
if(CapteurUSRADAR(>50){  
Serial.print("Distance =  
");Serial.println(CapteurUSRADAR());  
digitalWrite(MGAV,HIGH);  
digitalWrite(MGAR,LOW);  
digitalWrite(MDAV,HIGH);  
digitalWrite(MDAR,LOW);}
```

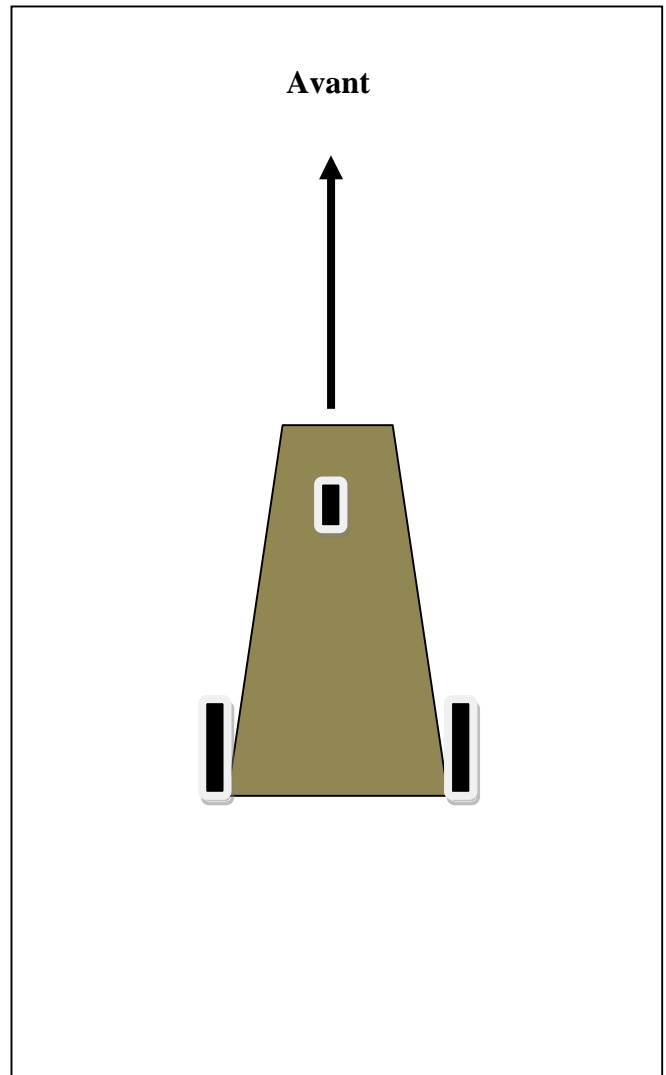


Figure 4 .5 : marche avant de robot

B-Marche à droite : le moteur gauche en marche et le moteur droite en arrêt.

```
void TraitementManuel(){  
  if((compar("AV")) || (BTHAV)){  
    digitalWrite(MGAV,HIGH);  
    digitalWrite(MGAR,LOW);  
  
    digitalWrite(MDAV,LOW);  
    digitalWrite(MDAR,LOW);  
  }  
  void TraitementAuto () {  
    if(CapteurUSRADAR(>50){  
      Serial.print("Distance =");  
      Serial.println(CapteurUSRADAR());  
      digitalWrite(MGAV,HIGH);  
      digitalWrite(MGAR,LOW);  
      digitalWrite(MDAV,LOW);  
      digitalWrite(MDAR,LOW);}  
}
```

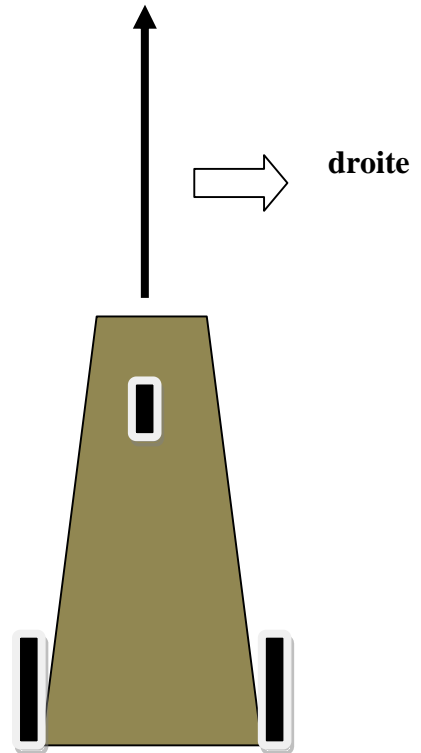


Figure 4 .6 : marche a droite de robot

C-Marche à gauche : le moteur gauche en arrêt et le moteur droite en marche.

```
void TraitementManuel(){  
  if((compar("AV")) || (BTHAV)){  
    digitalWrite(MGAV,LOW);  
    digitalWrite(MGAR,LOW);  
  
    digitalWrite(MDAV,HIGH);  
    digitalWrite(MDAR,LOW);  
  }  
  void TraitementAuto (){  
    if(CapteurUSRADAR(>50){  
      Serial.print("Distance =");  
      Serial.println(CapteurUSRADAR());  
  
      digitalWrite(MGAV, LOW);  
      digitalWrite(MGAR,LOW);  
  
      digitalWrite(MDAV,HIGH);  
      digitalWrite(MDAR,LOW);}  
}
```

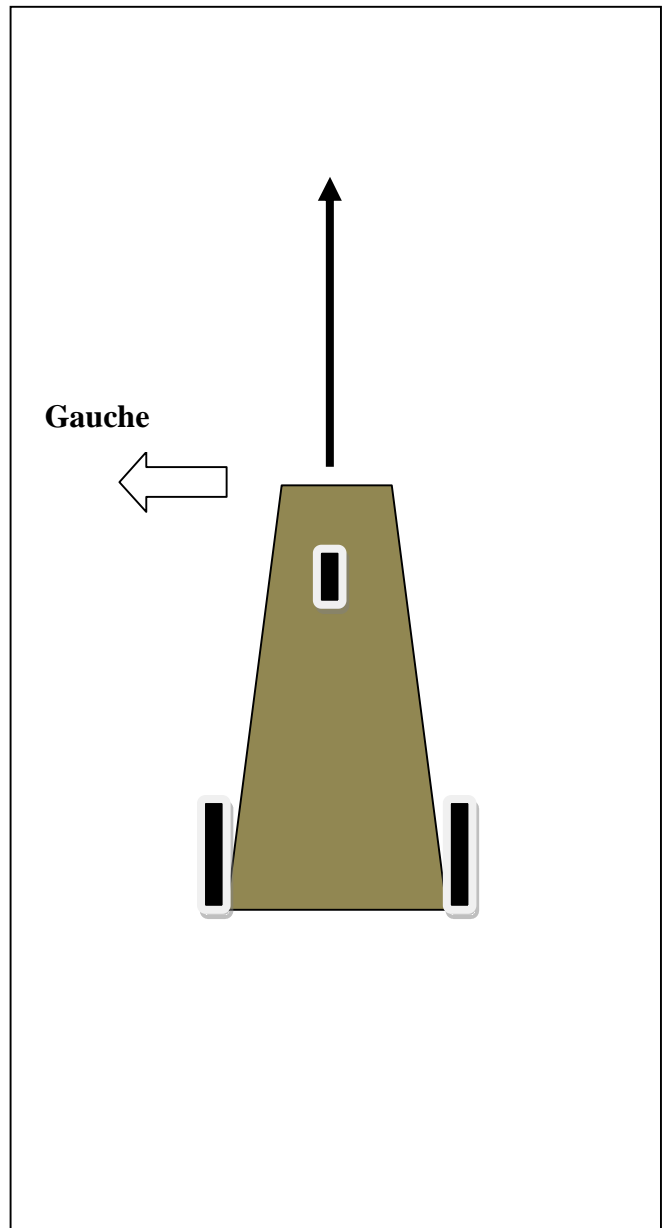


Figure 4.7 : marche a droite de robot

IV.7.Conclusion :

Chapitre IV Réalisation Pratique

A la fin de ce travail qui consiste à la mise en œuvre d'un robot mobile, le travail exige la maîtrise de plusieurs notions techniques aussi bien le domaine micro électronique et celui de l'informatique.

Ce projet couvre un grand nombre de domaines de compétences :

- ✓ étude de la commande de moteur en L293D.
- ✓ programmation des cartes arduino.
- ✓ Connaître le fonctionnement des capteurs.

Conclusion :

A la fin, on peut dire que ce projet nous a été une expérience très fructueuse, bénéfique et enrichissante. Il nous a permis de mettre le premier pas dans le monde pratique de la robotique. Il nous a permis également d'exploiter les différents éléments pertinents de la conception et la commande de robot. Sur le plan software, ce projet nous a mis directement en contact avec des langages de programmation tel que : ID arduino, processing, APP inventer2 .

Pour commander les moteurs, le servomoteur ou piloter notre robot en général, nous nous sommes intéressés à comprendre les caractéristiques, le mode de fonctionnement et la limite d'utilisation de chaque composant et comment associer tous ces éléments (carte arduino, capteur ultrasonique, Bluetooth, module Xbee, carte de puissance) et leur programme afin d'avoir une meilleure navigation pour notre robot dans son environnement.

L'objectif principal de notre projet, ce robot est appelé à devenir une plateforme d'essais et de développement très utile. Il constitue une plateforme intéressante pour faire des tests et implanter des algorithmes de haut niveau.

Ce mémoire est donc le résumé le plus possible du projet. Toutefois, En effet, ce projet est vaste et complexe et il a été impossible de documenter dans le détail de tous les composants du projet. Ainsi, certaines parties ont été résumées brièvement et il serait alors nécessaire de les documenter plus en détails pour permettre une meilleure navigation pour le robot. [40]

La réalisation matérielle et logicielle de ce projet a donné des résultats satisfaisants. Malgré les erreurs liées aux capteurs ultrasoniques. En perspective notre robot peut être amélioré:

Par la qualité du programme dont l'objectif est d'élever la qualité du programme en ajoutant les interruptions et en diminuant la consommation d'énergie.

- commander le robot avec une caméra.
- La commande à distance via Internet.
- La commande par GPS.

Les gammes de cartes Arduino :

il existe plusieurs versions de module Arduino, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d' USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.

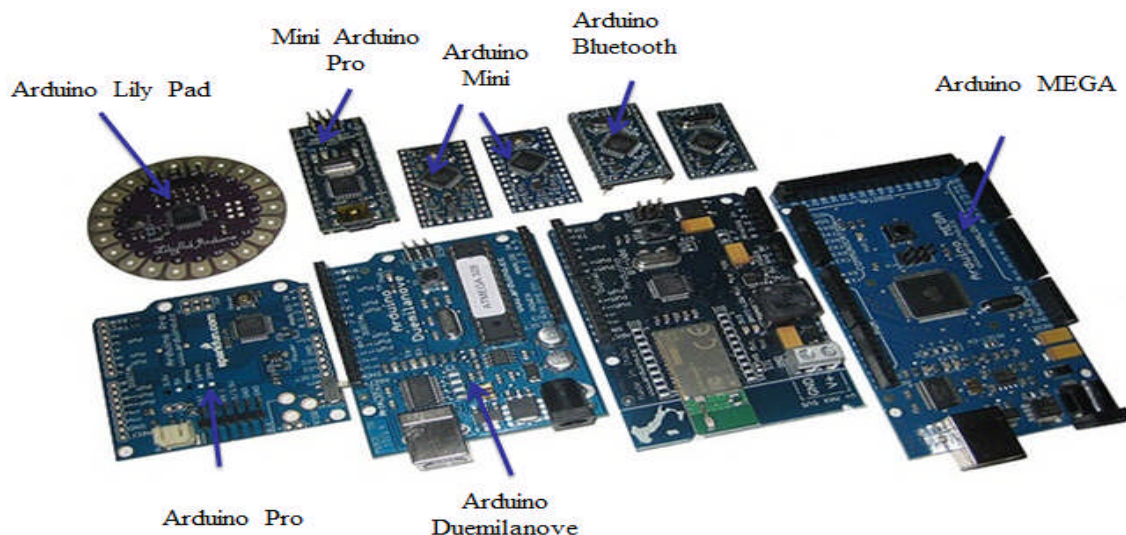


Figure 5.1: Types Des Cartes

Arduino uno :

Entrées/Sorties Numériques :

Elles sont numérotées de 0 à 13. Chacune de ces broches peuvent être utilisées comme entrées ou sorties digitales. Les signaux véhiculés par ces connecteurs sont des signaux logiques, c'est-à-dire qu'ils ne peuvent prendre que deux états : 5 Volts (HAUT) ou 0 Volt (BAS).

Certaines de ces broches ont des fonctions spécialisées :

Les broches (3, 5, 6, 9, 10, 11) peuvent produire un signal PWM .

Les broches 0 (Rx) et 1 (Tx) : R pour Recevoir et T pour Transmettre

Les broches (10, 11, 12 et 13) sont utilisées pour communiquer avec le protocole SPI (Serial Peripheral Interface).



Figure 5.2 : E/S Numériques Broches Analogiques :

Ils ne peuvent prendre que deux états HAUT et BAS, ces six entrées peuvent admettre un millier de valeurs (1024 exactement) analogiques comprises entre 0 et 5 Volts. Ces broches sont associées à des convertisseurs analogiques-numériques intégrés de 10 bits qui donnent une valeur comprise entre 0 et 1023.



Figure 5.3 : E/S Analogiques

Autres Broches :

GND

Ce sont les masses. Elles servent à fermer le circuit.

3V3

Cette broche délivre du 3V3 pour un ampérage maximum de 50 mA. Elle peut être utilisée pour alimenter un composant à ce voltage.

AREF

Voltage de référence pour les broches analogiques, à utiliser avec la fonction

Reset

Mettre cette broche à LOW permet de redémarrer la carte. Elle est beaucoup utilisée avec les shields pour permettre un redémarrage lorsque le bouton physique de la carte est inaccessible.

IOREF

Cette broche donne le voltage de référence à lequel la carte fonctionne. Dans le cas du Uno, elle sera à 5V. Sert surtout pour les shields.

Communication :

Pour communiquer, l'Arduino Uno peut employer différents protocoles comme nous l'avons vu précédemment :

- la voie série avec le port USB ainsi que les broches numériques 0 et 1.
- SPI avec les broches numériques 10, 11 12 et 13.
- I2C / Wire avec les broches analogiques A4 et A5
- de plus, la librairie Software Serial permet d'utiliser n'importe quelle broche numérique pour communiquer en voie série.

Gestion des mémoires dans la carte Arduino:

L'ATMega 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 0.5Ko également utilisé par le bootloader), l'ATMega328 a également 2Ko de mémoire RAM (volatile) et 1Ko d'EEPROM (non volatile: mémoire qui peut être lue l'aide de la librairie EEPROM).

Microcontrôleur ATMega328:

Le microcontrôleur utilisé sur la carte Arduino uno ATmega328 est un microcontrôleur ATMEL de la famille AVR 8 bits. Ces Principales caractéristiques sont:

- **FLASH**= mémoire programme de 32KO,
- **SRAM**= données (volatiles) 2KO,
- **EEPROM**= données (non volatiles) 1KO,
- **Digital I/O 'entrées-sorties Tout ou Rien**=3 ports PortB, PortC, PortD (soit 23 broches en tout I/O),
- **Tension d'alimentation interne**=5V.
- **Courant max broches E/S**=40mA.
- **Courant max sur sortie 3.3V**=50mA
- **Fréquence horloge** = 16MHz.
- **Dimensions**= 68,6mm*53.3mm

Descriptions des broches

- **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.. PD0)** 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 analogique 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.

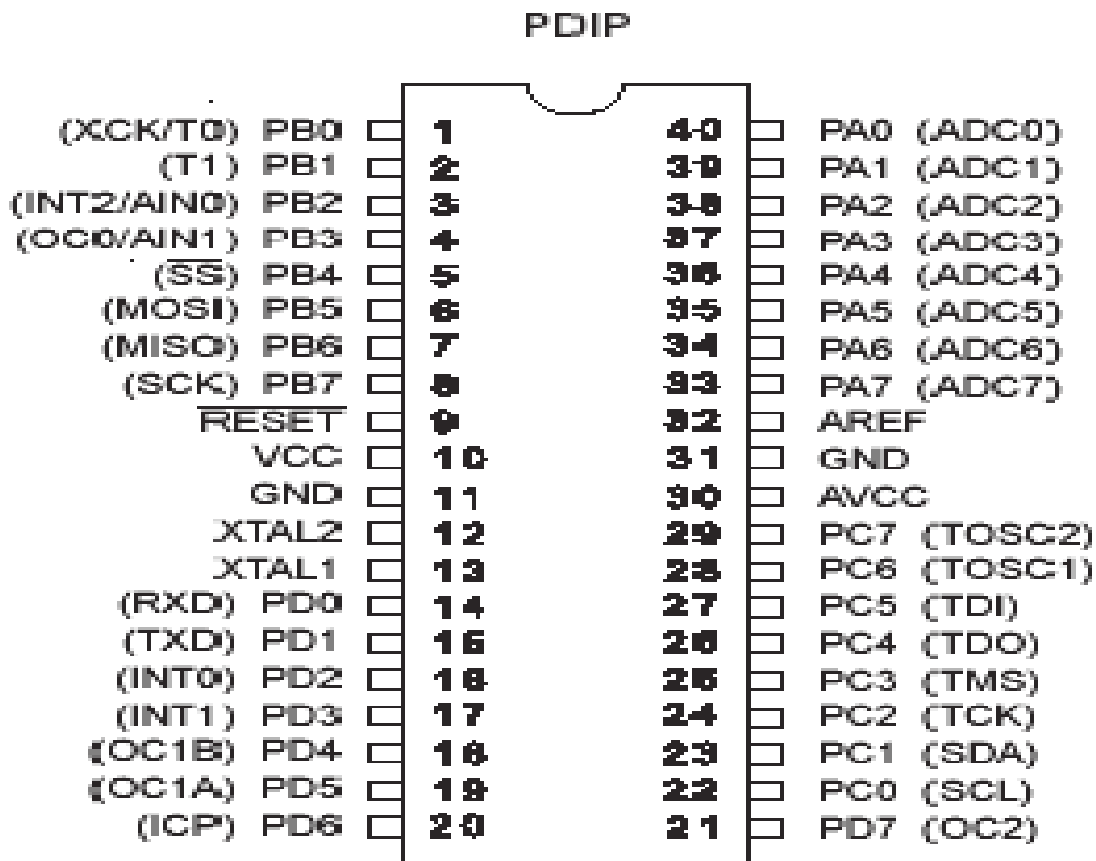


Figure 5.4: brochage typique d'un ATMEGA32

L'Architecture de l'atmega328 :

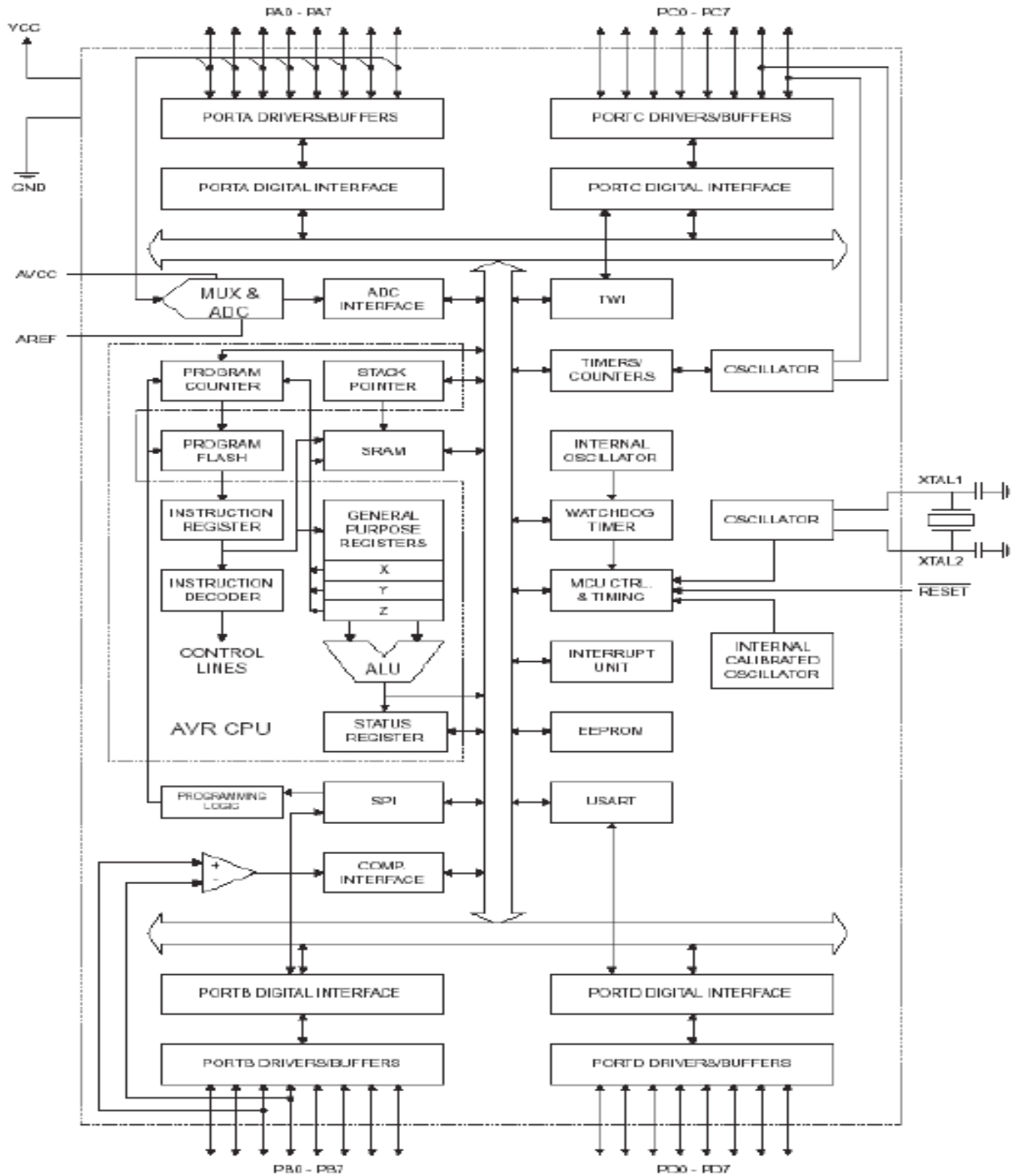


Figure 5.5: Architecture interne ATmega328

Module Xbee :

Brochage:

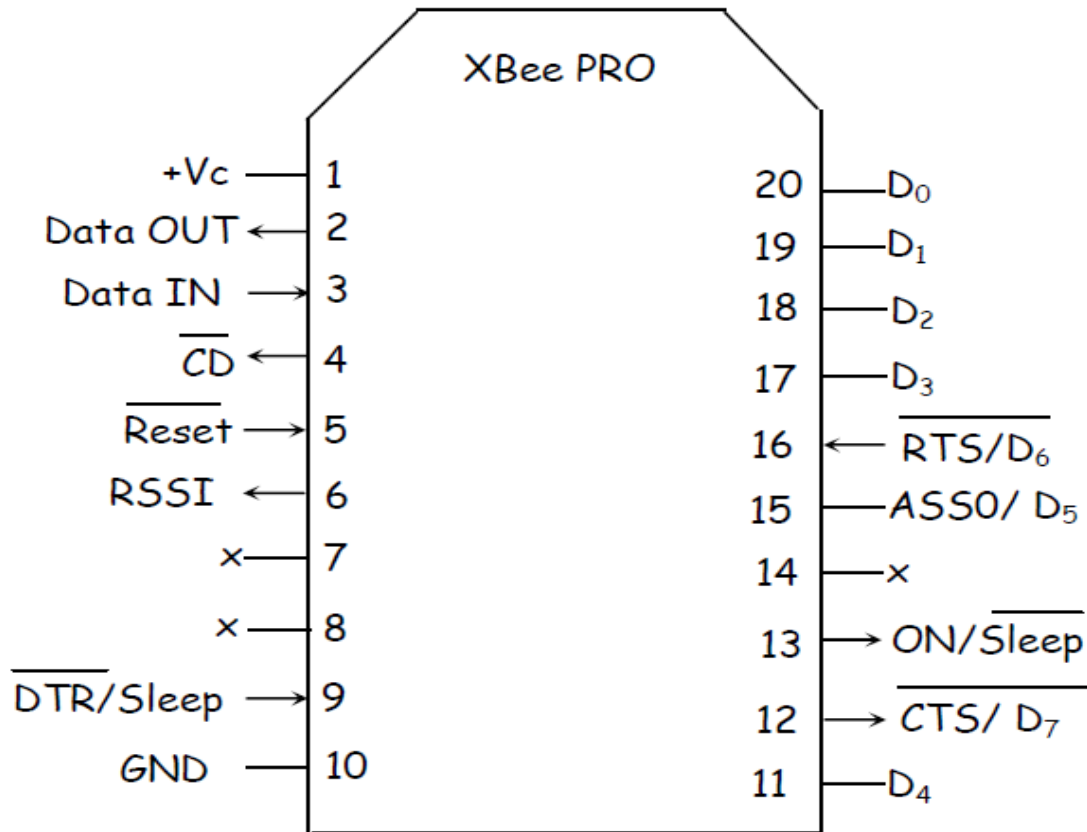


Figure 5.6 : les broches d'un Xbee

Configuration

La configuration des paramètres de la liaison RS232 (Vitesse, parité, Start et stop) se fait avec une commande AT.

Pour modifier ou lire les paramètres du module, on va dialoguer par des commandes "AT" à 9600 Baud ;

Ces modules peuvent être configurés avec plusieurs méthode on cite :

- Avec le moniteur série qui existe sur IDE arduino comme nous avons expliqué dans le chapitre 3.
- Avec le logiciel **X-CTU**.

Principales Commande AT:

ATCN : Pour quitter le mode commande.

ATCT + paramètre (0xFFFF): Modifie ou lit le Time Out qui fait repasser le module en mode IDLE si aucune commande AT ne parvient.

ATGT + paramètre (0xFFFF): Modifie ou lit le temps de garde.

ATCC + paramètre (0xFF): Modifie ou lit le caractère ASCII utilisé pour passer en mode commande.

ATWR : Sauve les paramètres dans la mémoire non volatile. Il faut impérativement attendre la réponse "OK" du module avant de lui envoyer une nouvelle commande.

ATCH + paramètre (0x0C à 0x17) : Modifie ou lit le canal utilisé dans la bande 2,4 GHz. Par défaut il y a 0x0C.

ATDH + paramètre (0xFFFFFFFF) : Modifie ou lit les 32 bits de l'adressage destinataire. Par défaut il y a 0x00000000

ATDL + paramètre (0xFFFFFFFF) : Modifie ou lit les 32 bits de l'adressage destinataire. Par défaut il y a 0x00000000

ATMY + paramètre (0xFFFF): Modifie ou lit les 16 bits de l'adressage source. Par défaut il y a 0x0000

ATSH : Lit les 32 bits MSB du numéro de série du module.

ATSL : Lit les 32 bits LSB du numéro de série du module.

ATNI + paramètre (20 octets ASCII): Sauve une chaîne de 20 caractères max pour l'identification du réseau : NI. Le caractère "espace" met fin à la commande

ATND : Cherche et donne les modules trouvés. Pour chacun on obtient: MY + SH + SL +DB + NI. La commande se termine au bout de 2,5 secondes et le module renvoie un "CR".

ATPL + paramètre (0 à 4) : Modifie ou lit la puissance de sortie du module. Par défaut il y a 4 soit la puissance max de 60 mW.

ATBD + paramètre (0 à 7) : Modifie ou lit la vitesse en Baud de la liaison RS232. Par défaut on a 3 soit 9600 bauds.

ATID + paramètre (0xFFFF) : Modifie ou lit l'adresse du Pan ID. Il faut que cette valeur soit la même pour que les modules puissent communiquer entre eux

Mise en sommeil (mode SLEEP) :

Elle se fait par la pin 9 (DTR). Si on met un niveau "1" sur la pin 9, le module passe en mode "Sleep" et ne consomme plus que 10 mA sous 3 V. Pour le repasser en mode normal il faut mettre un "0" sur la pin 9. Le temps de réveil est d'environ 13 ms.

Il faudra auparavant paramétrer le module par la commande AT suivante: **SM=1**.

3.3. Exemple de configuration avec le moniteur série sur L'IDE arduino :

+++ Pour entrer dans le mode commande.

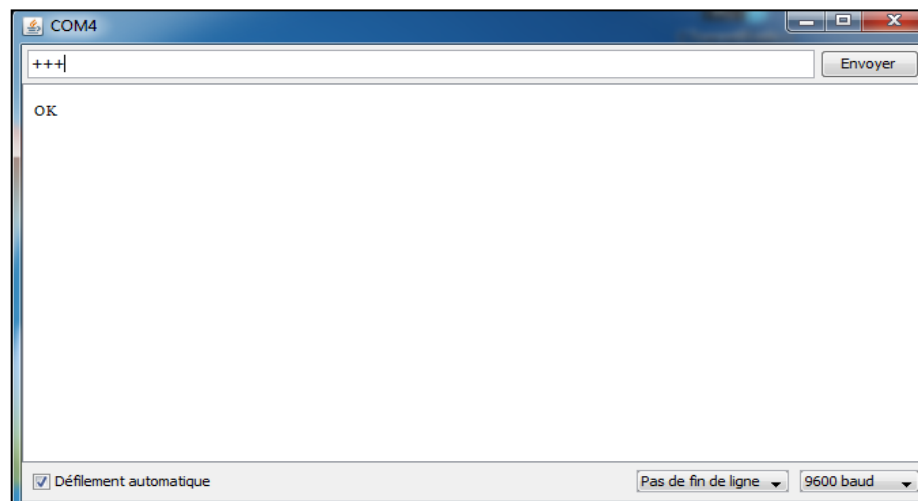


Figure 5.7: la configuration Xbee

ATID 1111 Adresse du réseau.

ATMY 1 Adresse du module dans le réseau.

ATDL 0 Adresse du destinataire dans le réseau.

ATWR Écrit la nouvelle configuration dans la mémoire flash du module.

ATCN Sort du mode configuration.

Note : Dans ce simple exemple, ses commande ce sont juste pour configurer l'adressage de module Xbee et comment les enregistrer dans la mémoire non volatile.

Capteur ultrasonique :

Mise en œuvre :

Il faut envoyer une impulsion niveau haut (+ 5v) d'au moins 10 μ s sur la broche Trig Input pour déclencher une mesure. (Voir Fig.1) En retour la sortie Output ou (Echo), va restituer une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet.

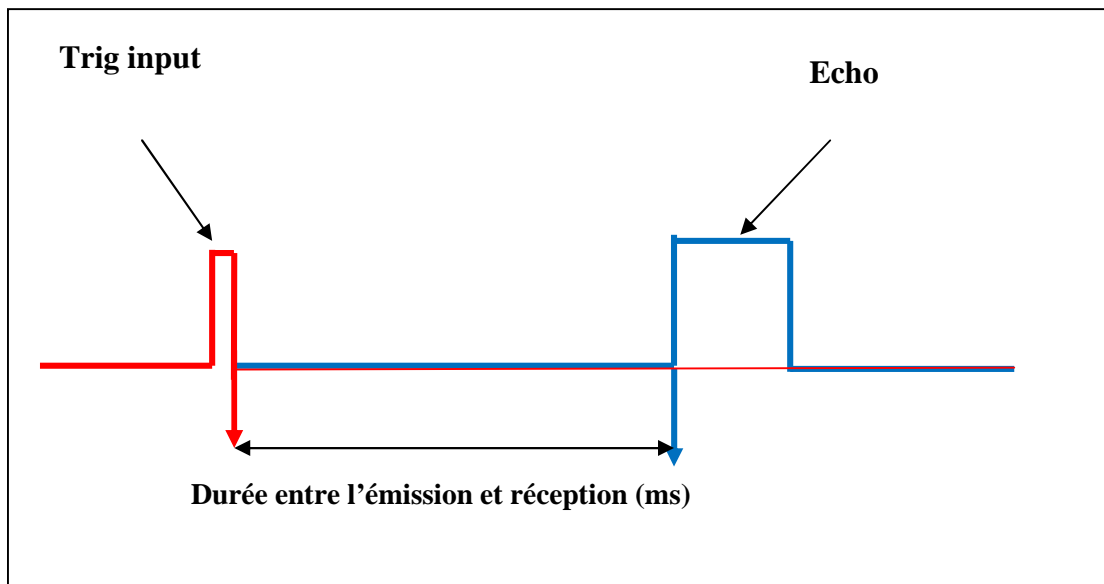


Figure 5.8 : émission et réception des ondes sonores par un HC-SR04

Afin de pouvoir calculer la distance en cm, on utilisera la formule suivante : Distance = (durée de l'impulsion en μ s) / 58.

L'étalonnage

Au démarrage de l'Arduino, le circuit règle sa mesure de distance en évaluant ce qui se trouve droit devant lui. Si un obstacle est présent à cet instant, la mesure peut être faussée. Il faut veiller à ce que le champ soit dégagé.

Exemple de câblage du module à l'Arduino:

- Broche 12 de l'Arduino vers Trig Input. (Trig)
- Broche 11 de l'Arduino vers Output. (Echo)

Exemple de programme :

Dans l'exemple a chaque fois que le capteur détecte une distance inférieur à 50cm led va s'allumer sinon elle va s'éteindre.

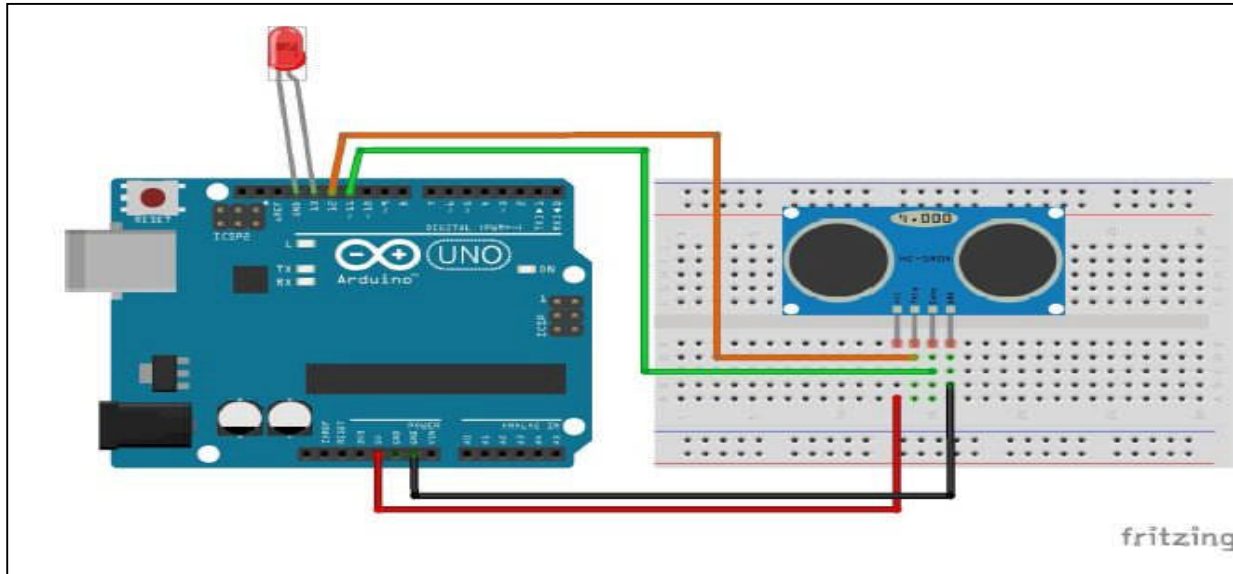


Figure 5.9 : câblage de l'ultrasonique

Code source

```
sketch_sep10a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide

sketch_sep10a $
int Trig = 12;
int Echo = 11;
long Lecture_Echo;
long Distance;
int led=13.
void setup()
{
  pinMode(led, OUTPUT);
  pinMode(Trig, OUTPUT);
  digitalWrite(Trig, LOW);
  pinMode(Echo, INPUT);
  Serial.begin(9600);
}
void loop()
{
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  Lecture_Echo = pulseIn(Echo, HIGH);
  Distance = Lecture_Echo / 58;
  Serial.print("Distance en cm : ");
  Serial.println(Distance);
  delay(500);
  if(Distance<50){
    digitalWrite(led,HIGH);
  }else{ digitalWrite(led,LOW); }
}
```

Compilation terminée.
C:\Users\M2DSF-1\AppData\Local\Temp\build4178272834349995818.tmp\sketch_sep10a.cpp.elf
C:\Users\M2DSF-1\AppData\Local\Temp\build4178272834349995818.tmp\sketch_sep10a.cpp.hex
Taille binaire du croquis : 3 450 octets (d'un max de 32 256 octets)

27 Arduino Uno on COM5

Module Bluetooth :

Le module Bluetooth HC-06 permet d'établir une liaison Bluetooth (liaison série) entre une carte Arduino et un autre équipement possédant une connexion Bluetooth (Smartphone, tablette, seconde carte Arduino, etc...). Ces deux modules peuvent être configurés grâce à des commandes AT (ou commandes Hayes).

Configurer le module bluetooth HC-06

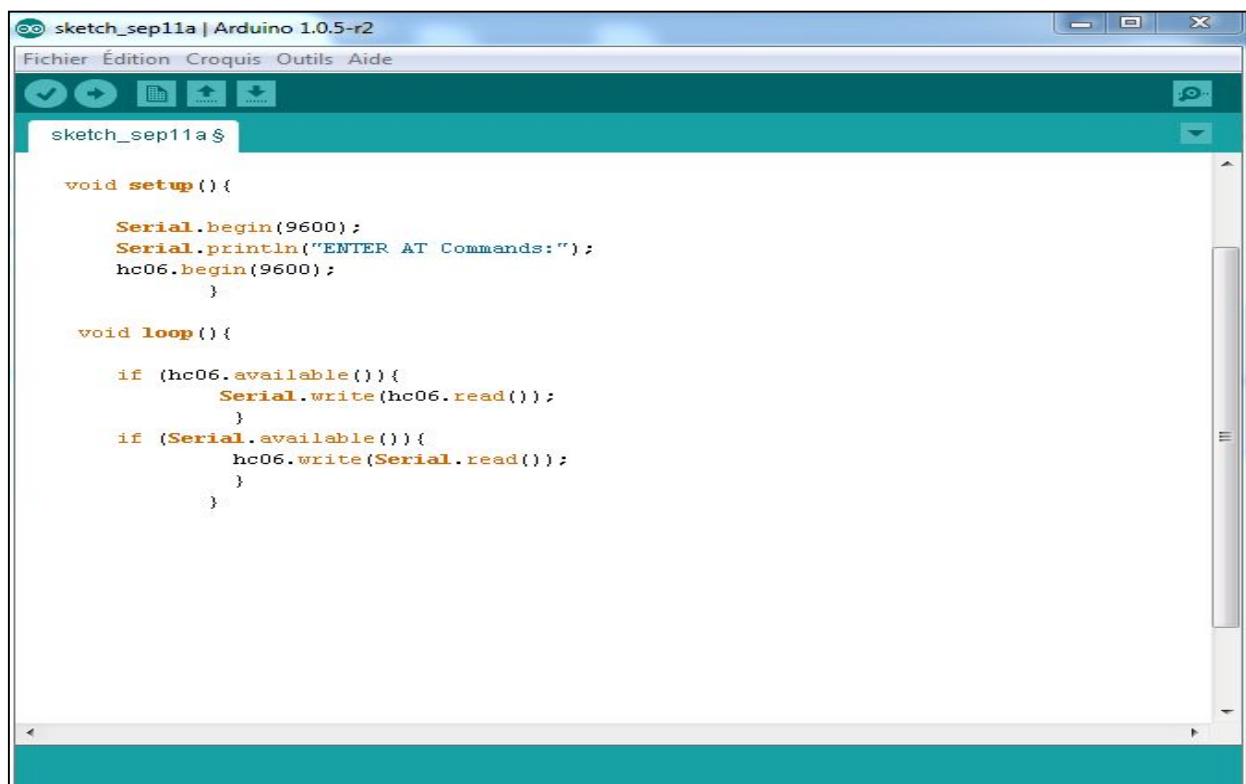
le module Bluetooth HC-06, il est paramétré par défauts encodés à l'intérieur du module.

(comme l'ID et le mot de passe) pour des raisons de sécurité il est préférable de changer ces paramètres.

Pour les configurer nous allons procéder comme suite :

On branche Rx de module à la pince 11 et Tx à la pince 10 et bien sûr sans oublier de brancher Vcc (3.3V-5V) et GND.

Téléverser le programme suivant dans la carte arduino :

The image shows a screenshot of the Arduino IDE interface. The window title is "sketch_sep11a | Arduino 1.0.5-r2". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, undo, redo, and uploading. The main text area shows the following code:

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("ENTER AT Commands:");  
  hc06.begin(9600);  
}  
  
void loop() {  
  if (hc06.available()) {  
    Serial.write(hc06.read());  
  }  
  if (Serial.available()) {  
    hc06.write(Serial.read());  
  }  
}
```

Réglez votre moniteur série sur "Pas de fin de ligne" & "9600 Bauds" et taper AT pour commencer la configuration.

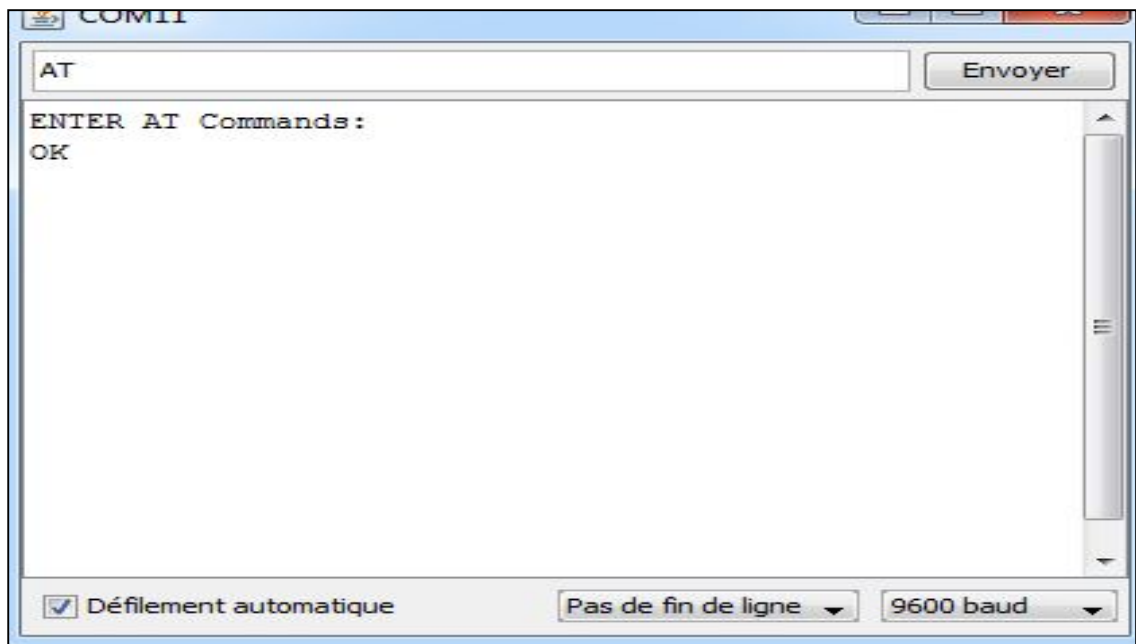


Figure 5.10: début de configuration

Pour changer le nom de votre module tapez: **AT+Nom** (par exemple : AT+teste) si vous recevez OK signifie que le nom est configuré.



Figure 5.11 : modification de nom

Pour changer le mot de passe tapez :**AT+Nom**(par exemple : AT+2017) le mot de passe qui est par défaut est <1234>.

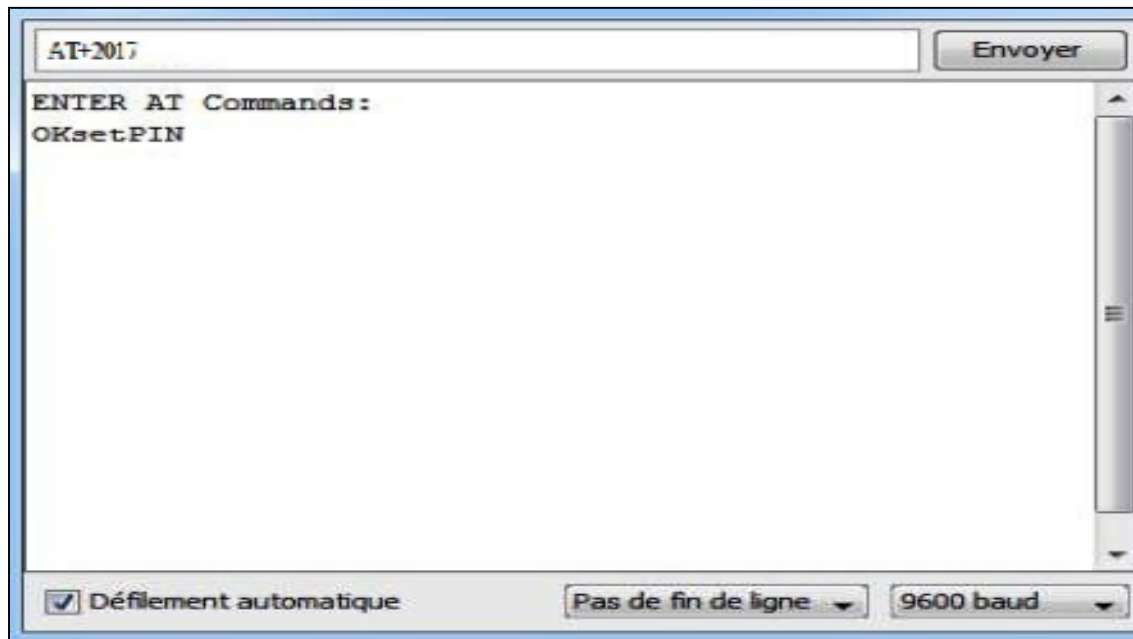


Figure 5.12 : modification de l’ID ou mot de passe

Le servomoteur:

Exemple pour piloter un servomoteur

Branchement :

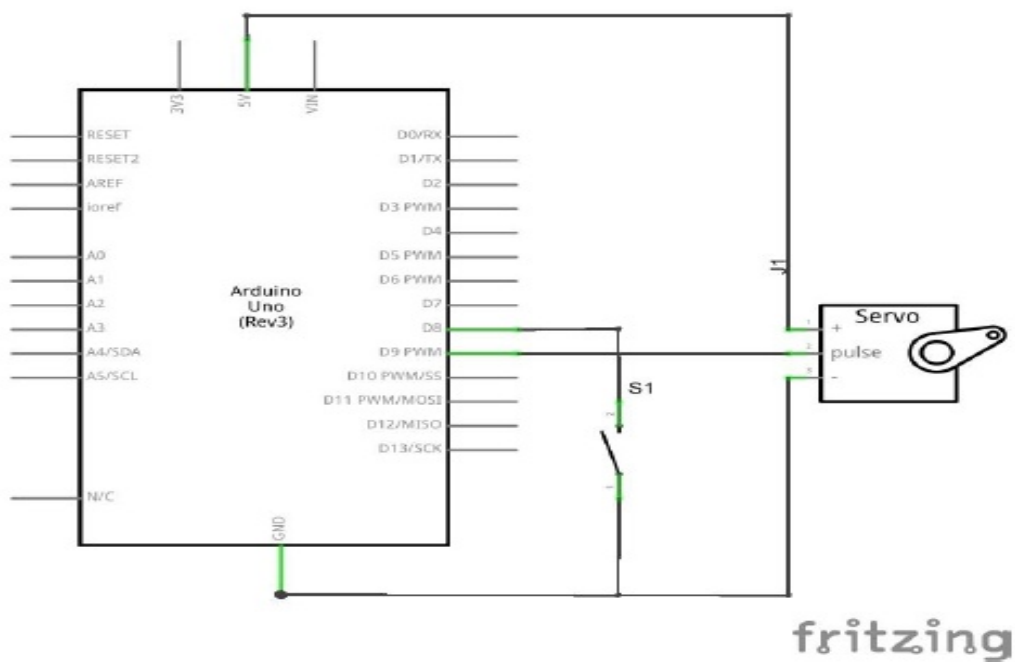


Figure 5 .13: schéma de branchement d’un servomoteur

Exemple d'un simple Programme:

Pour programmer un servomoteur, il est nécessaire de connaître les bases de programmation suivante :

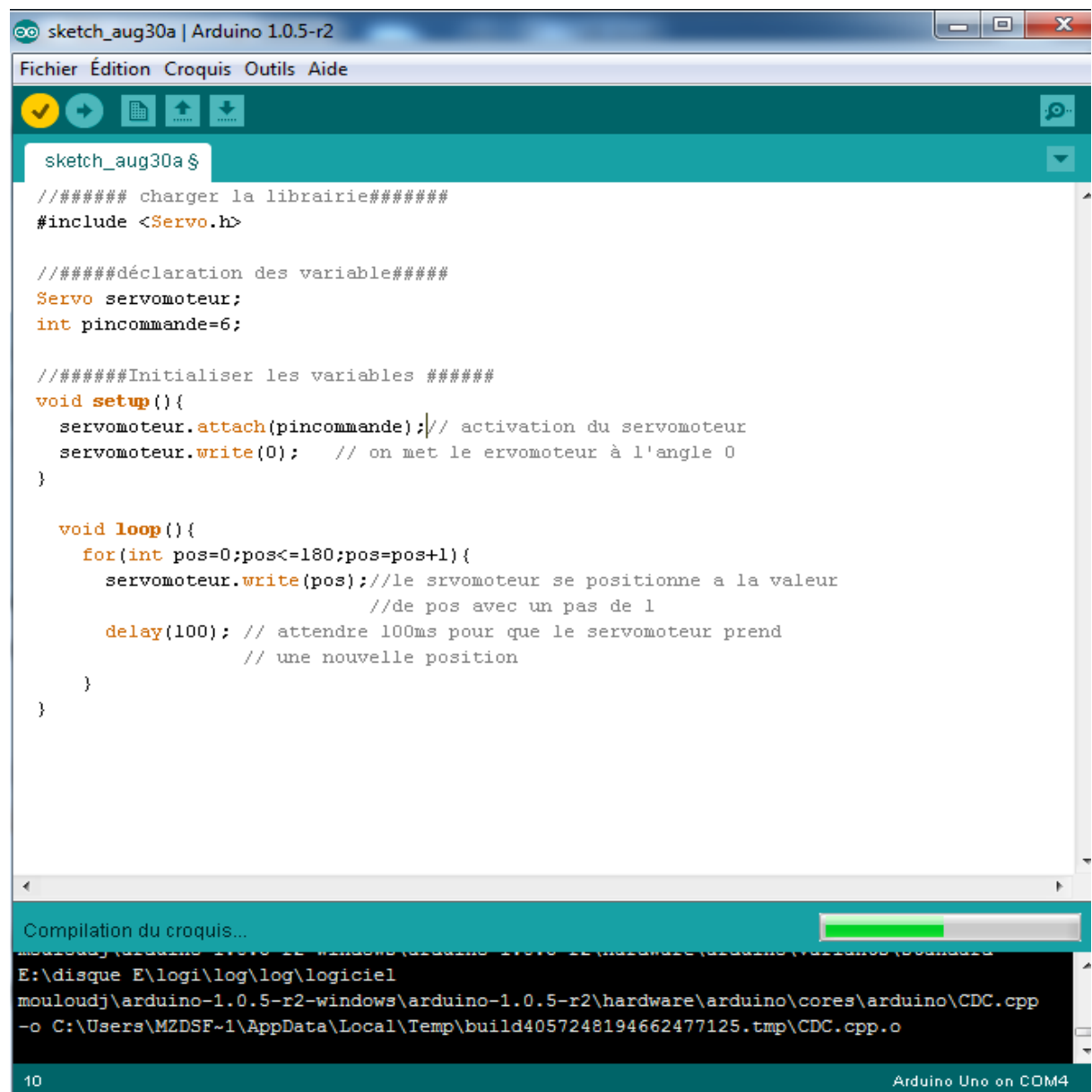
#include <Servo.h> : L'utilisation de la librairie servo.h qui prend en charge la communication de bas niveau entre l'Arduino et le servomoteur.

Servo servomoteur : La création d'un objet de type **Servo**, appelé ici myservo

myservo.attach(6) : Le fil de commande de ce servo sera connecté au PIN 6 et l'objet servomoteur pilotera ce PIN .

myservo.write(80): Demander au servomoteur de se déplacer à l'angle désiré, soit de façon absolue en lui indiquant une valeur entière (80°).

Code source servomoteur



```
sketch_aug30a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide
sketch_aug30a $
##### charger la librairie#####
#include <Servo.h>

#####déclaration des variable#####
Servo servomoteur;
int pincommande=6;

#####Initialiser les variables #####
void setup(){
  servomoteur.attach(pincommande); // activation du servomoteur
  servomoteur.write(0); // on met le ervomoteur à l'angle 0
}

void loop(){
  for(int pos=0;pos<=180;pos=pos+1){
    servomoteur.write(pos); //le srvomoteur se positionne a la valeur
                          //de pos avec un pas de 1
    delay(100); // attendre 100ms pour que le servomoteur prend
                // une nouvelle position
  }
}

Compilation du croquis...
E:\disque E\logi\log\logiciel
mouloudj\arduino-1.0.5-r2-windows\arduino-1.0.5-r2\hardware\arduino\cores\arduino\CDC.cpp
-o C:\Users\MZDSF~1\AppData\Local\Temp\build4057248194662477125.tmp\CDC.cpp.o
10 Arduino Uno on COM4
```

Le moteur DC :

Petit exemple pour commander un moteur DC

Pour commander un moteur dans les deux sens de rotation avec une carte arduino, on fait appelle au circuit L293D qui est un double pont-H.

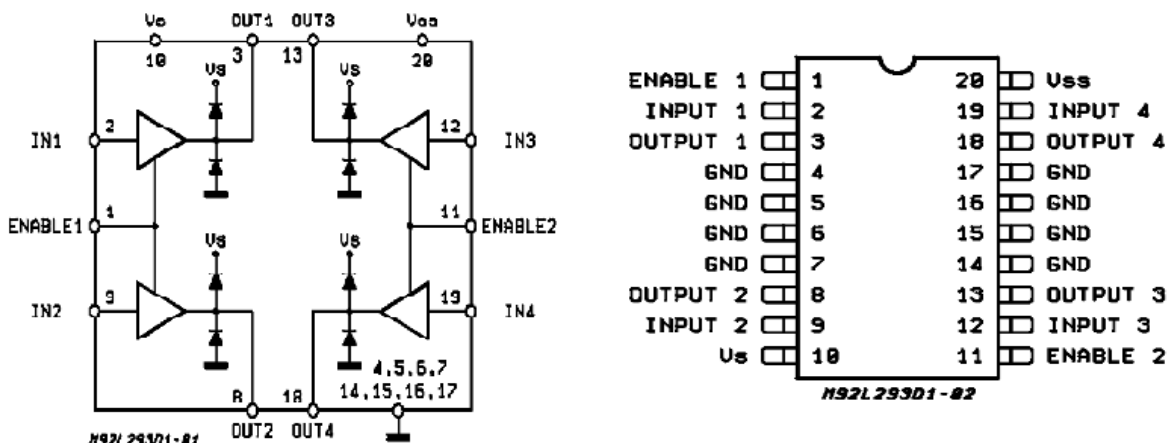


Figure 5.14 : l'architecture interne et externe de L293D

Code source

```
sketch_aug30a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide
sketch_aug30a $
int enablePin = 11; //pour le pwm
int motor_in1Pin = 13; //sens 1
int motor_in2Pin = 12; //sens 2
// ##### Réinitialiser les variables#####
void setup() {
  pinMode(motor_in1Pin, OUTPUT);
  pinMode(motor_in2Pin, OUTPUT);
  pinMode(enablePin, OUTPUT);
}

void loop() {
  SetMotor(255, false);
}

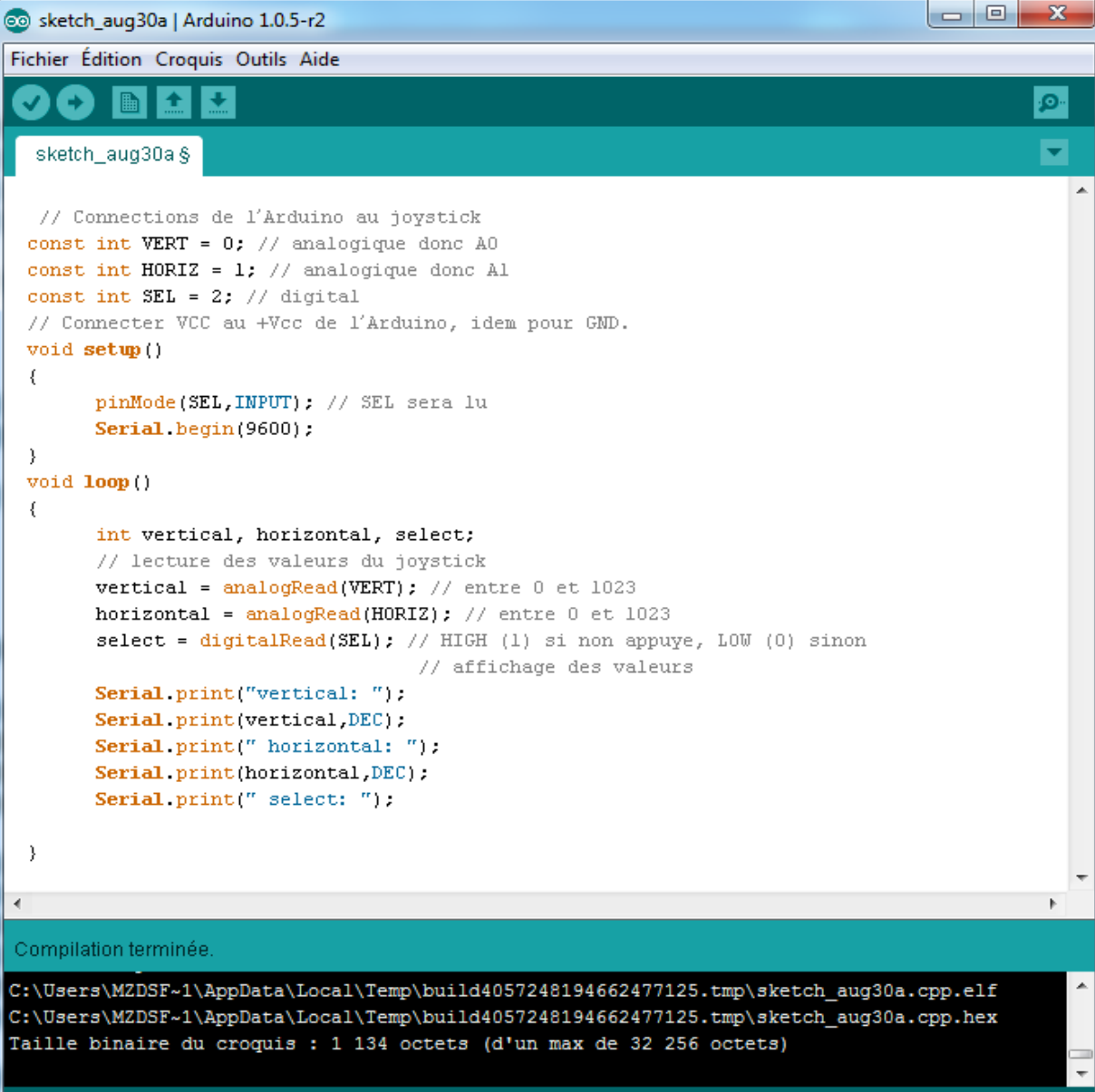
void SetMotor(int speed, boolean reverse)
{
  analogWrite(enablePin, speed);
  digitalWrite(motor_in1Pin, !reverse);
  digitalWrite(motor_in2Pin, reverse);
}

Compilation terminée.
C:\Users\MZDSF-1\AppData\Local\Temp\build4057248194662477125.tmp\sketch_aug30a.cpp.elf
C:\Users\MZDSF-1\AppData\Local\Temp\build4057248194662477125.tmp\sketch_aug30a.cpp.hex
Taille binaire du croquis : 1 134 octets (d'un max de 32 256 octets)
14 Arduino Uno on COM4
```

Le joystick :

Il est constitué de deux potentiomètres l'un prend les valeurs sur l'axe X et l'autre sur Y. et pour savoir à quelle position le joystick se positionne il suffit de lire avec la commande **analogRead(vert)** et **analogRead(horiz)** les valeurs des deux potentiomètres.

Code source :



```
sketch_aug30a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide
sketch_aug30a $

// Connections de l'Arduino au joystick
const int VERT = 0; // analogique donc A0
const int HORIZ = 1; // analogique donc A1
const int SEL = 2; // digital
// Connecter VCC au +Vcc de l'Arduino, idem pour GND.
void setup()
{
    pinMode(SEL, INPUT); // SEL sera lu
    Serial.begin(9600);
}
void loop()
{
    int vertical, horizontal, select;
    // lecture des valeurs du joystick
    vertical = analogRead(VERT); // entre 0 et 1023
    horizontal = analogRead(HORIZ); // entre 0 et 1023
    select = digitalRead(SEL); // HIGH (1) si non appuyé, LOW (0) sinon
    // affichage des valeurs
    Serial.print("vertical: ");
    Serial.print(vertical, DEC);
    Serial.print(" horizontal: ");
    Serial.print(horizontal, DEC);
    Serial.print(" select: ");
}

Compilation terminée.
C:\Users\MZDSF~1\AppData\Local\Temp\build4057248194662477125.tmp\sketch_aug30a.cpp.elf
C:\Users\MZDSF~1\AppData\Local\Temp\build4057248194662477125.tmp\sketch_aug30a.cpp.hex
Taille binaire du croquis : 1 134 octets (d'un max de 32 256 octets)
```

Langage de programmation :

C-arduino :

Résumé Arduino/C

Syntaxes du Programme:

- Chaque instruction se termine par un ";".
- Les accolades "{" et "}" sont les "conteneurs" du code du programme, elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades,
- Les commentaires sont des lignes de texte incluses dans le programme et qui ont pour but de nous informer de la façon dont le programme fonctionne. Ces lignes ajoutées sont ignorées par le compilateur. Les commentaires sont précédés des caractères "//" ou bien encadrés par "" et "".
- Il est formellement interdit de mettre des accents en programmation, sauf dans les commentaires,
- Un nombre en binaire doit être précédé de "B",
- un nombre en hexadécimal doit être précédé par les caractères "0x".

Commandes du programme Arduino:

Fonctions Arduino :

pinMode(pin,b); digitalWrite (pin,b); digitalRead (pin);	boolean boolean	0 = LOW 1 = HIGH
analogWrite (pin,val) analogRead (pin);	byte unsigned int	- 0..255 pins 5,6, 3,11, 9,10 - 0..1023 pins 14 15, 16,17 , 18,19 A0 .. A5
delay (ms); delayMicroseconds ();	unsigned long unsigned int	- 0..~1010 millisecondes (~50 jours) - 0.. 65535 us (~0.06 sec)
millis (temps); micros(temps);	unsigned long unsigned long	- 0..~1010 millisecondes (~50 jours) depuis reset - 0.. ~1010 us (~60 sec)
pulseIn (pin,b);	unsigned long	b=HIGH mesure imp à 1 HIGH attend l'impulsion et mesure sa durée

min(a,b) max(a,b)		choisit le min ou max
constrain (x, a, b) ;		garde la vaeur x entre a et b
map (value, fromLow, fromHigh, toLow, toHigh) ;		applique une règle de 3
bitSet(var,n); bitClear(var,n);	boolean boolean	bits numérotés depuis la gauche. s'applique aussi aux DDRs
bitRead(var,n); bitWrite(var,n,b);	boolean boolean	if (!bitRead(PORTC,bMousD)) - vrai si MousD pressée PORTC&(1<<bMousD) préférable
Serial.begin(9600); Serial.end ();		setup canal série du terminal désactive, rarement utilisé
Serial.print (); Serial.println();		(75) _ 75 (75,BIN) _ 100101 (75,HEX) _ 4B ("Texte 1""2") _ Texte 1"2 '1' code Ascii de 1 = 0x31=49
Serial.write();		(65) _ A ("abcd") _ abcd
Serial.available() Serial.read(); Serial.flush();		if (Serial.available() > 0) { octetRecu = Serial.read(); } vide le tampon
tone (pin, fréqHz); tone (pin, fréqHz,durée); notone ();		démarre un son démarre un son pour la durée spécifiée en ms stoppe le son
randomSeed(valeur); random(max); random(min,max);	int long long	-randomseed (analogRead(A0);) A0 flottant -valeur rendue entre 0 et max-1 -valeur rendue entre min et max-1
attachInterrupt(pin,fct,m); detachInterrupt();		Appelle la fonction s'il y a transition selon mode sur la pin 2 ou 3
interrupt();nointerrupt();		active/désactive toutes les interruptions

Contrôle et conditions

if (condition) { bloc d'instructions; } si vrai on fait On teste et on passe plus loin	if (condition) { instructions ; }
if (condition) { } else { } si vrai on fait, autrement on fait autre chose	if (condition) uneInstruction ; else uneInstruction ;
while (condition) { } on fait et refait tant que la condition est vraie	while (condition) { // instructions ; }

do { bloc d'instructions; } while (condition) ;	do { a++ ;} // a déclaré avant while (a < 5) ;
for (init ; condition ; modif) { bloc d'instructions; }	for (byte i; i<5; i++) { Cligno (); }
byte etat; switch (etat) { case 0: instructions; break; case 1: instructions; break; default: instructions; // optionnel }	enum { Avance, Recule,..} etat; switch (etat) { case avance: instructions; etat = recule; break; case recule: }

Autre instruction

Comparaison		== (égalité), !=(différent), < , >
Boolean		&& , , ! (not) valeur =0 faux ou différent de 0 vrai
Bitwise		& (and) , (or), ^ (xor) , ~ (not) >> , << décale
Opérateurs composés		++ (incréméntation) -- (décréméntation) (à revoir) += (addition composée) -= (soustraction composée) *= (multiplication composée) /= (division composée)

App inventor 2:

Est une application développée par Google, elle est actuellement entretenue par le Massachusetts Institute of Technology(MIT). Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont plus familiers avec les langages de programmation.

Elle est basée sur une interface graphique similaire, grâce à son interface entièrement graphique et à l'absence totale de ligne de code. Elle est particulièrement adaptée à l'initiation des enfants à la programmation, et ce, dès l'école primaire.

La programmation est réalisée sans taper une seule ligne de code, mais simplement en associant et en paramétrant des briques logicielles réalisées.

Google publie l'application le 15 décembre 2010. C'est le centre d'études mobiles au MIT qui gère le support technique de cette application sous le nouveau nom "MIT app Inventor".

A quoi ressemble un programme App inventeur 2:

La programmation se réalise en ligne à l'aide de son navigateur préféré, et sous l'environnement logiciel de son choix (Mac, Linux ou Windows). Seules contraintes, avoir un compte Gmail pour pouvoir y accéder, et un accès à internet évidemment. Les informations sont stockées sur des serveurs distants.

Fenêtres proposées pendant le développement:

Trois (3) fenêtres sont proposées pendant le développement :

- Une pour la création de l'interface homme machine : ce sera l'allure de notre application,
- Une pour la programmation par elle-même : elle permettra, par l'assemblage de blocs de créer le comportement de l'application,
- Une pour l'émulateur qui permettra de tester l'application. L'émulateur permet de remplacer un terminal,
- Réel pour vérifier le bon fonctionnement du programme ;
- La connexion d'un terminal réel sous Android permettra ensuite d'y télécharger le programme.

Site internet :

- [1][MIT App Inventor - Massachusetts Institute of Technology](#)
- [2]http://leroyd.com/articles/module_bluetooth.php
- [3]<https://cours-informatique-gratuit.fr/dictionnaire/android/> consulter septembre 2017
- [4]<https://circuits.io/>
- [5]<http://www.semageek.com/123d-circuits-un-outil-de-conception-electronique-gratuit-en-ligne/> Consulter juillet 2017
- [6]www.arduino.cc Consulter mai 2017
- [7]www.arduino.cc/en/Main/ArduinoBoardUno Consulter mai 2017
- [8]www.didel.com/ Consulter mai 2017
- [9]www.Elecfreaks.com Consulter Aout 2017
- [10]www.electrocouffigna.site.ac-strasbourg.fr/site-php/cours2007/mcc. Consulter mai 2017
- [11]www.locoduino.org/spip.php?article8/ Consulter mai 2017
- [12]www.roboshop.com/ Consulter mai 2017
- [13]www.sparkfun.com/ Consulter 2017
- [14]www.tinkercad.com
- [15]www.technologuepro.com/cours-genie-electrique/cours-27-capteurs-actionneurs-instrumentation/ Consulter mai 2017
- [16]www.technologuepro.com/cours-capteurs-actionneurs-instrumentation-industrielle/ch12-les--differentes-types-de-capteurs.pdf Consulter mai 2017

Mémoires de fin d'étude :

[17]Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde (Présenté par :KRAMA Abdelbasset, GOUGUI Abdelmoumen Le :08/06/2015 Université Kasdi Merbah Ouargla

[18] mémoire fin d'étude : Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde.

[19]-Mémoire de Fin d'Etudes De MASTER PROFESSIONNEL : Commande domotique d'un climatiseur. Présenté par Mr GRIBISSA Aghilas.

[20] mémoire fin d'étude master thème Conception et réalisation d'un robot mobile à base d'Arduino ;

[21]Mémoire de fin d'étude :conception et réalisation d'un système de mesure de flambement et de déformation à base d'une Arduino Uno(présenter par : Mr. KROUBI Tarik).

[44]Réalisation d'un Robot Mobile Avec Évitement d'obstacle Et Trajectoireprogrammé

Ouvrage :

[22] 65.pdf(Robotique Collective : Implémentation d'un comportement d'agrégation stratégique inspiré de comportement des cafards.par : Nehnouh Chakib , Debbat Fatima Département d'Informatique Université Mustapha Stambouli –Mascara).

[23] 2009ORLE2071_0_0.pdf(Contribution a la navigation de robots mobiles :approche par modèle direct et commande prédictive. Par :Nicolas Morette)

[24] 3a_robmob_poly.pdf(Télécom Physique Strasbourg Université de Strasbourg, Bernard BAYLE

[25] 9-Xbee.pdf(Classe de première SI)

[26] Arduino.pdf (Classes de 2nde SI-CIT et de première SI)

[27] Arduino Lesson 14. Servo Motors Created by : Simon Monk

[28] Conception et réalisation de robot suiveur de ligne FSR 2008.

[29] diduino_2014.pdf (titre : Arduino à l'école , v3 - avril 2014, Frédéric Genevey).

[30] Datasheet Atmel ATMEGA328 :

[31] Datasheets L293D (SGS-THOMSON MICROELECTRONICS)

[32] Datasheet Xbee.

[33] Etudes et Réalisations Projet Vierzon 2011.pdf (Calvignac Denizot Galera Malhomme Mendez Pomaret).

[34] Fiche F3 – Contrôler Des Moteurs Et Diriger Un Robot.pdf

[35] Fiche F4 -commander des servomoteurs classiques et a rotation continue.pdf

[36] Filliat Robotique Mobile ENSTAParisTech(titre : Robotique mobile. Par : David FILLIAT École Nationale Supérieure de Techniques Avancées ParisTech)

[37] HCSR04-datasheet.

[38] Introduction to Autonomous Mobile Robots

[39] Microcontrôleur ATMEGA, Janvier 2005 V1.2(AtMega32DocFr.pdf).

[40] Projet Spinos: Conception Et Contrôle D'un Robot Mobile À Vitesses Différentielles. (Présenté par: Julien Beaudry le 5 décembre 2001) ; PFE jbeaudry.pdf .

[41] Robot-thao-SLE2a.pdf Modélisation et commande d'un robot mobile.

[42] Xbee_config.pdf (titre :Module Xbee Pro par : D. MENESPLIER . Décembre 2010).

[43] Xbee Arduino.pdf (Jérôme Abel Jan 21, 2013)

[45] Cours de robotique fondamentale (Projet Coprin INRIA Sophia Antipolis par David.Daney sophia.inria.fr)

[46] Travail de fin d'etudes Locomotion d'un robot mobile (Université de Liège - Faculté des Sciences Appliquées Institut Montefiore par Stéphane Lens, 3ELIN)

[47] Chapitre : I Généralités sur les robots mobiles(5385a8ca0d804.pdf)

[48] John-David Warren & Josh Adams & Harald Molle, "Arduino Robotics", edition Technology In Action.

[49] HC-SR04 - Module de détection aux ultrasons - Utilisation avec Picaxe (pj2-hc-sr04-utilisation-avec-picaxe-1343_2.pdf)

Résumé

L'exploration et l'exploitation d'un environnement inconnu par un ensemble de robots font partie des applications de l'intelligence Artificielle.

Donc la conception et la réalisation d'un robot mobile font l'objet dans ce mémoire. L'objectif est de réaliser un robot capable de se déplacer avec évitement d'obstacles dans l'environnement ou il se trouve d'une manière autonome, ou de commander ce robot a distance via une manette.

Mots clés : Robot mobile, navigation, robot uni-cycle, autonome, obstacle.