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


01/10/2014

I.A.E.S


**Université de Blida 1, SAAD DAHLEB**
**Institut d'Aéronautique et des Etudes Spatiales**


**En Vue de l'Obtention du Diplôme Master**

**Option : Avionique**

**Thème :**

# Robot mobile autonome commandé par un microcontrôleur.


**Réaliser par :**

DJOUDER Kamel.

HASSAK Billal.

**Encadreurs :**

BENSELAMA Zoubir Abdeslam.

BENCHERCHALI M.AMINE.

# Dédicace

***Je dédie ce modeste travail***

*A ma très chère mère,*

*A mon très cher père,*

*A mes frères MENED, ABDERRAHIM, BADREDDINE,*

*A mes cousins et cousines,*

*A toute ma famille proche,*

*A mon binôme BILLAL,*

*A tous mes collègues :*

*Mustapha hakemi, Sofiane Ghebalou, KARIM Cherfaoui, HADJI, REDOUANE Hemi, LARBI, SALIM.*

*Et A tous mes amis et connaissances, proche et lointains.*

*KAMEL*

# Dédicace

***Je dédie ce modeste travail***

*A la mémoire des Martyrs,*

*A ma très chère mère,*

*A mon très cher père,*

*A mes sœurs,*

*A mes frères,*

*A toute ma famille proche,*

*A mon binôme KAMEL,*

*A tous mes collègues : YACIN, HICHAM, LHAJ, MUMUS, REDOIANE.*

*Et A tous mes amis et connaissances, proche et lointains.*

*BILLAL*

# Remerciement

*Avant tout,* ***DIEU*** *merci de nous avoir donné la volonté, la patience et le courage pour accomplir ce travail*

*Nous tenons à formuler notre gratitude et notre profonde remerciements à notre encadreur* ***Mr Benselama Zoubir abdeslem*** *et notre Co-promoteur* ***Mr Bencherchali M.Amine*** *pour leur disponibilité et l'aide précieux, qui nous ont toujours accueillie avec bienveillance, qui n'ont ménagé ni leur temps ni leur efforts pour nous guidé, qu'ils acceptent nos remerciements les plus sincères.*

*Nos remerciements vont aux membres de jury qui ont accepté de juger notre travail,*

*à l'ensemble des enseignements qui ont contribué par leur compétence a notre formation.*

*Enfin, nos remerciements toutes les personnes qui nous ont aidé de loin ou de prés.*

*A tous, un grand merci.*

**Résumé :**

Dans ce mémoire on va faire une réalisation d'un robot mobile autonome muni de capteurs variés et capable de se déplacer par lui-même dans une pièce en tenant compte de son environnement grâce à un capteur de détection des obstacles. Il comporte une étude du système de motorisation, de la perception de l'environnement, et l'élaboration d'une stratégie de commande mise en œuvre par un PC embarqué.

Notre robot fonctionne avec des moteurs électriques a courant continue qui vont générez notre mouvement et qui sont commandées par un signal PWM pour faire varier leur vitesse, le contrôle de ses moteurs se fait à l'aide d'un microcontrôleur qui est intégrer sur une carte Ardnuino Uno et un pont 'H' qui fournit des forts courant aux moteurs pour lui permettre de faire changer le sens de rotation et le sens de courant par un bon agencement d'impulsion.

**Abstract:**

In this thesis we will be a realization of an autonomous mobile robot equipped with various sensors and able to move by himself in a room in the light of its environment using a sensor to detect obstacles. It includes a study of the opener, perception of the environment, and the development of a control strategy implemented by an embedded PC.

Our robot works with electric motors that power continues to go build our movement and are controlled by a PWM signal to vary their speed, control of its engines is done using a microcontroller that is integrated on a map and a bridge Ardnuino Uno 'H' which provides high current to the motors to enable them to change the direction of rotation and the direction of current pulse with a good fitting.

**ملخص:**

في هذه الأطروحة سنكون إدراك من الروبوت المتحرك مستقلة مجهزة مختلف أجهزة الاستشعار و قادرة على التحرك بنفسه في غرفة في ضوء بيئته باستخدام جهاز استشعار للكشف عن العقبات. وهو يشتمل على دراسة الافتتاحية، تصور البيئة، و وضع استراتيجية الرقابة المنفذة من قبل PC المضمنة.

لدينا روبوت يعمل مع المحركات الكهربائية أن السلطة لا تزال تذهب بناء حركتنا و التي تسيطر عليها إشارة PWM تختلف سرعتهم ، ويتم السيطرة على محركاتها باستخدام متحكم التي تتكامل على الخريطة وجسر Ardnuino أونو 'H ' الذي يوفر الحالية المرتفعة إلى المحركات لتمكينهم من تغيير اتجاه دوران واتجاه نبض الحالية مع تركيب جيد.

# Sommaire :

## Chapitre II: La carte Arduino

## Chapitre III : Réalisation et programmation.

## Les figures :

## LISTE DES TABLEAU :

# Chapitre I:

## Généralités sur les robots mobiles autonomes.

## I.1 Introduction:

L'objet de la robotique est l'automatisation de systèmes mécaniques. En dotant le système de capacités de perception, d'action et de décision, l'objectif est de lui permettre d'interagir rationnellement avec son environnement, et de façon autonome. Les robots mobiles sont partout : dans les usines et dans les champs, au fond des mers et dans l'espace, dans les jardins et les salons…..etc. Ils ont une importance économique grandissante. Ils n'ont pas seulement pénétré le monde industriel, ils sont aussi entrain de pénétrer notre vie quotidienne et notre culture, et certains d'entre eux participent au renouvellement de la vision que nous avons de nous-même.

Dans ce chapitre nous devons présenter une généralité sur les robots mobiles autonomes.

## I.2 Bref historique :

C'est vers la fin des années 1960 que, de deux sources complètement différentes, est apparu le concept de robot mobile autonome.

- D'un côté, spécifiquement au Stanford Research Institute, ont été menées des recherches sur les possibilités d'équiper des machines de capacités de déduction et de réaction « logique » à des événements extérieurs. Pour essayer en vraie grandeur les principes développés par les chercheurs, on construit Shakey. C'est une machine à roues, connectée à un gros ordinateur, et qui évolue dans un univers de cubes et de pyramides de tailles et de couleurs différentes. Ses missions : prendre un objet et le porter ailleurs, quelle que soit sa position, absolue dans la salle d'évolution, relative par rapport à d'autres objets. Ses moyens de perception : essentiellement une caméra qui lui permet d'acquérir des images de son environnement. Ses performances : une cinquantaine de minutes pour effectuer une « mission ».
- De l'autre côté, l'industrie nucléaire a besoin de machines permettant d'agir à distance, dans des environnements encombrés et inaccessibles à l'homme ; General Electric développe un quadrupède pour essayer de résoudre ce problème. À peu près en même temps, au début de l'industrie spatiale, s'échafaudent (en particulier au Jet Propulsion Laboratory de Pasadena) les projets de Luna et Mars Rover, destinés à permettre l'exploration des planètes avec des opérateurs restant sur Terre. Laboratoires et industriels, informaticiens et mécaniciens vont continuer leurs travaux en parallèle pendant plusieurs années. En découleront, côté recherche en informatique, les développements de l'intelligence artificielle et, côté industriel, la télé-opération et une partie de la robotique classique. Les robots mobiles autonomes, eux, résultent de la synthèse de ces travaux. Celle-ci s'est effectuée vers la fin des années 1970, avec trois pôles géographiques principaux (le Japon, les États-Unis et la France) et une très large gamme d'applications (du robot domestique au robot militaire ou d'exploration planétaire). Le nombre de chercheurs a crû notablement et leur orientation vers la solution de problèmes industriels s'est

précisée, en particulier vers le domaine des tâches non directement productives.

Nous citons quelques dates importantes dans la robotique mobiles :

- 1947 : Premier manipulateur électrique télé opéré.
- 1954 : Premier robot programmable.
- 1961 : Utilisation d'un robot industriel, commercialisé par la société Unimation (USA), sur une chaîne de montage de General Motors.
- 1961 : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot

**I.3 Définition du robot mobile autonome :**

L'autonomie des robots est un enjeu clé de la robotique moderne. Un système robotisé n'a de raison d'être que s'il est capable d'effectuer un maximum de taches sans la supervision de l'homme. Un système téléguidé représenté bien moins d'intérêt puisque l'opérateur doit prendre en charge les taches de bas niveau comme la navigation au lieu de se concentrer sur les objectifs essentiels de sa mission.

Si l'on veut définir clairement ce qu'est un robot mobile autonome on peut dire qu'il s'agit d'un système capable de:

- se localiser dans son environnement.
- trouver des zones d'intérêt à explorer ou des objets dans son environnement et lies a sa mission.
- de planifier ses actions, pour par exemple définir une trajectoire pour se rendre d'un point 'A' a un point 'B'.
- d'interagir, le cas échéant, avec son environnement pour réaliser certaines tâches .Il peut s'agir par exemple de trouver et d'actionner une poignée de porte pour pouvoir passer d'une pièce a une autre.

**I.4 Exemples d'applications :**

Aujourd'hui, le marché commercial de la robotique mobile est toujours relativement restreint en dehors des robots aspirateurs vendus à plusieurs millions d'exemplaires. Cependant, il existe de nombreuses perspectives de développement qui en feront probablement un domaine important dans le futur. Les applications des robots peuvent se trouver dans de nombreuses activités "ennuyeuses, salissantes ou dangereuses" (3 D's en anglais pour Dull, Dirty, Dangerous), mais également pour des applications ludiques ou de service, comme l'assistance aux personnes âgées ou handicapées.

**Figure1.1** : Exemples de robots commerciaux ou de recherche.

Parmi les domaines d'applications possibles de la robotique, citons :

- La robotique de service (hôpital, bureaux, maison).
- La robotique de loisir (jouets, robot,' compagnon').
- La robotique industrielle ou agricole (entrepôts logistiques, récolte de productions agricoles, mines).
- La robotique en environnement dangereux (spatial, industriel, militaire, catastrophes naturelles).

**I.5 classifications des robots mobiles :**

Une classification est proposée dans la littérature qui définit le degré d'autonomie du robot mobile.

- Véhicule télécommandé par un opérateur qui lui impose chaque tâche élémentaire à réaliser.
- Véhicule télécommandé au sens de la tâche à réaliser. Le véhicule contrôle automatiquement ses actions.
- Véhicule semi-autonome réalisant sans l'aide de l'opérateur des tâches prédéfinies.
- Véhicule autonome qui réalise des tâches semi-définies. Ce type de véhicule pose des problèmes d'un niveau de complexité élevé de représentation des connaissances, de capacité décisionnelle et de génération de plans qui sont résolus à bord dans la mesure du possible.

## I.6 Niveau d'automatisation d'un système :

| Niveaux d'automatisation d'un système | | |
|---|---|---|
| **Niveau d'intervention de l'opérateur humain** | **Type de commande** | **Type de système** |
| Planification des tâches | Stratégie | robot |
| Ordonnancement | Tactique | Télé-opération |
| Conduit (optimisation) | Conduite | Télécommande |
| Asservissement | Reflexe | Téléguidage |

**Tableau1.2**: Niveau d'automatisation d'un système.

La figure 1.2 schématise les niveaux d'automatisation d'un système en fonction des rôles relatifs de l'opérateur humain et de la machine dans les fonctions de gestion du système. S'agissant d'engins mobiles, la classification correspondante permet de positionner les robots mobiles par rapport aux autres types de systèmes.

### I.6.1 planification des taches :

La tâche définie au niveau supérieur et La génération de plan repose sur plusieurs notions:

- La connaissance de la situation initiale.
- La stabilité des connaissances en cours d'exécution.
- La connaissance parfaite de l'impact des actions.

### I.6.2 ordonnancement :

Si les notions de planification n'est plus vérifiée dons l'élaboration du plan doit subir des modifications par rapport d'informations supplémentaires en cours d'exécution par reconnaissance de situation.
Cette dernière établie selon deux types d'applications :

- Si le robot en télé-opérer, alors l'opérateur intervient pour interpréter les informations en sa possession et de prendre les décisions concernant la modification du plan.
- Si le robot possède une forte autonomie, alors la reconnaissance de situation consiste à identifier des propriétés courantes de l'environnement utiles à la mission.

### I.6.3 conduite :

- Le robot est tenu d'utiliser ses propres moyens de planification de la trajectoire et d'évitement d'obstacles pour atteindre la destination qui lui a été indiquée.

- Le robot doit aussi faire de la replanification, s'il se rend compte que le chemin qu'il a pris n'est pas le bon, pour des raisons comme un éloignement trop important de son objectif ou l'arrivée dans une impasse.

### I.6.4 asservissement :

Ici nous parlerons de l'action réflexe qui consiste à faire appel à une ressource sans qu'elle soit prévue par le plan original. Cet appel est réalisé au vue d'une situation particulière. L'exécution de cette action a pour objet de se dégageait des situations locale en espérant se rapprocher au plan initial. L'exemple le plus typique et le contournement des obstacles.

### I.7 Architecture des robots mobiles :

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

- La structure mécanique et la motricité.
- La charge utile.
- La structure de commande.

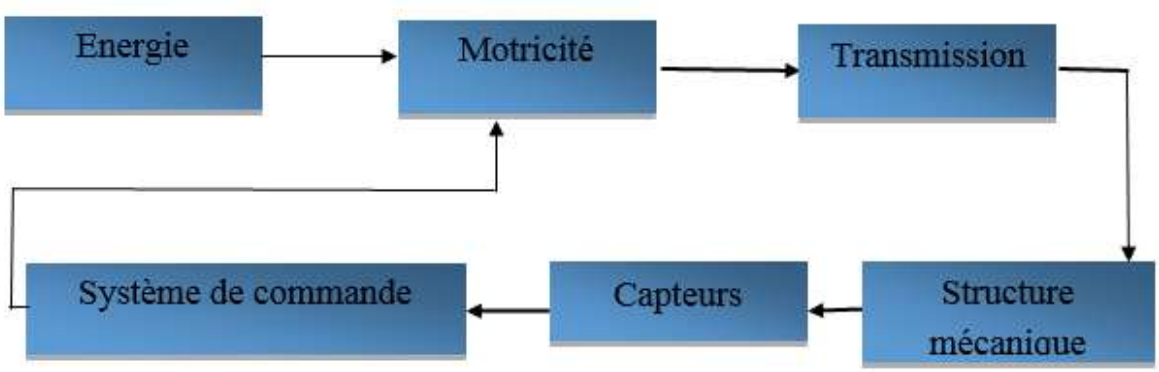



**Figure 1.3** : boucle de commande d'un robot mobile autonome.

### I.7.1 La structure mécanique et la motricité :

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

**a) Structure mécanique :**

- **Les 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.

**Figure 1.3:** Robot mobile à roues.

- **Les mobiles à chenilles:**
  Les chenilles assurent à un mobile, une meilleure adhérence au sol. Elles sont employées lorsque le sol est perturbé, essentiellement en environnement extérieur. La commande est réalisée en imposant une différence de vitesse aux chenilles droites et gauches.

**Figure1.4:** Robot mobile à chenilles.

- **Les mobiles à pattes :**
  Les structures précédentes ne sont plus adaptées dans les cas d'applications sur des terrains avec de grandes différences d'amplitudes où il est nécessaire de choisir les points d'appuis. La solution consiste à discrétiser les points de contact entre le mobile et le sol par des pattes, à l'image des animaux terrestres, qui pour la majorité, disposent de deux, quatre, six, voire plus de points d'appuis. La conception et la commande de tels mécanismes sont complexes.

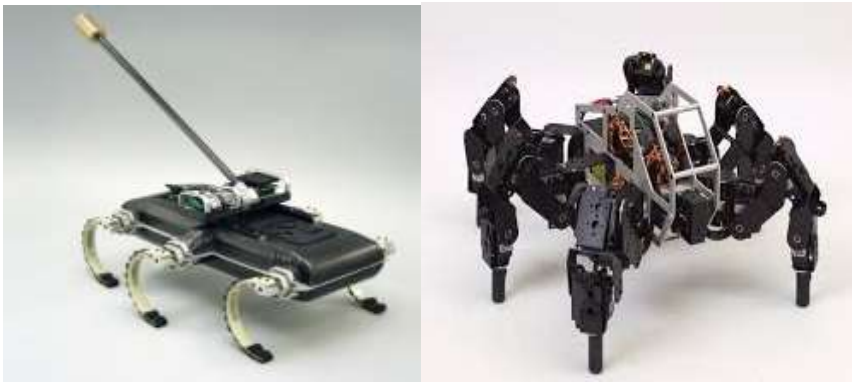


**Figure1.5**:Robot mobile a pattes.

- **mobiles marcheurs :**

  Les robots mobiles marcheurs sont destinés à réaliser des tâches variées dont l'accès au site est difficile, dangereux ou impossible à l'homme. Leur anatomie à nombreux degrés de liberté permet un rapprochement avec les robots manipulateurs. La locomotion est commandée en termes de coordonnées articulaires. Les méthodes de commande des articulations définissent le concept d'allure qui assure le déplacement stable de l'ensemble. Les différentes techniques étudiées se rapprochent de la marche des animaux et notamment de celle des insectes. L'adaptation au support est un problème spécifique aux marcheurs. Il consiste à choisir le meilleur emplacement de contact en alliant l'avance et la stabilité avec l'aide de capteurs de proximité, de contact ou de vision.

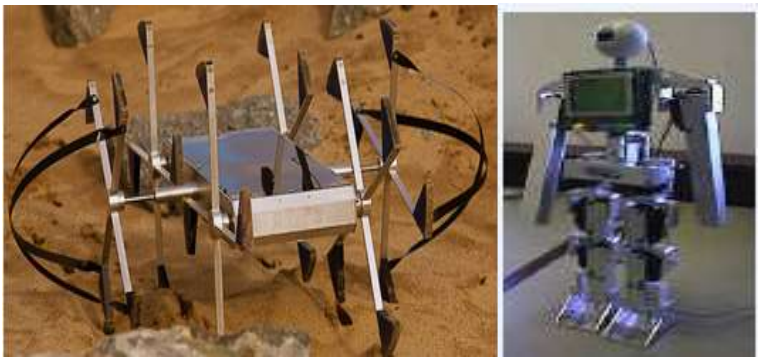


**Figure1.6**:Robot marcheur.

**b) la motricité :**

Les déplacements des robots sont réalisés par des moteurs de types électrique, thermique ou hydraulique. L'énergie électrique la plus fréquemment employée offre l'avantage d'une commande aisée. Par contre le transport et la génération présentent des difficultés. Plusieurs méthodes sont employées :

- Par batteries qui sont soit rechargées périodiquement de manière automatique ou manuelle, soit par un échange avec d'autres lorsqu'elles sont déchargées.
- Par groupe électrogène embarqué dont l'inconvénient constitue la masse élevée. L'énergie de base est alors thermique.
- Par cordon ombilical qui réduit l'autonomie du robot. L'énergie thermique est essentiellement employée par des véhicules de forte puissance comme énergie de base pour la traction ou pour activer un compresseur hydraulique.

**I.7.2 la charge utile :**

Le déplacement d'un robot est dicté par une action à réaliser sur l'environnement. La charge utile concerne directement l'application du robot. Il s'agit en général de véhiculer soit un outil soit un instrument ou des capteurs de mesures afin de détecter un obstacle pour une prise de décision intelligente.

Nous présentons dans cette section les capteurs les plus couramment utilisés en robotique mobile pour les besoins de la navigation ainsi que des modèles probabilistes associés qui seront utilisés dans plusieurs méthodes de navigation.

**a)Les capteurs proprioceptifs :**

Les capteurs proprioceptifs permettent une mesure du déplacement du robot. Ce sont les capteurs que l'on peut utiliser le plus directement pour la localisation, mais ils souffrent d'une dérive au cours du temps qui ne permet pas en général de les utiliser seuls.

- **Odométrie :**

  L'odométrie permet d'estimer le déplacement de la plateforme à partir de la mesure de rotation des roues (ou du déplacement des pattes). La mesure de rotation est en général effectuée par un codeur optique disposé sur l'axe de la roue, ou sur le système de transmission (par exemple sur la sortie de la boite de vitesse pour une voiture). Le problème majeur de cette mesure est que l'estimation du déplacement fournie dépend très fortement de la qualité du contact entre la roue (ou la patte) et le sol. Elle peut être relativement correcte pour une plate-forme à deux roues motrices sur un sol plan de qualité uniforme, mais est en général quasiment inutilisable seule pour un robot à chenille par exemple. Pour limiter ce problème, il peut être intéressant de positionner le codeur optique sur une roue non motrice qui glissera moins. Notons cependant que l'erreur de ces méthodes se retrouve en général principalement sur l'estimation de la direction du robot, tandis que la mesure de la distance parcourue est souvent de meilleure qualité.

- **Les systèmes radar doppler et optiques :**

  Au lieu de mesurer le déplacement par des mesures sur les roues, il est possible d'utiliser un radar pointé vers le sol qui permet de mesurer la vitesse du véhicule par effet Doppler. Il existe aussi des systèmes optiques, basés sur le même principe que les souris d'ordinateur, qui mesurent le déplacement du véhicule en analysant le mouvement relatif du sol. Ces systèmes présentent l'avantage d'être plus précis que la mesure passant par les roues, notamment car ils sont indépendants des dérapages possible de ces roues. Ils sont indépendant en général relativement chers et encombrants et sont assez rares sur les petites plates-formes.

- **Les systèmes inertiels :**

  La mesure de déplacement potentiellement la plus fiable provient de la mesure des accélérations de la plate-forme par des capteurs inertiels. Cette mesure est potentiellement fiable car elle ne dépend pas de la nature locale de l'environnement, cependant les capteurs inertiels sont tous entachés de bruit de mesure qui produit une dérive de l'estimation de la position au cours du temps.

  La qualité des mesures inertielles dépend très fortement du type de capteurs utilisés. Historiquement, les premiers capteurs ont été réalisés à base de systèmes mécaniques et peuvent fournir des mesures extrêmement précises, au prix d'un coût et d'une masse très élevés. Ces dernières années ont vu apparaître de nouvelles technologies de capteurs, notamment basés sur les techniques de micro-électronique, qui ont permis la réalisation de capteurs inertiels 'bas coût' et l'apparition de ces capteurs dans des produits grand public. La précision de ces capteurs est toutefois de quelques ordres de grandeur plus faible, ce qui rend leur utilisation isolée quasiment impossible.

Ces capteurs fournissent toutefois un très bon complément à l'odométrie, notamment pour l'estimation de la direction.

**b) Capteurs extéroceptifs :**

Les capteurs extéroceptifs sont employés en robotique mobile pour collecter des informations sur l'environnement d'évolution du système mobile. Ils sont le complément indispensable aux capteurs proprioceptifs présentés précédemment.

- **Les télémètres :**

  Il existe différents types de télémètres, qui permettent de mesurer la distance aux éléments de l'environnement, utilisant divers principes physiques :

- **Télémètres à ultrason :**

  Les télémètres à ultrason sont historiquement les premiers à avoir été utilisés. Il**s** utilisent la mesure du temps de vol d'une onde sonore réfléchie par les obstacles pour estimer la distance. Ces télémètres sont très simple et peu cher, et sont donc très répandus, mais possèdent de nombreux inconvénients. En premier lieu, deux télémètres voisins ne peuvent être utilisés simultanément, car il est impossible de savoir par lequel des deux télémètres une onde réfléchie a été émise (phénomène de 'cross talk'). Un robot possédant plusieurs télémètres doit donc les activer l'un après l'autre, ce qui entraîne un taux de rafraîchissement global des mesures relativement faible. Ces télémètres possèdent une "zone aveugle", de quelques centimètres, en dessous de laquelle ils ne peuvent détecter les obstacles. Cette zone est due à une temporisation entre l'émission de l'onde sonore et le début de la détection de l'onde réfléchie qui est nécessaire pour ne pas perturber cette mesure. De plus, l'onde réfléchie est très sensible aux conditions environnementales locales. Ainsi, si l'angle entre l'obstacle et la direction de l'onde sonore est trop faible, il n'y aura pas de retour de l'onde sonore et l'obstacle ne sera pas perçu. L'onde de retour dépend également de la texture de l'obstacle. Un mur couvert de moquette pourra par exemple ne pas être détecté. Les télémètres ultrason détectent les obstacles se situant dans un cône relativement large (d'angle au sommet d'environ 30 degrés). Cette caractéristique peut être à la fois un avantage et un inconvénient. C'est un inconvénient car un obstacle détecté n'est pas localisé en angle à l'intérieur du cône de détection, et on obtient donc une mesure de la position relativement imprécise. C'est par contre un avantage car des éléments relativement fins (les pieds de table ou de chaise par exemple) sont détectés dans ce cône, alors qu'ils pourraient ne pas être détectés par des télémètres ayant un angle d'ouverture très fin.

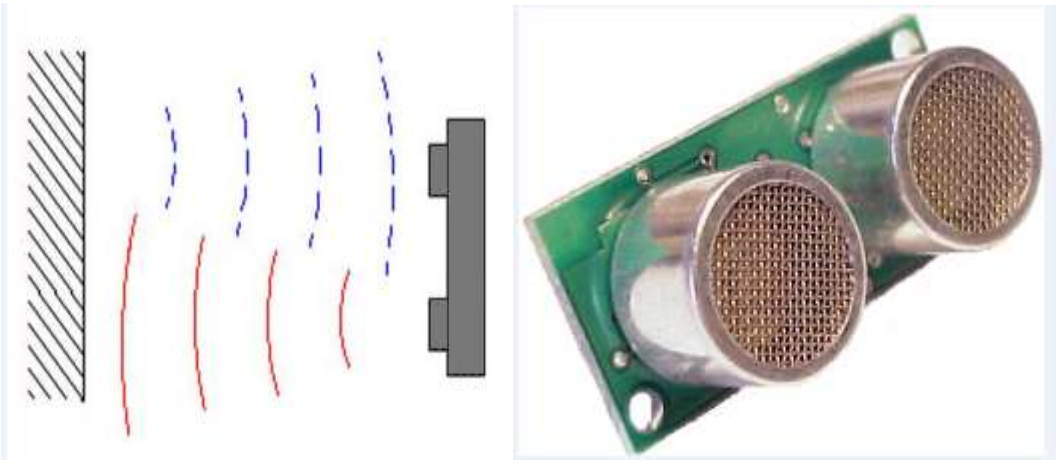


**Figure1.7** : Télémètres à ultrason.

- **Télémètres à infrarouge :**

Les télémètres infrarouges possèdent l'avantage d'avoir un cône de détection beaucoup plus restreint. Ils utilisent une lumière infrarouge au lieu d'une onde sonore pour la détection et peuvent être basés sur différentes techniques qui permettent de recueillir plus ou moins d'information. Il est possible de mesurer simplement le retour ou le non-retour d'une impulsion codée, ce qui permet de détecter la présence ou l'absence d'un obstacle dans une certaine portion de l'espace. Il est également possible de réaliser une triangulation sur le faisceau de retour de l'onde lumineuse, ce qui permet d'avoir une mesure de la distance de l'obstacle.

Les inconvénients de ces télémètres sont liés à leur portée, en général relativement restreinte, et à leur sensibilité aux sources de lumières qui contiennent un fort rayonnement infrarouge. Un projecteur du type de ceux utilisés pour la télévision pointé sur le robot, par exemple, sature en général complètement le récepteur et empêche toute détection d'obstacle. Ils sont également très sensibles à la couleur et à la nature de la surface de l'obstacle (par exemple, ils détectent difficilement les vitres et les obstacles noir mats).

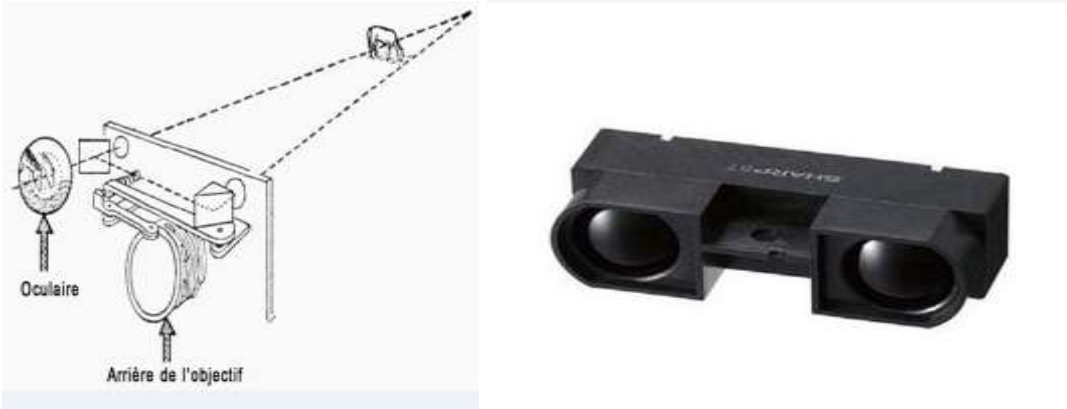



**Figure1.8 :** Télémètres à infrarouge.

- **Télémètres laser :**

Les télémètres les plus utilisés à l'heure actuelle pour des applications de cartographie et de localisation sont les télémètres laser à balayage. Ils utilisent un faisceau laser mis en rotation afin de balayer un plan, en général horizontal, et qui permet de mesurer la distance des objets qui coupent ce plan. Cette mesure peut être réalisée selon différentes techniques soit en mesurant le temps de vol d'une impulsion laser, soit par triangulation. Les télémètres courants ont une bonne résolution angulaire car ils permettent d'obtenir une mesure de distance tous les demi degrés, sur une zone de 180 ou 360 degrés selon les modèles. La mesure est de plus relativement précise (avec un bruit de l'ordre de quelques centimètres) à une distance relativement grande (plusieurs dizaines de mètres). La fréquence d'acquisition est en général de l'ordre de la dizaine de Hertz, voire proche de la centaine pour certains modèles. Ces télémètres sont très utilisés en environnement intérieur car ils fournissent des données abondantes et précises sur la position des objets caractéristiques de l'environnement tels que les murs. Ils possèdent toutefois un certain nombre d'inconvénients. En premier lieu, leur zone de perception est restreinte à un plan et ne permet donc pas de détecter les obstacles situés hors de ce plan (un petit objet posé au sol par exemple). Ils ne peuvent pas non plus détecter les objets ne réfléchissant pas correctement la lumière du laser (en premier lieu les vitres, mais aussi certains objets très réfléchissants, tels que les objets chromés). Pour limiter ces inconvénients, il est possible de les utiliser en conjonction avec des capteurs à ultrason qui ont un cône de détection plus large et qui peuvent détecter les vitres.

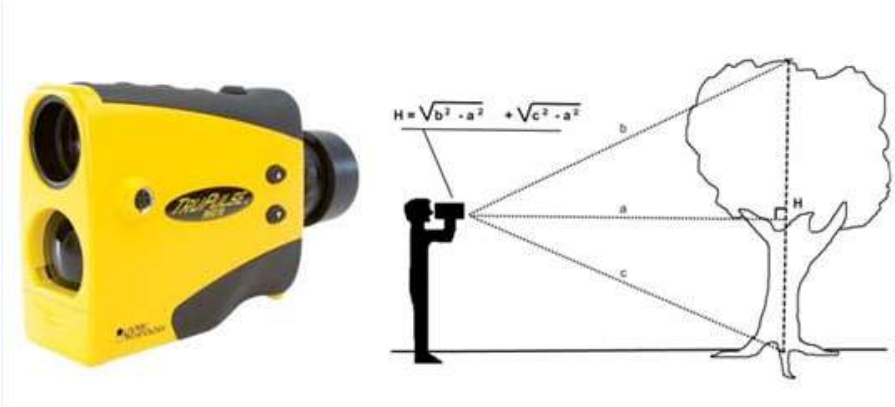



**Figure1.9** : télémètre laser.

- **Les caméras :**

L'utilisation d'une caméra pour percevoir l'environnement est une méthode attractive car elle semble proche des méthodes utilisées par les humains et fournit une grande quantité d'information sur l'environnement. Le traitement des données volumineuses et complexes fournies par ces capteurs est cependant souvent difficile, mais c'est une voie de recherche très explorée et prometteuse pour la robotique.

- **Caméras simples :**
  Une caméra standard peut être utilisée de différentes manières pour la navigation d'un robot mobile. Elle peut être utilisée pour détecter des amers visuels (des points particuliers qui servent de repère, tels que des portes ou des affiches) à partir desquels il sera possible de calculer la position du robot. Si ces amers sont simplement ponctuels, ou de petite taille, il sera en général simplement possible d'estimer leur direction. Dans le cas où les amers sont des objets connus en 2 ou 3 dimensions, il sera en général possible d'estimer complètement la position du robot par rapport à la leur. Elle peut également être utilisée pour détecter des "guides" de navigation pour le robot, tels que des routes ou des couloirs.

- **Caméras stéréoscopiques :**
  Lorsque l'on dispose de deux caméras observant la même partie de l'environnement à partir de deux points de vue différents, il est possible d'estimer la distance des objets et d'avoir ainsi une image de profondeur, qui peut être utilisée pour l'évitement d'obstacles ou la cartographie. Cette méthode suppose toutefois un minimum d'éléments saillants dans l'environnement (ou un minimum de texture) et peut être limitée, par exemple dans un environnement dont les murs sont peint de couleurs uniformes. La qualité de la reconstruction risque également de dépendre fortement des conditions de luminosité. La résolution et l'écartement des deux caméras impose également les profondeurs minimum et maximum qui peuvent être perçues, ce qui peut être limitatif pour la vitesse de déplacement du robot.
  Des techniques similaires peuvent également être utilisées pour estimer la profondeur à partir d'une caméra en mouvement, la difficulté étant alors d'estimer à la fois la profondeur et les positions relatives de la caméra lors de la prise des deux images.

- **Caméras panoramiques :**
  Les caméras panoramiques (catadioptriques) sont constituées d'une caméra standard pointant vers un miroir de révolution (par exemple un simple cône, ou un profil plus complexe qui peut s'adapter à la résolution exacte que l'on veut obtenir sur le panorama). L'image recueillie permet d'avoir une vision de l'environnement sur 360 degrés autour de la camera. Le secteur angulaire vertical observé dépend de la forme du miroir et peut être adapté aux besoins de chaque application. Ce type de caméra est très pratique pour la navigation car une image prise par une caméra panoramique orientée verticalement permet de caractériser une position, indépendamment de la direction du robot. En effet, pour une position donnée et pour deux orientations différentes, la même image sera formée par la caméra, à une rotation autour du centre près, tandis que pour une caméra standard, orientée horizontalement, la scène serait différente.
  Ces caméras sont donc très pratiques lorsque l'on caractérise une position de manière globale, mais peuvent aussi être utilisées pour détecter des amers ou pour estimer le flux optique. Dans ce cas, toutefois, comme la géométrie de l'image formée est relativement complexe et comme la résolution obtenue varie énormément selon la direction observée, les algorithmes doivent être adaptés, ce qui pose un certain nombre de problèmes. Concernant le flux

optique, cependant, les caméras panoramiques possèdent l'avantage de contenir toujours le point d'expansion et le point de contraction dans l'image, ce qui rend l'estimation du mouvement beaucoup plus aisée.

- **Le GPS :**
  Les besoins de localisation étant omniprésents dans de très nombreux secteurs de la vie actuelle, l'idée d'avoir un système de localisation le plus universel possible à donner lieu à l'apparition du Global Positionning System (GPS). C'est un système de balises dont on a placé les balises sur des satellites en orbite terrestre et qui est par conséquent accessible de quasiment partout à la surface du globe. Ce système permet donc d'avoir une mesure de sa position dans un repère global couvrant la terre avec une précision variant de quelques dizaines de mètres à quelques centimètres suivant les équipements.
  Ce système est cependant loin de résoudre tous les problèmes de localisation des robots mobiles. Il fonctionne en effet difficilement dans des environnements urbains, et n'est pas utilisable à l'intérieur des bâtiments. Sa précision est de plus souvent trop faible pour qu'un robot terrestre puisse utiliser ces informations seules. En pratique, il est souvent couplé à un système inertiel qui permet de palier aux pertes du signal GPS et il ne remplace de toute façon pas les capteurs du robot qui lui permettent de percevoir son environnement immédiat, qui constitue la source d'information principale pour la navigation à court terme (par exemple l'évitement d'obstacles, par opposition à la navigation à long terme qui consiste à rejoindre un but distant).
- **Les capteurs tactiles :**
  Les capteurs tactiles sont des détecteurs sensibles au contact. Il existe plusieurs variétés de capteurs tactiles qui peuvent avoir de nombreuses formes et différents niveaux de sensibilité. La plupart de ces capteurs ne sont pas coûteux. Dans un système de détection à distance, ce type de capteur n'est pas le centre d'intérêt mais peut quand même être utilisé pour compléter le système, en assurant par ailleurs une sécurité supplémentaire

### I.7.3 La structure de commande :

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

- Perception de l'environnement.
- Communication homme-machine.
- Commande.

#### a) Perception de l'environnement :

La capacité d'autonomie d'un robot mobile est liée à sa faculté de percevoir et d'interpréter son environnement. Les variations d'états aléatoires de l'espace d'évolution demandent au robot une constante adaptation et des réactions différentes de celles prévues lors de la planification du mouvement. Le module de perception se décompose en plusieurs éléments : le transducteur, la modélisation, la détermination du type d'événement et la prise de décision.

L'acquisition d'un état de l'environnement s'effectue soit par la mesure d'une énergie propre issue de l'espace d'évolution, soit par la réflexion d'énergie émise par une source artificielle. La donnée à traiter est contenue dans les caractéristiques de l'énergie réfléchie.
Il n'est pas question ici de détailler les divers capteurs utilisés en robotique mobile car plusieurs articles de la rubrique Robotique de ce traité traitent de ce sujet.

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

**c) commande :**
Trois modules essentiels composent la partie commande :
- Le planificateur de tâches.
- Le contrôleur d'exécution.
- Le contrôleur d'état du système.

Ces modules peuvent être agencés selon trois architectures différentes :

- **Les contrôleurs hiérarchiques:**

  Ces approches s'appuient sur les techniques de l'intelligence artificielle classique. Elles visent à reproduire le mode de raisonnement humain ou tout au moins une certaine vision du mode de raisonnement humain. Pour cela, ils sont appuyé sur un schéma d'intelligence artificielle conçu dans les années cinquante par Newell et Simon. Le traitement est décomposé en une série d'opérations successives.

- **Les contrôleurs réactifs**:
  L'approche réactive fut introduite par Rodney Brooks en 1986. Dans cette architecture, les senseurs sont directement liés aux actionneurs. Un ensemble de comportements réactifs, fonctionnant en parallèle, contrôle le robot sans utiliser de modèle du monde. Cette architecture se base sur des modèles de l'intelligence animale : l'action globale d'un robot est décomposée en une combinaison de comportements plutôt qu'un long processus de raisonnement.
  Cette architecture permet de réagir rapidement à des modifications de l'environnement. Par contre, les objectifs ne sont pas définis explicitement dans une telle architecture. Ils sont définis implicitement par les comportements programmés. Le comportement global du robot émerge de l'interaction entre les comportements, l'environnement et les perceptions.

- **Les contrôleurs hybrides :**

Les approches hiérarchiques et réactives sont diamétralement opposées. Cependant, chacune présente des caractéristiques intéressantes. Pour cela, des chercheurs ont essayé de les combiner en mettant au point des architectures hybrides permettant notamment d'allier des capacités de raisonnement et de décision de haut niveau, s'appuyant sur des représentations abstraites des connaissances, avec des comportements réactifs garantissant robustesse et flexibilité.

La plupart des contrôleurs actuellement utilisés choisissent une solution intermédiaire entre ces deux approches sous la forme d'une architecture hybride. Cette notion est à distinguer de celle de système hybride en intelligence artificielle (ou système hybride neurosymbolique) qui désigne un système alliant des composantes utilisant des techniques issues de l'intelligence artificielle symbolique classique et des réseaux de neurones artificiels.

L'architecture hybride se compose de deux niveaux. Le premier est chargé des tâches de navigation de haut niveau, telles que la localisation, la cartographie et la planification. Pour cela, il s'appuie sur un second niveau réactif qui est chargé d'exécuter les commandes avec le plus de précision possible et de gérer les éléments non modélisés de l'environnement tels que les obstacles inconnus ou dynamiques. L'action conjointe de ces deux niveaux permet de réagir rapidement face aux variations imprévues de l'environnement, tout en permettant la réalisation d'actions planifiées à plus long terme.

## I.8 Conclusion :

Nous avons vu dans ce chapitre, d'une manière générale, les différentes notions qui permettent la compréhension de la robotique mobile, la classification des robots et ces différents niveaux d'automatisation, leur architecture qui joue un rôle très important en tenant compte de la structure mécanique et leurs divers types.

L'approfondissement dans la science de robotique mobile nécessite une connaissance de certaines notions comme l'autonomie, l'intelligence artificielle les différentes la perception de l'environnement et les capteurs choisis pour une mission précise.

# Chapitre II:

# La carte Arduino.

**II.1-introduction :**

Nous avons vu dans le chapitre précédent les différentes notions qui permettent la compréhension de la robotique mobile, dans ce chapitre on va parler sur un équipement très important c'est l'Arduino qui représente la partie principale de notre robot, toutes les instructions de commandes de notre projet sont chargées dans un microcontrôleur qui est intégrer sur notre carte Arduino Uno l'une de la famille des cartes Arduino.

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, nous allons programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique.

Il existe plusieurs types de cartes Arduino, la différence entre eux soit les caractéristiques technique (la puissance le processeur…etc.), soit l'utilisation.

**II.2 Historique d'Arduino :**

Le projet Arduino est né en hiver 2005. Massimo Banzi enseigne dans une école de Design à Ivrea en Italie, et souvent ses étudiants se plaignent de ne pas avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Banzi en discute avec David Cuartielles, un ingénieur Espagnol spécialisé sur les microcontrôleurs.

Ils décident de créer leur propre carte en embarquant dans leur histoire un des étudiants de Banzi, David Mellis qui sera chargé de créer le langage de programmation allant avec la carte. En deux jours David écrira le code! Trois jours de plus et la carte était créé…Ils décidèrent de l'appeler Arduino (un bar fréquenté par les élèves à proximité de l'école)… Ça devient un hit tout de suite auprès des étudiants. Tout le monde arrive à en faire quelque chose très rapidement sans même avoir de connaissances particulière ni en électronique ni en informatique: réponse à des capteurs, faire clignoter des leds, contrôler des moteurs… Ils publient les schémas, investissent 3000 euros pour créer les premiers lots de cartes: 200. Les 50 premières partent directement à des élèves de l'école. En 2006 5 000 cartes vendues…En 2007 plus de 30 000! En 2011 : >120 000, sans compter les clones ! Prototype de l'Arduino.

**II.3 Carte Arduino :**

Il s'agit d'une carte électronique basée autour d'un microcontrôleur AtMega du fabricant Atmel, dont le prix est relativement bas pour l'étendue possible des applications. Voilà à quoi ressemble la carte que nous allons utiliser

**FigureII.1:** la carte Arduino Uno.

En vue des performances qu'elles offrent, les cartes Arduino sont relativement un peu couteuses, ce qui est un critère majeur pour le débutant.

### II.4 La composition de carte Arduino Uno :

En la carte Arduino se compose de plusieurs composants mais en principe se compose de 8 parties principales comme la figure nous montre

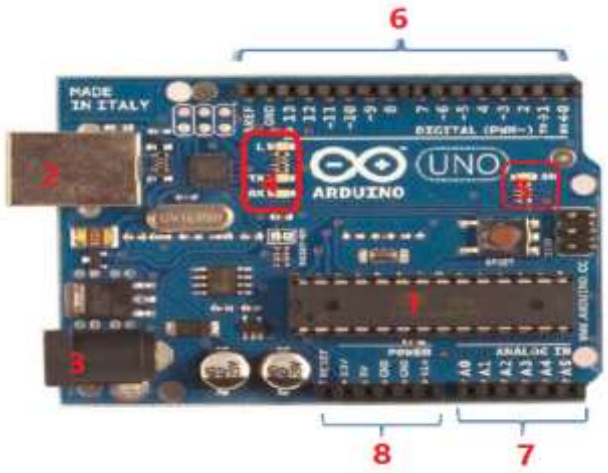



**Figure II.2:** les composants principaux de la carte Arduino Uno.

### II.4. 1. Le microcontrôleur :

Voilà le cerveau de notre carte (en **1**). Le microcontrôleur est un composant électronique programmable, C'est lui qui va recevoir le programme que nous aurons créé et qui va le stocker dans sa mémoire puis l'exécuter. Grâce à ce programme, il va savoir faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur.


**Figure II.3:** le microcontrôleur.

**II.4.2 PORT USB :**

Ce port USB nous permettre de faire connecter notre carte ARDUINO avec un ordinateur. Pour changer les donner entre la carte et l'ordinateur.

**Figure II.4:** porte USB de la carte Arduino Uno.

**II.4.3. Alimentation :**

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB ou bien par une alimentation externe qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte. Pas de danger de tout griller donc! Veuillez seulement à respecter l'intervalle de 7V à 15V (même si le régulateur peut supporter plus, pas la peine de le retrancher dans ses limites)

**Figure II.5:** entrée de l'alimentation externe pour la carte.

**II.4.4. Visualisation :**

Les trois "points blancs" entourés en rouge (**4**) sont en fait des LED dont la taille est de l'ordre du Millimètre. Ces LED servent à deux choses :

- Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel.
  Note : quand on branche la carte au PC, elle clignote quelques secondes.
- Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.


**Figure II.6:** les LED de la carte Arduino Uno.

### II.4.5 .Tester le matériel:

Avant de commencer à programmer, il faut, avant toutes choses, tester le bon fonctionnement de la carte. Car ce serait idiot de programmer la carte et chercher les erreurs dans le programme alors que le problème vient de la carte, nous allons tester notre matériel en chargeant un programme qui fonctionne dans la carte.

Et parfois la carte va y avoir un défaut dans la composition de la carte, pour tester ça il suffit juste de broncher notre carte à l'alimentation et on va voir une LED verte qui clignoté sur la carte ce qui signifier que la carte est bonne.

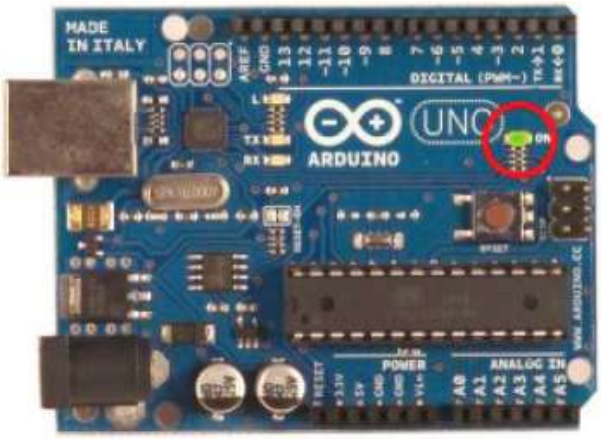



**Figure II.7:** la carte Arduino connecter et alimenter.

### II.4.6. Les entrées-sorties:

- **Broches analogiques :**

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

**Note** : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19.


**Figure II.8:** les entrées analogiques.

- **Autres broches:** Il y a deux autres broches disponibles sur la carte :
  - AREF : Tension de référence pour les entrées analogiques (si différent du 5V).
  - Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

**II.5 Les avantage de la carte Arduino:**

- Pas cher.
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses librairies disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso etc...).
- Existence de « shield » (boucliers en français) : ce sont des cartes supplémentaires qui se connectent sur le module Arduino pour augmenter les possibilités comme par exemple : afficheur graphique couleur, interface Ethernet, GPS, etc...

**II.6 Les caractéristique technique de la carte Arduino:**

- Microcontrôleur : Atmega328.
- Tension d'alimentation interne = 5V.
- Tension d'alimentation (recommandée)= 7 à 12V, limites =6 à 20 V.
- Entrées/sorties numériques : 14 dont 6 sorties PWM
- Entrées analogiques = 6.
- Courant max par broches E/S = 40 mA.
- Courant max sur sortie 3,3V = 50Ma.
- Mémoire Flash 32 KB dont 0.5 KB utilisée par le boot loader.
- Mémoire SRAM 2 KB.
- Mémoire EEPROM 1 KB.
- Fréquence horloge = 16 mhz.
- Dimensions = 68.6mm x 53.3mm.

**II.7 Les types de la carte Arduino :**

Des cartes Arduino ils en existent beaucoup, Peut-être une centaine toutes différentes, nous avions vous montrer les quelles on peut utiliser et celle que nous utiliserions dans ce mémoire.

Il existe beaucoup type des cartes Arduino mais on peut classer les cartes Arduino en trois grandes familles:

- Les cartes Arduino officielles ou classique, compatible hardware et software avec le "forme factor" et l'ide Arduino.

- Les cartes Arduino « compatibles » qui ne sont pas fabriqués par smart projects, mais qui sont totalement compatibles avec les Arduino officielles.
- Les cartes dérivées d'Arduino, compatible avec les Shields Arduino classique (mais pas avec l'ide Arduino de base).

Pour tout ce qui est des cartes Arduino dites "classiques" elles sont basées généralement sur le même microcontrôleur AVR à savoir un ATmega328p du fabricant ATmel.
Par conséquent toutes les cartes utilisant ce microcontrôleur ont les mêmes caractéristiques avec, selon la carte, quelques bonus en plus.

Histoire de ne pas me répéter par la suite voici les spécifications pour toutes les cartes Arduino classiques à base d'Atmega328p.

### II.8 Quelques cartes Arduino les plus utilisé :

#### II.8.1 La carte Uno et Duemilanove :

Nous choisirons d'utiliser la carte portant le nom de « Uno » ou « Duemilanove ». Ces deux versions sont presque identiques.


**Figure II.10:** la carte Uno ou Duemilanove.

#### II.8.2 La carte MEGA :

La carte Arduino Mega est une autre carte qui offre toutes les fonctionnalités des précédentes, mais avec des options en plus. On retrouve notamment un nombre d'entrées et de sorties plus importantes ainsi que plusieurs liaisons séries.


**Figure II.11: La** carte Arduino Mega.

### II.8.3 Seeeduino :

La Seeeduino est une carte Arduino 2009 (Arduino Duemilanove pour les intimes) revue et corrigée à la sauce Seeedstudio.

En gros Seeedstudio a regardé tout ce qui n'allait pas dans l'Arduino 2009 et y a apporté sa petite touche de "seeed".


**Figure II.12: La** carte Seeeduino.

### II.8.4 Arduino Pro (par Sparkfun):

C'est une version 'hyper low cost' de l'Arduino 2009 conçu pour faire des circuits finit, Pas de port USB ou autres, juste le strict nécessaire pour fonctionner.


**Figure II.13**: La carte Arduino Pro.

**II.8.5 Arduino Pro mini (par sparkfun) :**

C'est l'équivalent de l'Arduino mini, c'est une carte Arduino miniaturisée conçue comme la version "normale" pour être intégrée dans un projet finit. Pareil le strict nécessaire mais dans le moins de place possible.

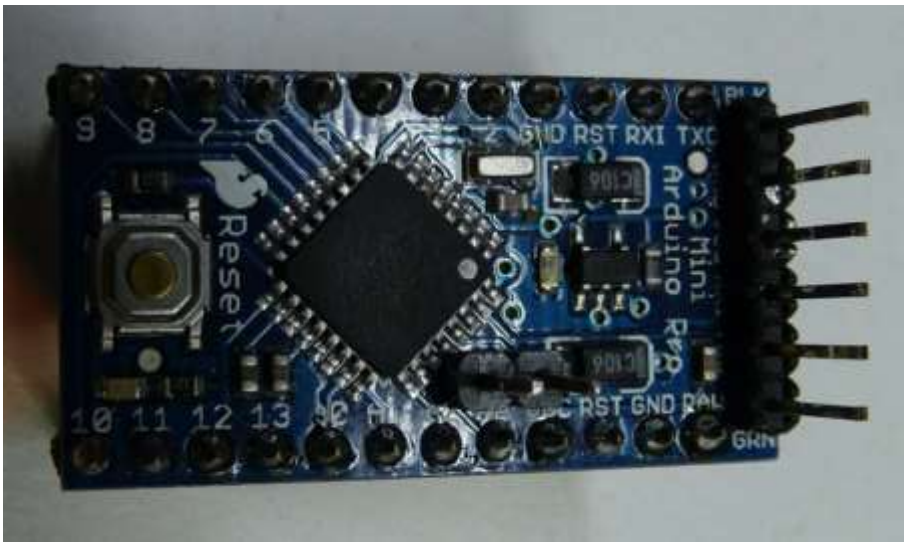



**Figure II.14:** La carte Arduino Pro mini.

**II.8.6 Arduino Blackwidow:**

Carte Arduino avec module wifi intégré, n'est plus en vente sous ce nom mais sous le nom de diamondback (clone). Pratique pour faire des systèmes sur wifi sans avoir à acheter une carte Arduino + shield wifi.

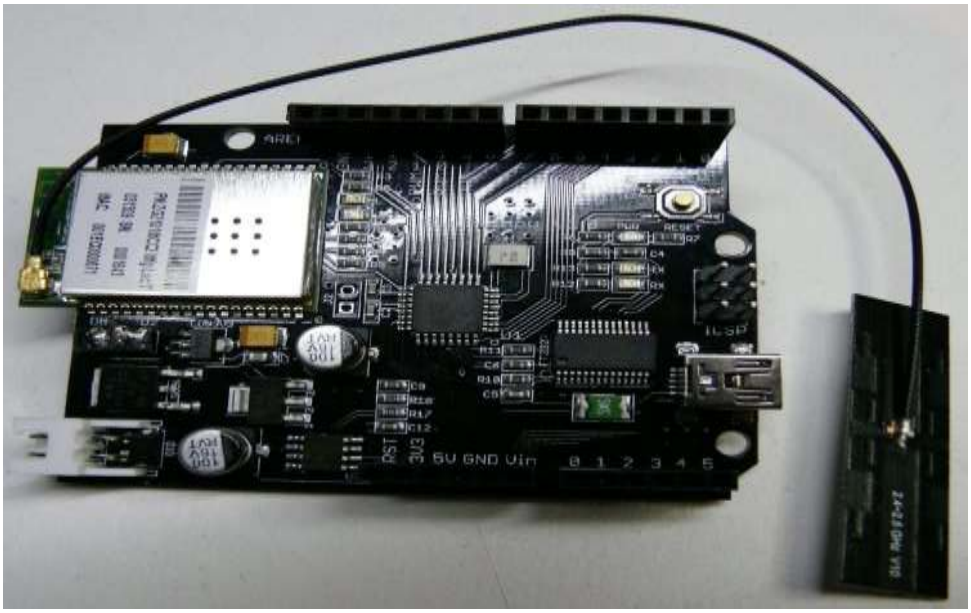


**Figure II.15: la** carte Arduino Blackwidow.

**II.9 Applications :**

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

- contrôler les appareils domestiques.
- fabriquer votre propre robot.
- faire un jeu de lumières.
- communiquer avec l'ordinateur
- télécommander un appareil mobile (modélisme)
- etc.

**II.10 Le logiciel Arduino** :

Puisque notre carte c'est une carte électronique programmable donc sa nécessite une interface de communication (Homme/Machine).

Le logiciel situé sur un ordinateur embarqué va nous permettre de chargé le programmer et l'envoyer à notre carte Arduino Uno.

**II.10.1 la présentation de l'interface de logiciel** :

La page principale de logiciel Arduino se compose principalement de quatre zones voire la figure :

- Le cadre numéro 1 : ce sont les options de configuration du logiciel
- Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes
- Le cadre numéro 3 : ce bloc va contenir le programme que nous allons créer
- Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur.

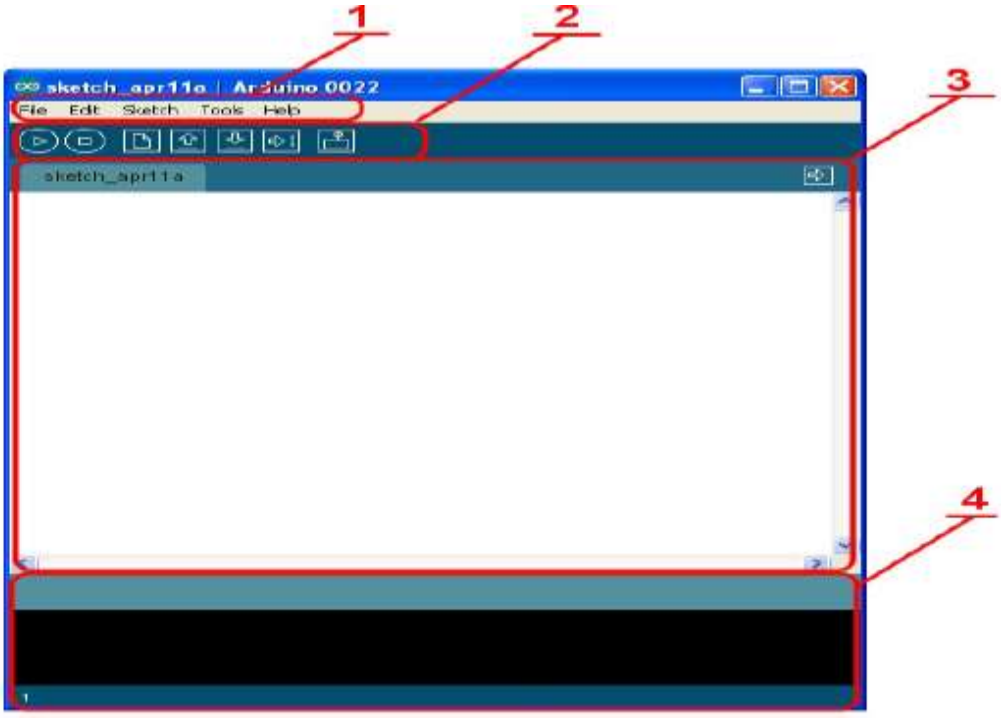



**Figure II.16:** l'interface de logiciel Arduino.

### II.10.2 Quelques instructions sur le logiciel:

- **Le menu FILE :**
  C'est principalement ce menu que l'on va utiliser le plus. Il dispose d'un certain nombre de choses qui vont nous être très utiles :

- New (nouveau) : va permettre de créer un nouveau programme. Quand on appuie sur ce bouton, une nouvelle fenêtre, identique à celle-ci, s'affiche à l'écran
- Open... (ouvrir) : avec cette commande, nous allons pouvoir ouvrir un programme existant
- Save / Save as... (enregistrer / enregistrer sous...) : enregistre le document en cours / demande où enregistrer le document en cours
- Examples (exemples) : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existants ; avec ça, vous pourrez vous aider pour créer vos propres programmes


**Figure II.17:** Le menu File**.**

➢ **Les boutons :**

Il existe 7 boutons comme la figure nous montre :


**Figure II.18:** les boutons de l'logiciel Arduino.

- Bouton 1 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.
- Bouton 2 : Créer un nouveau fichier.
- Bouton 3 : Sauvegarder le programme en cours.
- Bouton 4 : On n'y touche pas pour l'instant.
- Bouton 5 : affichez les résultats analogiques (monitor série).
- Bouton 6 : Charger un programme existant.
- Bouton 7 : Compiler et envoyer le programme vers la carte.

➢ **Choisir la carte Arduino Uno :**

Le choix de la carte est important car chaque carte a des spécifications par rapport à une outre, et parfois il va y avoir une incompatibilité quand on va lancer le programme dans la carte et pour notre mémoire on va utiliser la carte Arduino Uno et ça se fait avec le clique sur le menu Tools /Board :''Arduino Uno'' /Arduino Uno.

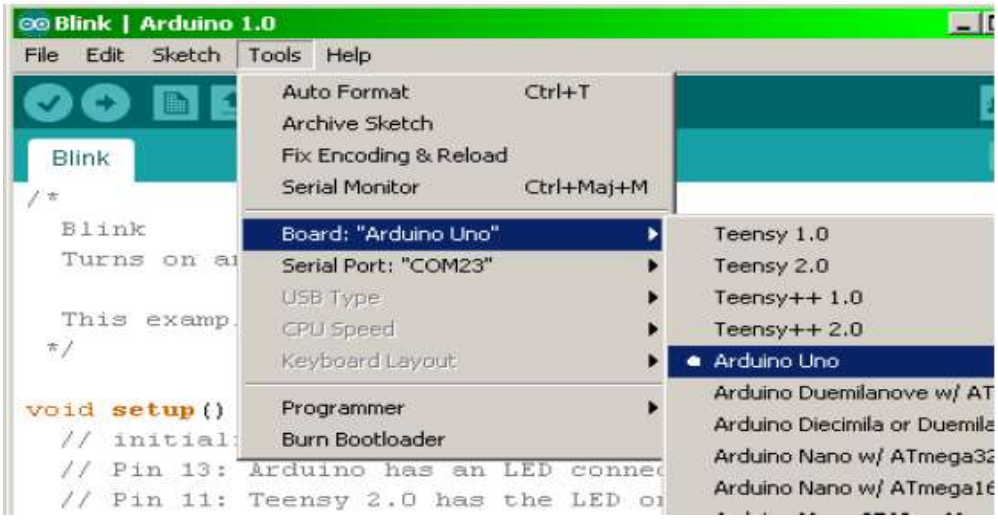



**Figure II.19:** la sélection de la carte Arduino Uno.

### II.10.3 la programmation

Le programme c'est l'ensemble des variables et des constantes et des instructions bien organiser dédier à réaliser une tâche bien définie et avant de réaliser le programme on va voir une vision brève sur l'ensemble des instructions, les variables, les constants, les structures, les fonctions utiliser dans le langage Arduino.

- **Structure :**

| **Fonctions de base**<br>Ces deux fonctions sont Obligatoires dans tout Programme en langage Arduino<br>:<br>• void setup ()<br>• void loop() | **Structures de contrôle**<br>• i f<br>• if...else<br>• for<br>• Switch case<br>• while<br>• do... While<br>• break<br>• continue<br>• return<br>• goto | **Syntaxe de base**<br>• ; (point-virgule)<br>• {} (accolades)<br>• / / (commentaire sur une Ligne)<br>• /* * / (commentaire sur Plusieurs lignes)<br>• #define<br>• #include |
|---|---|---|
| **Opérateurs arithmétiques**<br>• = (égalité)<br>• + (addition)<br>• - (soustraction)<br>• * (multiplication)<br>• / (division)<br>• % (modulo) | **Opérateurs de comparaison**<br>• == (égal à)<br>• != (différent de)<br>• < (inférieur à)<br>• > (supérieur à)<br>• <= (inférieur ou égal à)<br>• >= (supérieur ou égal à) | **Opérateurs booléens**<br>• && (ET booléen)<br>• \| \| (OU booléen)<br>• ! (NON booléen) |
| **Pointeurs**<br>• * pointeur<br>• & pointeur<br>Voir également :<br>• Manipulation des Ports | **Opérateurs bit à bit**<br>• & (ET bit à bit)<br>• \|(OU bit à bit)<br>• ^(OU EXCLUSIF bit à bit)<br>• ˜(NON bit à bit)<br>• << (décalage à gauche)<br>• >> (décalage à droite) | **Opérateurs composés**<br>• ++ (incrémentation)<br>• -- (décrémentation) (à Revoir)<br>• += (addition composée)<br>• -= (soustraction composée)<br>• *= (multiplication Composée)<br>• /= (division composée)<br>• &= (ET bit à bit composé)<br>• \|= (OU bit à bit composé) |

➢ **Variables et constantes :**
Les variables sont des expressions que vous pouvez utiliser dans les programmes pour stocker des valeurs, telles que la tension de sortie d'un capteur présente sur une broche analogique.

| **Constantes prédéfinies** | **Types des données** | **Conversion des types De données** |
|---|---|---|
| Les constantes prédéfinies du langage Arduino sont des valeurs particulières ayant une signification spécifique.<br><br>• HIGH \| LOW<br>• INPUT \| OUTPUT<br>• true \| false<br>A ajouter : constantes Décimales prédéfinies<br><br>**Expressions numériques**<br>• Expressions numériques entières<br>• Expressions numériques à virgule | Les variables peuvent être de type variés qui sont décrits ci-dessous.<br><br>Synthèse des types de Données Arduino<br>• boolean<br>• char<br>• byte<br>• in t<br>• unsigned in t<br>• long<br>• unsigned long<br>• floa t (nombres à virgules)<br>• double (nombres à virgules)<br>• Les chaînes de caractères<br>• objet String NEW<br>• Les tableaux de variables<br>• le mot-clé void (fonctions)<br>• Word<br>• PROGMEM<br>Voir également :<br>• Déclaration des variables<br>Pour info : les types de données avr-c | • char ()<br>• byte ()<br>• int ()<br>• long ()<br>• float ()<br>• word ()<br><br>**Portée des variables et Qualificateurs**<br>• Portée des variables<br>• static<br>• volatile<br>• cons t<br><br>**Utilitaires**<br>• sizeof () (opérateur Sizeof)<br><br>**Référence**<br>• Code ASCI I |

➢ **Fonctions :**

Les fonctions sont des petites programmes enregistrer sur le toolbox de dans le but les utiliser directement dans le besoin.

| **Entrées/Sorties Numériques** | **Temps** | **Trigonométrie** |
|---|---|---|
| • pinmode (broche, mode)<br>• digitalwrite (broche, valeur)<br>• int digitalread(broche)<br><br>**Entrées analogiques**<br>• int analogread (broche)<br>• analogreference (type)<br><br>**Sorties "analogiques" (génération d'impulsion)**<br>• analogwrite (broche, Valeur) – PWM<br><br>**Entrées/Sorties Avancées**<br>• tone ()<br>• notone ()<br>• shiftou t(broche, Brochehorloge, ordrebit, valeur)<br>• unsigned long Pulsein (broche, valeur)<br>**Communication**<br>• Seria l | • unsigned long millis ()<br>• unsigned long micros ()<br>• delay (ms)<br>• delaymicroseconds (us)<br>**Math**<br>• min (x, y)<br>• max (x, y)<br>• abs (x)<br>• constrain (x, a, b)<br>• map (valeur, tolow, Fromhigh, tolow, Tohigh)<br>• pow (base, exposant)<br>• sq (x)<br>• sqr t(x)<br>Pour davantage de fonctions mathématiques, voir aussi la librairie math.h : log, log10, Asin, atan, acos, etc...<br>**Nombres randomisés (hasard)**<br>• randomseed (seed)<br>• long random(max)<br>• long random (min, max) | • sin (rad)<br>• cos (rad)<br>• tan (rad)<br>**Bits et Octets**<br>• lowbyte ()<br>• highbyte ()<br>• bitread ()<br>• bitwrite ()<br>• bitse t ()<br>• bitclear ()<br>• bi t ()<br>**Interruptions Externes**<br>• attachinterrup t(interrupti On, fonction, mode)<br>• detachinterrup (interruption)<br>**Interruptions**<br>• interrupts ()<br>• nointerrupts ()<br>Voir également la librairie Interrupt.h. |

**II.11 Conclusion :**

Par sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle

Aujourd'hui l'utilisation des cartes Arduino est vaste et efficace grâce à sa simplicité et l'efficacité comme on va voir dans cette application sur le robot mobile autonome.

# Chapitre III:

## Réalisation et programmation.

## III.1 Introduction :

Dans ce chapitre nous allons aborder la partie pratique qui concerne la réalisation du robot mobile autonome, de points de vue composants nécessaires, branchage asservissement et commande.

L'objectif de notre robot c'est de créer un mouvement on évite les obstacles à l'aide d'un détecteur de distance qui surveille l'environnement et qui envoi les données à la carte Arduino Uno, cette dernière suite d'un programme stocké, elle va gérer le mouvement des deux moteur a courant continue par l'intermédiaire d'un pont H dans le but d'avoir une rotation des moteurs dans les deux sens.

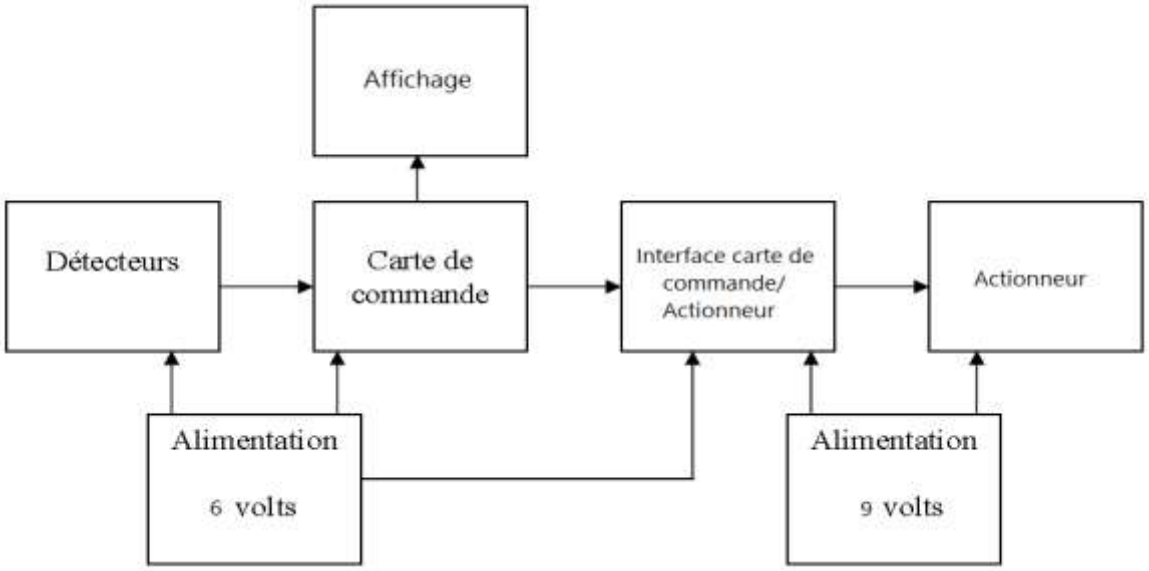
Les applications doivent être assurées par le robot :

- Le robot est autonome.
- Il détecte les obstacles par rayon infrarouge
- Gestion électronique par un microcontrôleur intégrer sur une carte Arduino Uno.
- Utilisation d'un circuit intégré pour la commande des moteurs.

## III.2 Compositions du robot :

- Deux roues motrices indépendantes.
- une carte Arduino Uno pour minimiser le nombre de composants.
- Le pilot moteur (pont H).
- Le servomoteur.
- Capteur de distance infrarouge.
- écran d'affichage LCD.
- Batteries (6v et 9v).

## III.3 Schéma synoptique :




**FigureIII.1** Schéma synoptique de l'ensemble des cartes.

### III.3.1 Détecteurs :

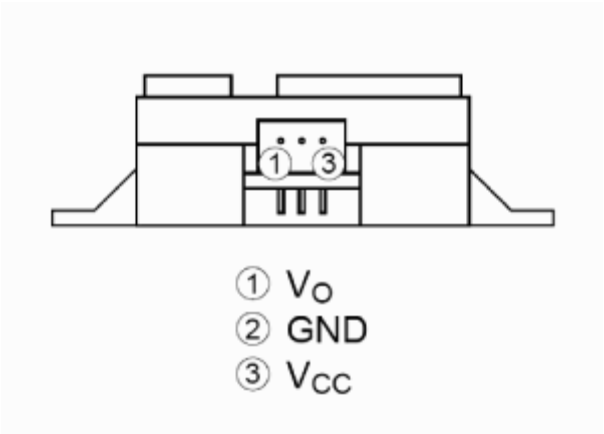
#### III. 3.1.1 Capteur de distance infrarouge:

Le capteur GP2D12 est un capteur de distance :

- opérant dans une plage de mesure de 10cm à 80cm (bon compromis pour un robot...).
- infrarouge (utilise un système optique lumineux dans une longueur d'onde invisible à l'œil nu).
- analogique (tension de sortie entre 0 et 2,4V).
- non-linéaire (la tension de sortie n'est pas directement proportionnelle à la distance).
- fonctionnant sous 5V.
- Alimentation entre +4.5V et 5.5V (idéal avec une carte Arduino).

Le brochage est standard avec 3 broches :

- Vcc = +5V.
- GND = 0V.
- tension de sortie Vo.

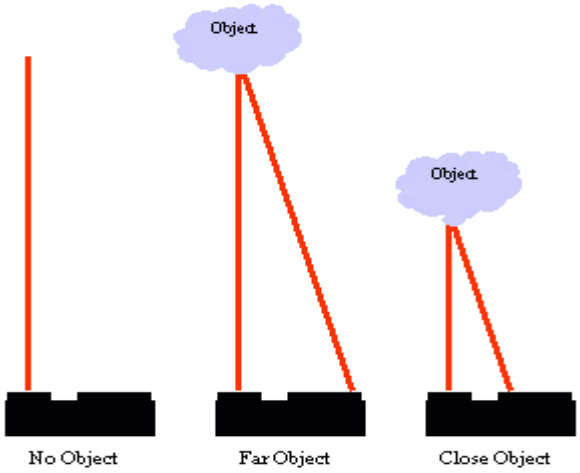



**FigureIII.2** capteur GP2D12

#### III.3.1.1.1 Principe de fonctionnement :

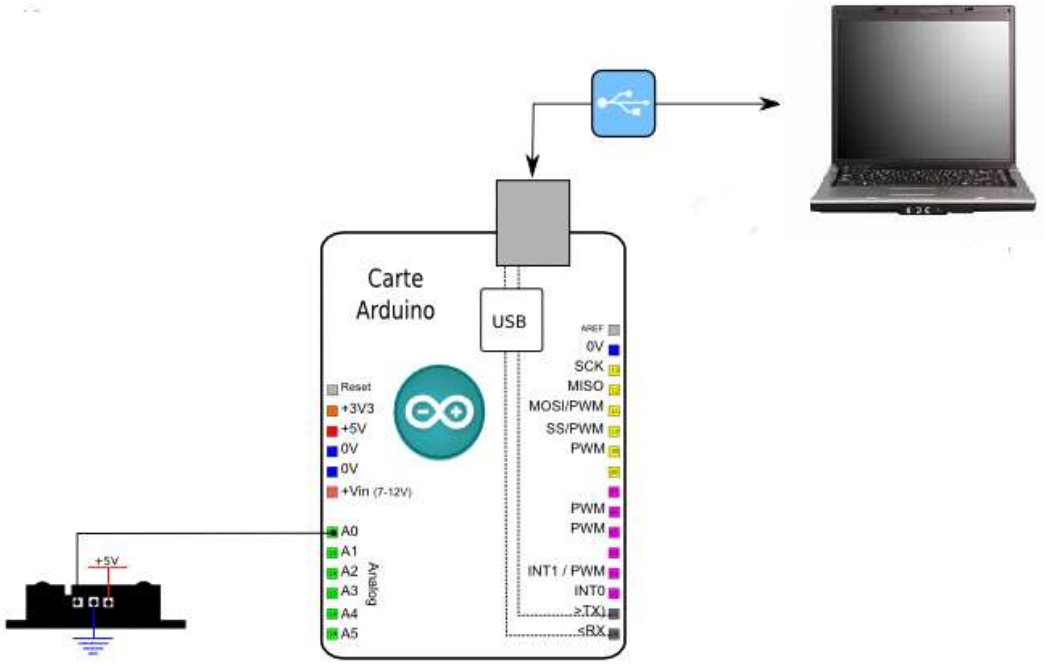
Ce capteur utilise un ingénieux système optique :

- une LED infrarouge émet un rayon infrarouge invisible à l'œil nu qui est réfléchi par les objets.
- une barrette photo réceptrice reçoit le rayon réfléchi, ce qui permet d'en déduire l'angle de réflexion et donc la distance.




**FigureIII.3** Principe de fonctionnement du capteur GP2D12.

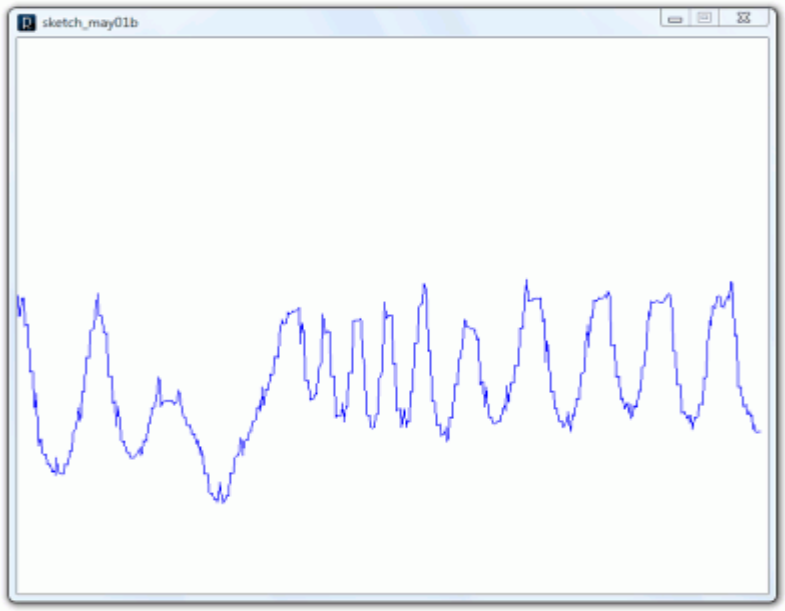
### III.3.1.1.2 Montage de test avec une carte Arduino :




**FigureIII.4** Arduino avec capteur.

On utilise la visualisation du résultat sous forme graphique de la conversion analogique dans une interface de traitement sur l'ordinateur :

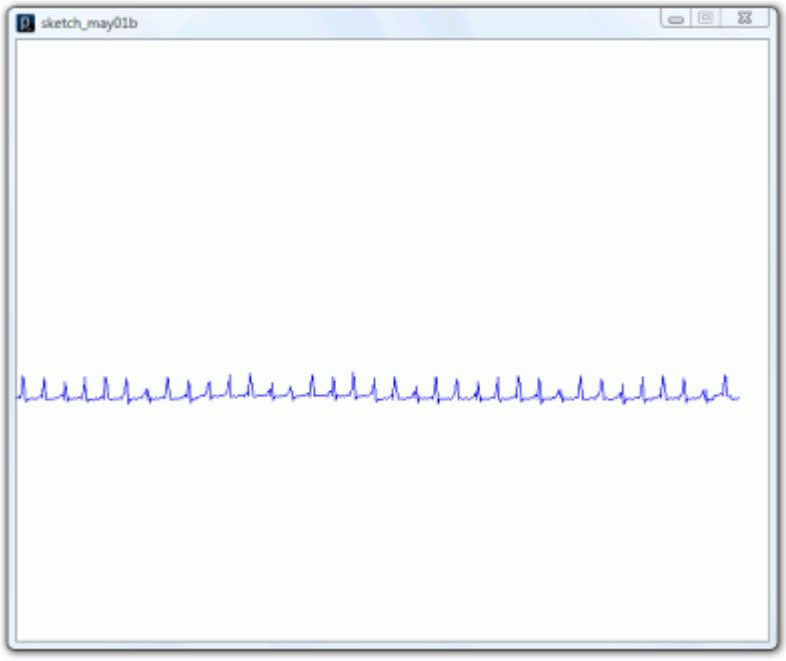
Réactivité du capteur à la mobilité de l'objet :




**FigureIII.5** Réponse du capteur a la mobilité.

- Réponse rapide du capteur.

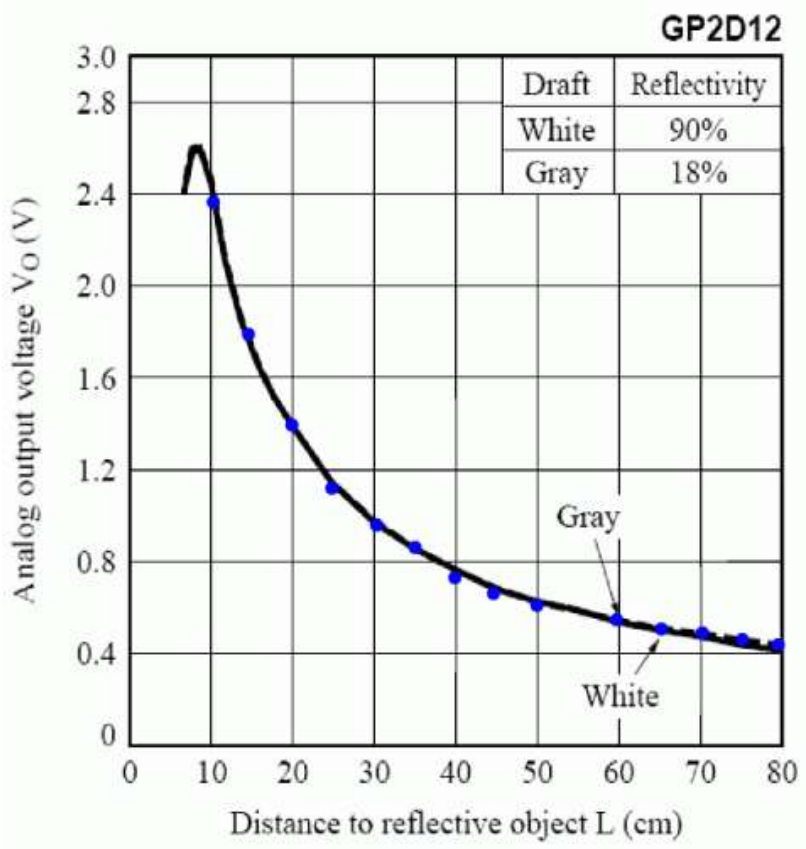
Réponse du capteur à une distance fixe :




**FigureIII.6** Réponse du capteur à une distance fixe**.**

### III.3.1.1.3 Tension en fonction de la distance :

Voici les mesures empiriques retrouvées (reportées en bleu sur la courbe théorique) :

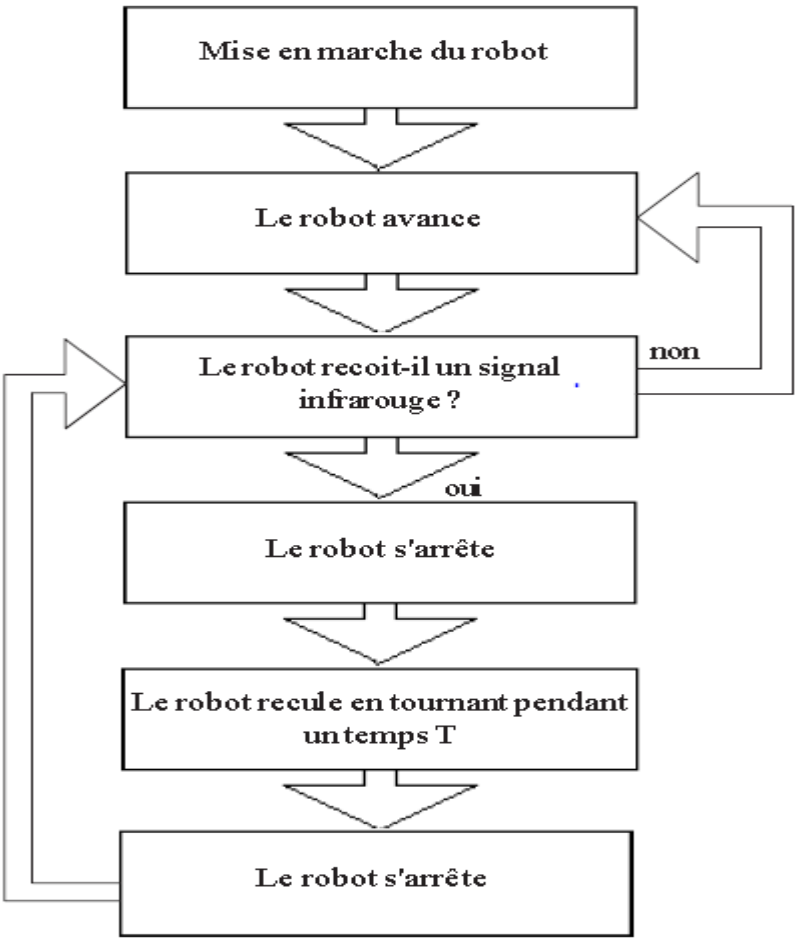



**FigureIII.7** La tension en fonction de la distance.

On remarque que :

- la courbe réelle est proche de la courbe théorique
- la sensibilité est fortement diminuée au-delà de 40cm

### III.3.1.1.4 Mise en marche du robot :




**FigureIII.8** Mise en marche du robot.

### III.3.2 Carte de commande :

Le microcontrôleur est un élément primordial d'un robot, car il permet l'exécution de logiciels informatiques donnant son autonomie au robot. On trouve souvent dans un robot des modèles à très faible consommation, notamment pour des robots de petite taille, qui ne peuvent pas emporter avec eux une source d'énergie importante.

L'intérêt principal de carte Arduino Uno est leur facilite 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 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 TWI/I2C, exploitation de servomoteurs ...etc. (pour plusieurs informations voir chapitre2 et annexe)

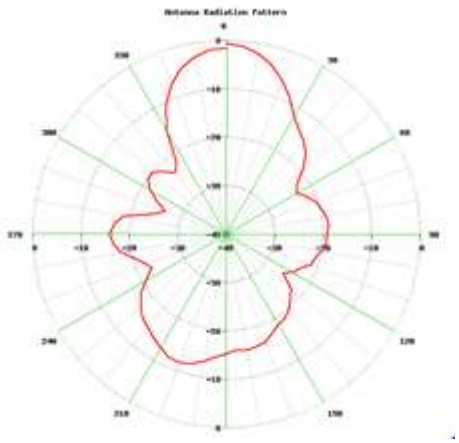
### III.3.3 les actionneurs :

Généralement, un actionneur peut être considéré comme un constituant d'un système mécanique (exemple : bras, patte, roue motrice...) et correspond à un degré de liberté.

Les interfaces haptiques permettent au robot de saisir des objets. Les moteurs permettent à des éléments mobiles de bouger suivant un ou plusieurs degrés de liberté.
Dans notre cas les actionneurs se sont des moteurs électriques à courant continue.

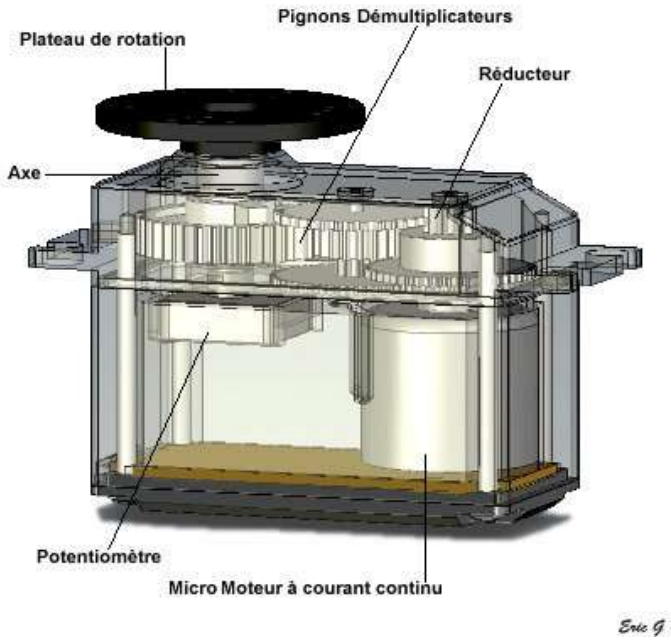
### III.3.3.1 servomoteur :

Produire une image de l'environnement dans un diagramme polaire. Pour cela faire tourner le capteur de distance autour d'un axe vertical et mesurer la distance pour chaque angle de rotation. Il y a besoin d'un moteur dont on connait la position angulaire.
On peut utiliser un servo moteur de modélisme. Commande de position par largeur d'impulsion.



**FigureIII.9** diagramme polaire d'un servomoteur.

Les servomoteurs sont des ensembles mécaniques composés d'un moteur courant continu, d'un servomoteur, d'un potentiomètre, d'un réducteur et d'un système électronique d'asservissement. Le moteur peut tourner selon un certain angle, qui dépend du servomoteur : il y a une butée mécanique qui l'empêche d'aller plus loin, et le potentiomètre permet de connaitre la position de l'axe du servomoteur, c'est à dire l'angle que forme l'axe du servomoteur avec la butée.



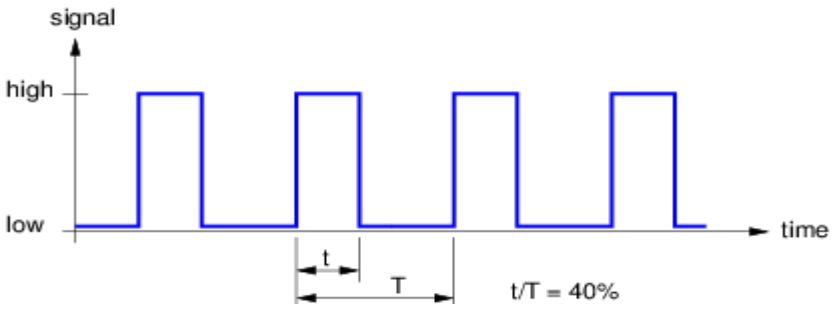

**FigureIII.10** Servomoteur.

Lorsque l'on envoie une commande au moteur, l'asservissement alimente le moteur jusqu'a que la commande correspond à la tension du potentiomètre, c'est à dire à la bonne position.

Si on remplace le potentiomètre par une résistance fixe, qui correspond à un angle de 0 degrés, lorsque l'on lui enverra un signal pour aller dans un sens ou dans l'autre, il va tourner dans ce sens, mais il se croira toujours à 0 degrés, et va continuer à tourner indéfiniment.

### III.3.3.1.1Asservissement de la vitesse :

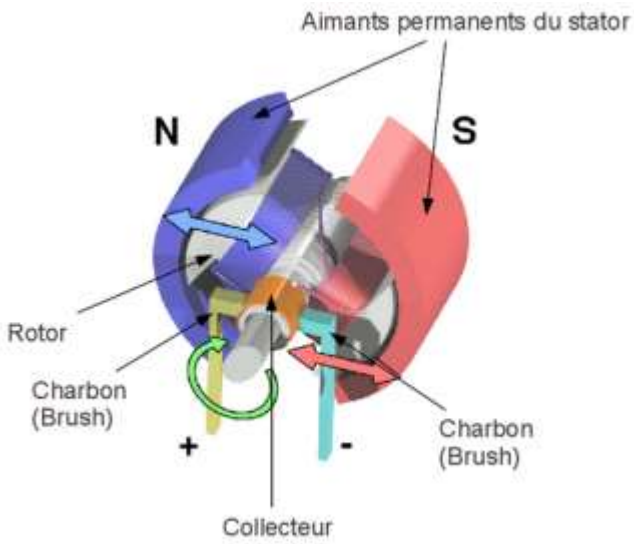
La commande de la vitesse est réalisée en PWM un processeur est utilisé pour la production des signaux et le contrôle de la vitesse réelle par une rétroaction d'une mesure capteur infrarouge  par disque à ligne et led/capteur infrarouge.




**FigureIII.11** Signal PWM.

### III.3.3.2 Moteurs électriques à courant continue :

La plupart des robots mobiles sont ainsi actionnés par des moteurs électriques à courant continu avec ou sans collecteur, alimentés par des convertisseurs de puissance fonctionnant sur batterie.



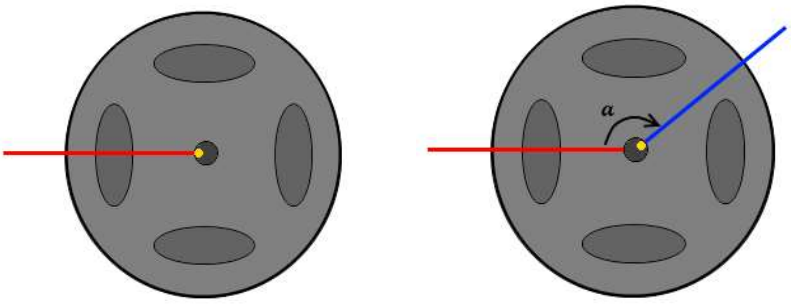

**FigureIII.12** Compostions d'un moteur a courant continue.

Le moteur à courant continu est composé de deux parties principales : le rotor (partie qui tourne) et le stator (partie qui ne tourne pas, statique). En électrotechnique le stator s'appelle aussi inducteur et le rotor s'appelle l'induit**.**

Le rotor composé de fils de cuivre enroulés sur un support lui-même monté sur un axe. Cet axe, c'est l'**arbre** de sortie du moteur. C'est lui qui va transmettre le mouvement à l'ensemble mécanique (pignons, chaîne, actionneur…) qui lui est associé en aval. Dans le cas d'un robot sur roues, on va mettre la roue sur cet axe, bien souvent par l'intermédiaire d'un réducteur qui diminue la vitesse de rotation tout en augmentant le couple.

#### III.3.3.2.1 La vitesse de rotation :

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

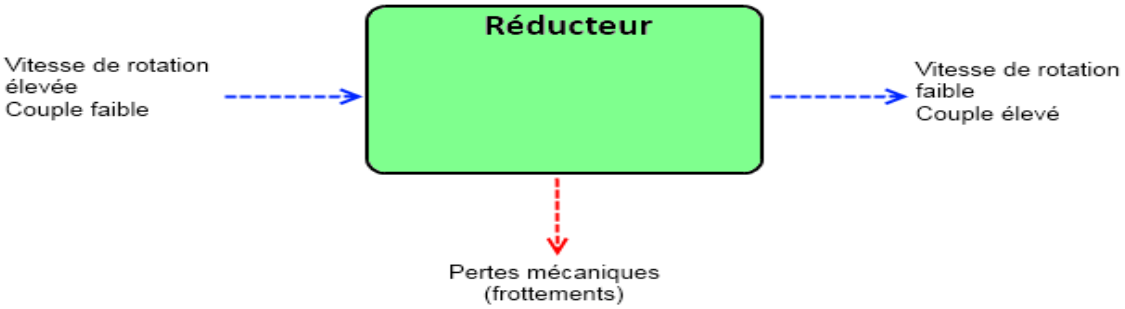



**FigureIII.13** Les roues.

Marquage de l'axe du moteur par un point jaune (gauche). Au bout d'une seconde (droite), mesure de l'angle αentre la position de départ et d'arrivée du point jaune. On obtient alors la vitesse de rotation de l'axe du moteur. Cette mesure est exprimée en angle par seconde.

### III.3.3.2.2 Les réducteurs:

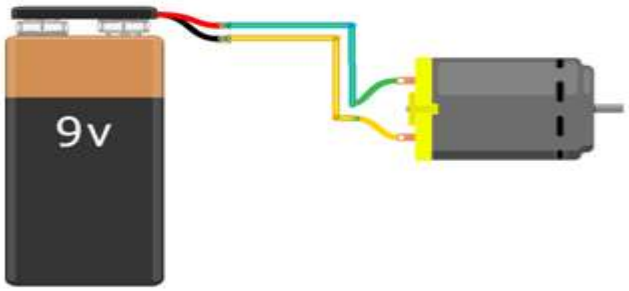
Un moteur électrique est bien souvent très rapide en rotation. Hors si vous avez besoin de faire un robot qui ne va pas trop vite, il va falloir faire en sorte de réduire sa vitesse de rotation. On peut très bien mettre un "frein" qui va empêcher le moteur de tourner vite, ou bien le piloter (on va voir ça toute à l'heure).
Cela dit, même si on réduit sa vitesse de rotation, le moteur ne va pas pouvoir supporter des charges lourdes.




**FigureIII.14** La réduction de la vitesse.

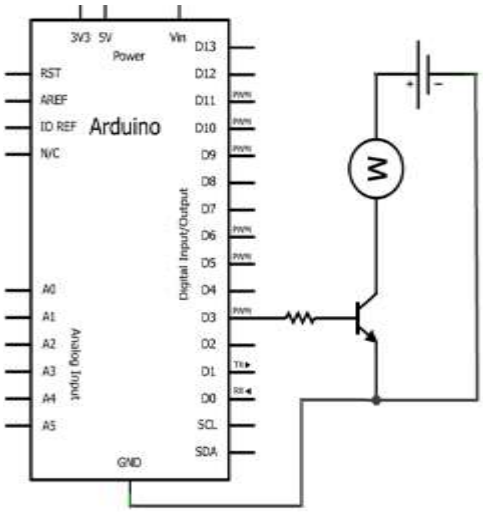
### III.3.3.2.3 Alimentation du moteur :

Connecter un moteur sur une source d'énergie : la pile




**FigureIII.15** Alimentation du moteur avec batterie.

### III.3.3.2.4 Arduino avec le moteur :




**FigureIII.16** Alimentation Arduino avec moteur.

Le transistor est commandé par une sortie de la carte Arduino via la résistance sur la base. Lorsque l'état de la sortie est au niveau 0, le transistor est bloqué et le courant ne le traverse pas. Le moteur ne tourne pas. Lorsque la sortie vaut 1, le transistor est commandé et devient saturé, c'est-à-dire qu'il laisse passer le courant et le moteur se met à tourner. Le problème, c'est que tout n'est pas parfait et ce transistor cumule des inconvénients qu'il est bon de citer pour éviter d'avoir de mauvaises surprises :

- parcouru par un grand courant, il chauffe et peut être amené à griller s'il n'est pas refroidi
- il est en plus sensible aux parasites et risque d'être endommagé
- enfin, il n'aime pas les hautes tensions

Pour répondre à ces trois contraintes, trois solutions. La première consisterait à mettre un transistor qui accepte un courant assez élevé par rapport à la consommation réelle du moteur, ou bien d'adjoindre un dissipateur sur le transistor pour qu'il refroidisse. La deuxième solution concernant les parasites

serait de mettre un condensateur de filtrage. On en a déjà parlé avec les boutons poussoirs. Pour le dernier problème, on va voir que l'on a besoin d'une diode

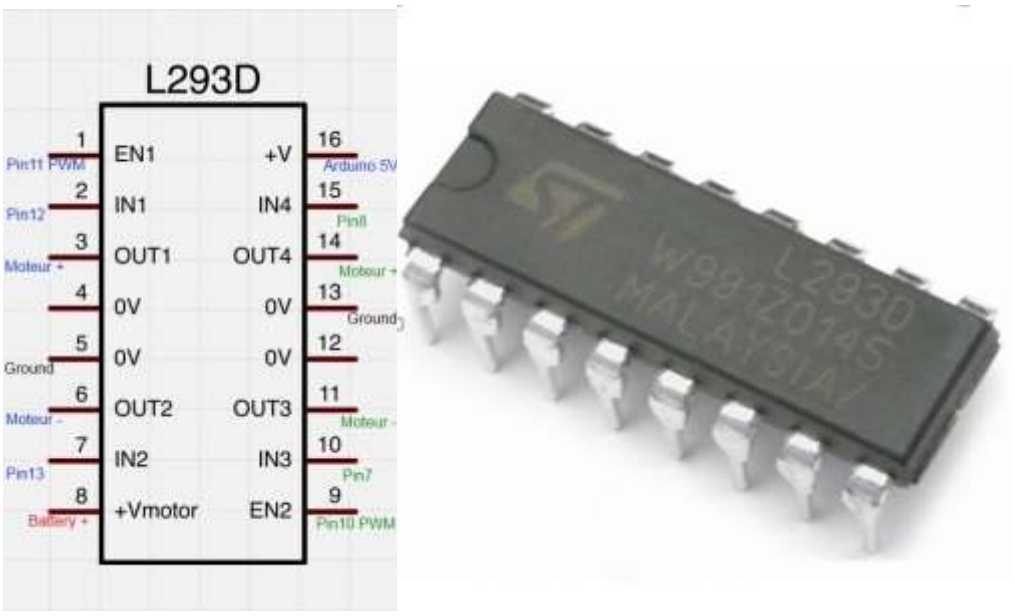
### III.3.4 Alimentation :

La carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5Vpar le port USB mais en utilise une batterie de 6V ou bien par une alimentation externe qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V, associe à un régulateur qui se chargera à réduire la tension à 5V pour le bon fonctionnement de la carte.

### III.3.5 Interface carte de commande de commande :

#### III.3.5.1 Le pont en H :

Nous allons chercher à faire tourner un moteur DC dans les 2 sens grâce à un Arduino. Pour nous aider, nous allons utiliser un composant pas cher et simple à utiliser: le composant L293D.




**FigureIII.17** Pont H L239D.

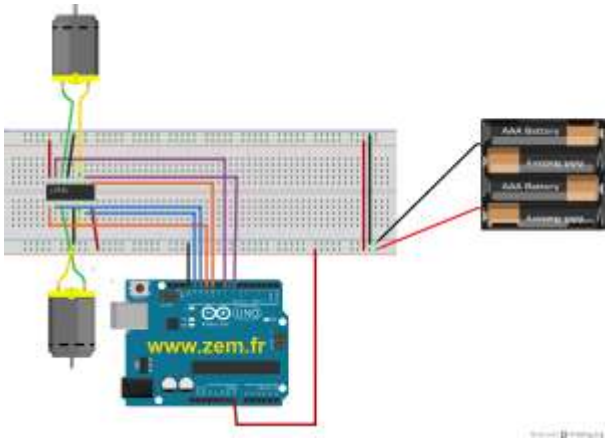
Ce composant est un "quadruple demi pont en 'H', c'est un double pont en H. Ce composant est fait pour fonctionner avec des tensions de 4.5V à 36V et sera capable de délivrer 600 mA par canaux (dans notre cas cela fera 1,2A par moteur puisque nous utiliserons les demi ponts par paire pour tourner dans les deux sens). Un courant de pic peut être toléré allant jusqu'à 1,2A par canaux (donc 2,4A dans notre cas). Enfin, ce composant existe en deux versions, le L293 et le L293D. La seule différence (non négligeable) entre les deux est que le L293D intègre déjà les diodes en parallèle des transistors. Un souci de moins à se

préoccuper ! En revanche, cela implique donc des concessions sur les caractéristiques (le courant max passe à 1A par canaux et 2A pic pour la version sans les diodes).

Pour simplifier le design du circuit, il existe le composant L293Dqui intègre 2 ponts en H dans une seule et même puce. Le circuit permet de contrôler soit 4 moteurs dans un seul sens de rotation, soit deux moteurs dans les deux sens.

### III.3.5.2 Utilisation du L293D avec Arduino et 2 moteurs DC :

Avec un seul pont L293D et un Arduino on va être capable de piloter 2 moteurs à courant continu indépendamment l'un de l'autre. Si la puissance des moteurs est faible nous pourrons même utiliser le 5V en sortie de notre Arduino pour alimenter nos moteurs DC.
Brochage du L293D avec 2 moteurs sous Arduino.

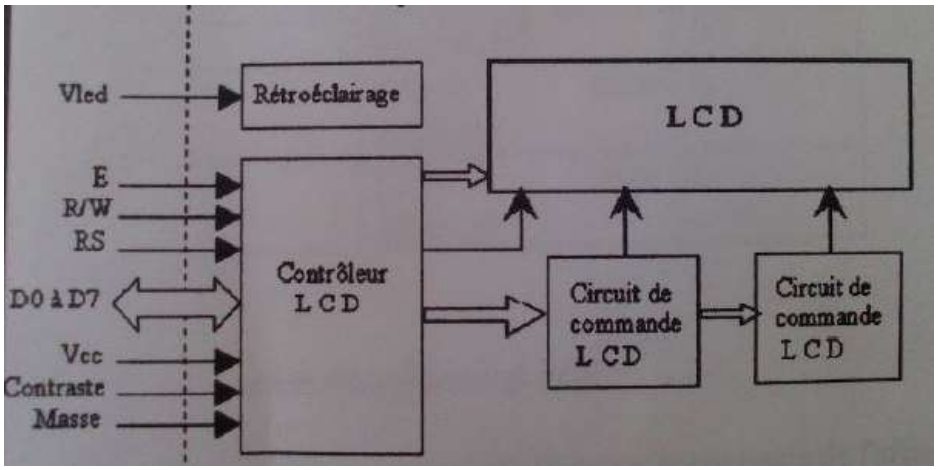



**FigureIII.18 :** Branchement du 2moteurs avec Arduino Uno.

### III.3.6 Affichage :

Les afficheurs à cristaux liquides sont des modules intelligents et qui nécessitent peu de composants externes pour un bon fonctionnement. Plusieurs afficheurs sont disponibles sur le marché et ne diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères) mais aussi par leurs caractéristiques technique et leurs tensions de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant cet éclairage est gourmand en intensité.

**III.3.6.1 Principe de fonctionnement :** La figure suivante représente le schéma fonctionnel d'un afficheur LCD :




**FigureIII.19 :** Schéma fonctionnel d'un LCD.

L'affichage comporte d'autres composants que l'afficheur à cristaux liquides (LCD) seul. Un circuit intégré de commande spécialisé, le LCD-Controller est chargé de la gestion du module. Le « contrôleur » remplit une double fonction : d'une part il commande l'affichage et de l'autre se charge de la communication avec l'extérieur.

### III.3.6.2 Rôle des principales broches des afficheurs LCD :

- V0 **:** potentiel continue permettant d'ajuster le contraste de l'affichage.
- RS **:** l'état logique de cette ligne définit la nature des informations envoyées à l'afficheur.
- Si RS=1 on envoie une donnée.si RS=0 on envoie une commande.
- R/W : permet de lire ou écrire ou sein de la RAM de l'afficheur.
- E **:** validation des informations transmises à l'afficheur.
- D0 à D7 **:** 08 lignes de données bidirectionnelles (selon la position de R/W).

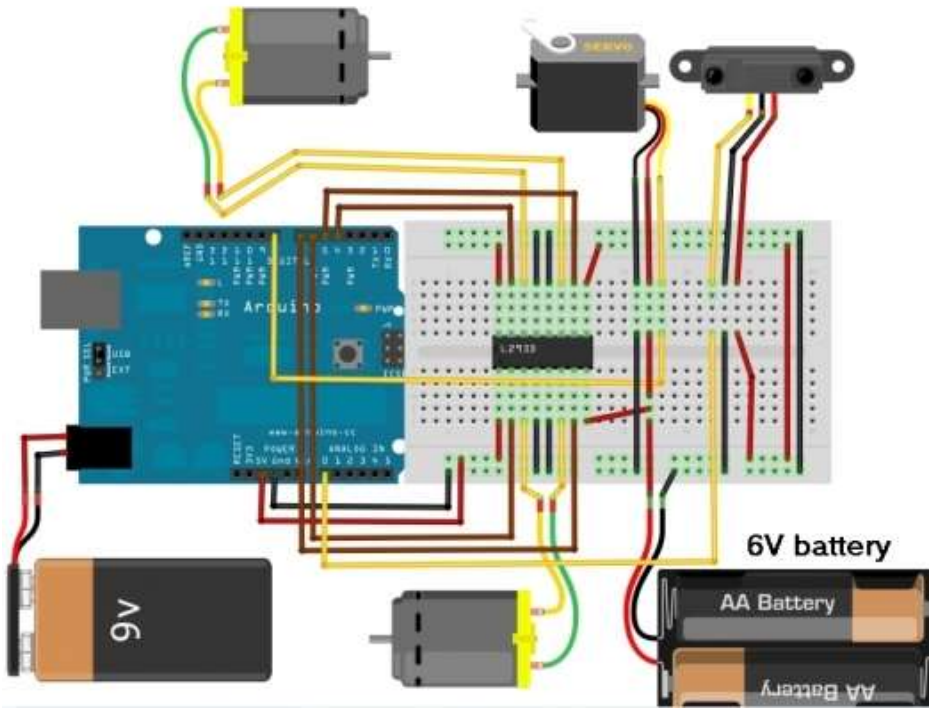
**Remarque :** les afficheurs à logique intégrée peuvent être pilotés en mode 8 bits ou en mode 4 bits par transfert multiplexé sur les bits D4 à D7 du LCD.

### III.4 communication PC-Arduino :

Toutes les données provenant des capteurs seront transmis de la carte Arduino vers le PC via la liaison série RS232. On utilise le code ASCII pour le codage d'information puis on envoie les bits en commençant par le bit de poids faible sachant que le protocole RS232 peut communiquer un mot compris entre 5 et 8 bits. Viennent ensuite se greffer plusieurs autres bits : le ou les bits de stop, l'éventuel bit de parité et enfin le bit de Start.

**III.5 construction:**

Voici notre montage :




**FigureIII.20 :** montage et câblage du robot.

**III.5.1 La répartition de puissance :**

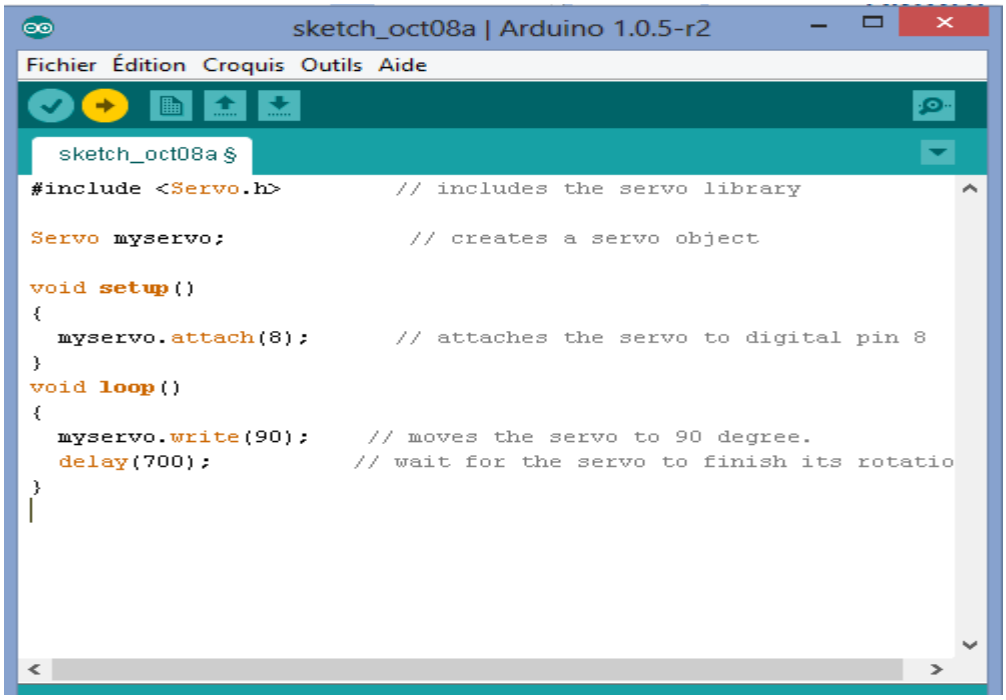
La sonde et les deux broches de validation de pilote de moteur L293D sont alimentées à travers 5V régulée d'alimentation de l'Arduino.

L'Arduino utilise la puissance de 9 V et la régule à 5 V à l'aide d'un régulateur de tension intégré. Les moteurs servo et sont alimentés par 6V.

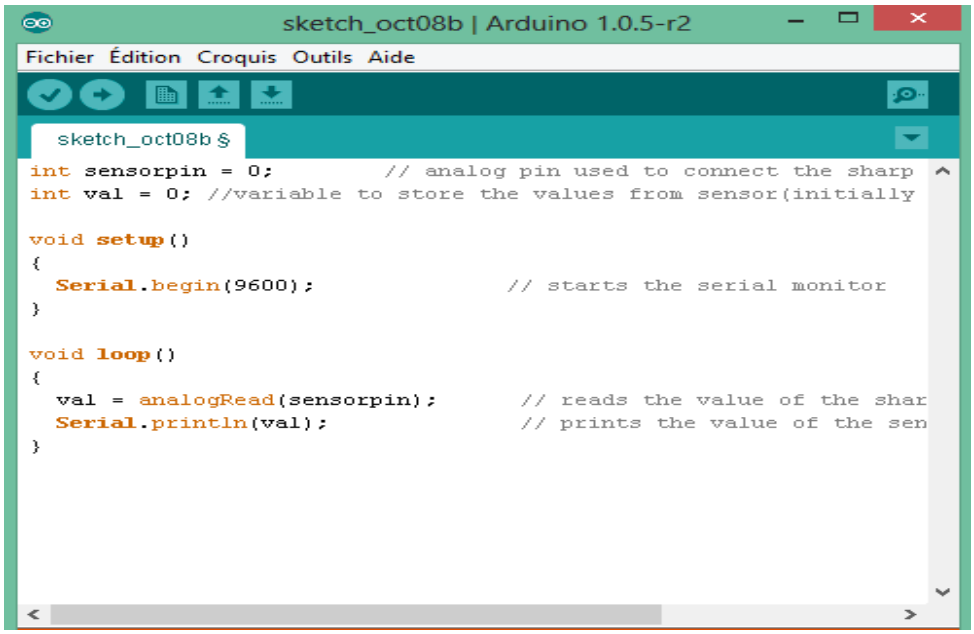
### III.6 programmation :

La première chose que nous devons faire est configurer notre Arduino.

### III.6.1 Code servomoteur :



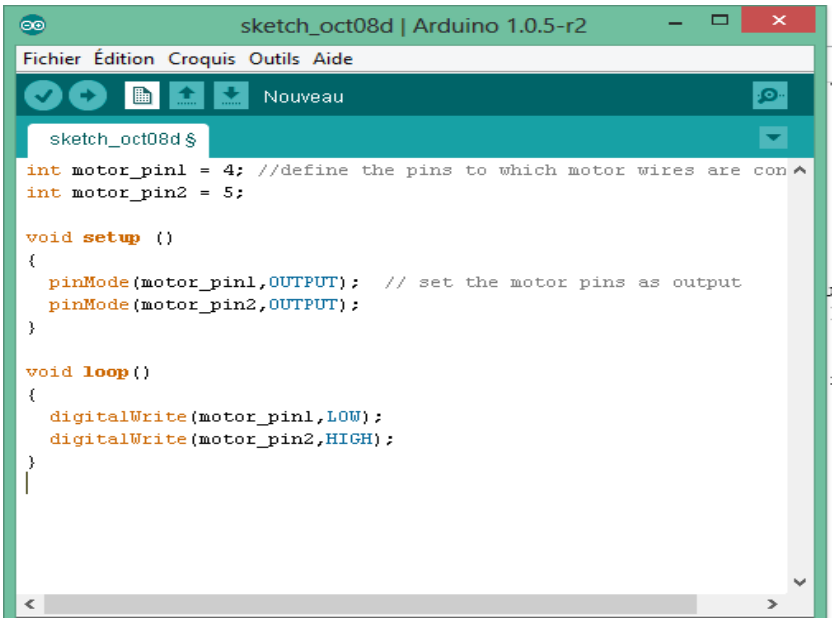

Le code ci-dessus va déplacer le servo à la position centrale soit 90.maintenant on va monter le capteur en face avant sur le palonnier avec le collier ou ruban adhésif double face. Après le capteur est monté, nous pouvons maintenant déplacer avec notre servomoteur en déplaçant vers la gauche ou la droite en utilisant des valeurs entre 0 et 180 au lieu de 90.



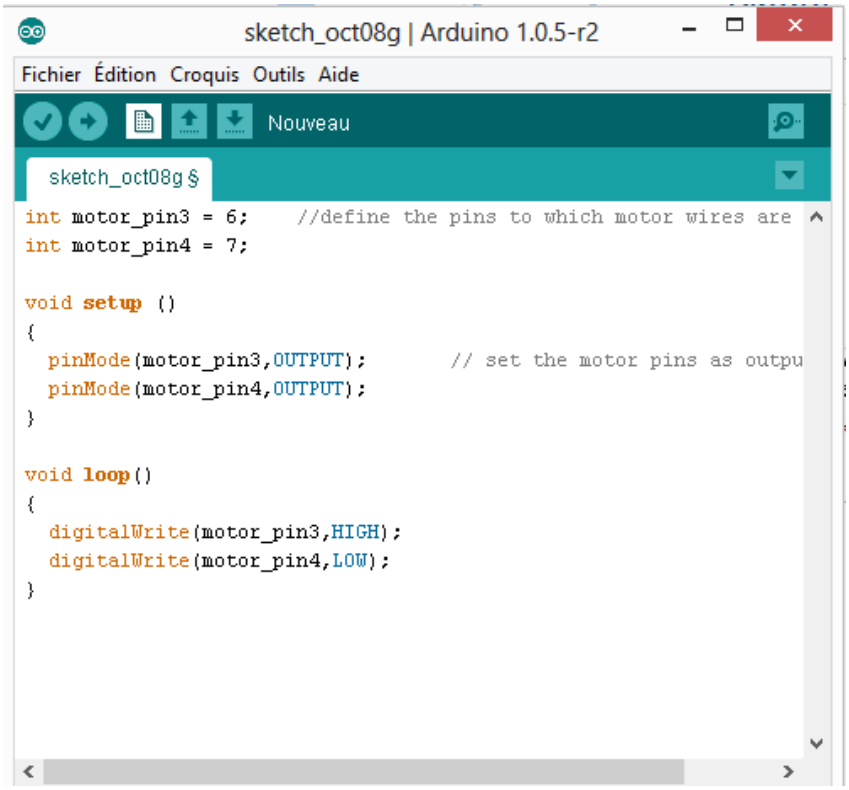

### III.6.2 Code moteurs :


Cela fera un de vos tourner le moteur dans un sens direction. Il est peut-être avant ou en arrière.

S'il va en arrière on utilise ce code et notre moteur tournera vers l'avant :




**Remarque:** dans ce code que nous avons utilisé pour motor_pin1 LOW et HIGH pour motor_pin2 soit opposé au code précédent. D'où le moteur se déplace en sens inverse pour le deuxième moteur.



Cela va tourner le moteur dans une seconde direction. It peut être avant ou en arrière. Si elle se déplace vers l'arrière puis utiliser la même logique que pour le moteur précédent pour déplacer le deuxième moteur avant.

### III.6.3 Programme générale :

**Déclaration et initialisation des broches de la carte Arduino Uno**

```
#include <Servo.h>              //intégrer la bibliothèque du servomoteur
int motor_pin1 = 4;
int motor_pin2 = 5;
int motor_pin3 = 6;
int motor_pin4 = 7;
int servopin = 8;
int sensorpin = 0;
int dist = 0;
int leftdist = 0;
int rightdist = 0;
int object = 500; //distance à laquelle le robot doit chercher une autre
route


Servo myservo;
```

**Fonction d'initialisation de la carte Arduino Uno**

```
void setup ()
{
  pinMode(motor_pin1,OUTPUT);
  pinMode(motor_pin2,OUTPUT);
  pinMode(motor_pin3,OUTPUT);
  pinMode(motor_pin4,OUTPUT);
  myservo.attach(servopin);
  myservo.write(90);
  delay(700);        // le temps d'arrêt est de l'ordre de 700
Millisecondes
}
```

**Initialisation les broches des moteurs (droite et gauche) comme étant des sorties**

**Fonction principale, elle se répète (s'exécute) à l'infini (boucle)**

```
void loop()
{
  dist = analogRead(sensorpin);        //lit le capteur

  if(dist < object) {                 //si la distance est inférieure à 550
  forward();                                //puis aller de l'avant
  }
  if(dist >= object) {   //si la distance est supérieure ou égale à 550
    findroute();
  }
}

void forward() {                  // utiliser une combinaison qui fonctionne
  digitalWrite(motor_pin1,HIGH);
  digitalWrite(motor_pin2,LOW);
  digitalWrite(motor_pin3,HIGH);
  digitalWrite(motor_pin4,LOW);
  return;
```

**Écriture en sortie les broches des pins des moteurs (HIGH=état haut 5V ; LOW=état bas GND) pour que le robot avance**

```
 }

void findroute() {
  halt();                  // Arrêtez
 backward();              // revenir en arrière
  lookleft();              // aller à la sous-routine œil gauche
  lookright();             // aller à la sous-routine regarder à droite

  if ( leftdist < rightdist )
  {
    turnleft();
  }
 else
 {
  turnright ();
 }
}

void backward() {
  digitalWrite(motor_pin1,LOW);
  digitalWrite(motor_pin2,HIGH);
  digitalWrite(motor_pin3,LOW);
  digitalWrite(motor_pin4,HIGH);
  delay(500);      // le temps d'arrêt est de l'ordre 500 milliseconde
  halt();
  return;
}

void halt () {
  digitalWrite(motor_pin1,LOW);
  digitalWrite(motor_pin2,LOW);
  digitalWrite(motor_pin3,LOW);
  digitalWrite(motor_pin4,LOW);
  delay(500);                  // attendre après l'arrêt 500 milliseconde
  return;
}

void lookleft() {
  myservo.write(150);
  delay(700);                        // attendre le servo s'y rendre
  leftdist = analogRead(sensorpin);
  myservo.write(90);
  delay(700);                    // attendre le servo s'y rendre
  return;
}

void lookright () {
  myservo.write(30);
  delay(700);              // attendre le servo s'y rendre
  rightdist = analogRead(sensorpin);
  myservo.write(90);
  delay(700);             // attendre le servo s'y rendre
  return;
}

void turnleft () {
  digitalWrite(motor_pin1,HIGH);  // utiliser la combinaison qui
fonctionne pour nous.
  digitalWrite(motor_pin2,LOW);   // moteur droit tourne vers l'avant et
le moteur gauche vers l'arrière
```

**Écriture en sortie les broches des pins des moteurs (HIGH=état haut 5V ; LOW=état bas GND) pour que le robot recule en arrière**

**Écriture en sortie les broches des pins des moteurs en état bas pour que le robot s'arrête**

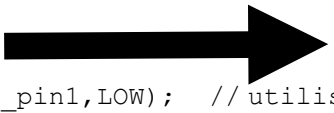
**Perception de l'environnement par le capteur (côté droite)**

**Perception de l'environnement par le capteur (côté gauche)**

**Commande pour tourner à droite**

```
  digitalWrite(motor_pin3,LOW);
  digitalWrite(motor_pin4,HIGH);
  delay(1000);       // attendre que le robot pour faire le tour
  halt();
  return;
}

void turnright () {
  digitalWrite(motor_pin1,LOW);  // utiliser la combinaison qui
fonctionne pour vous
  digitalWrite(motor_pin2,HIGH); // moteur gauche tourne vers l'avant et
le moteur droire tourne vers la droite
  digitalWrite(motor_pin3,HIGH);
  digitalWrite(motor_pin4,LOW);
  delay(1000);         // attendre que le robot pour faire le tour
  halt();
  return;
}
```



**Commande pour tourner à gauche**

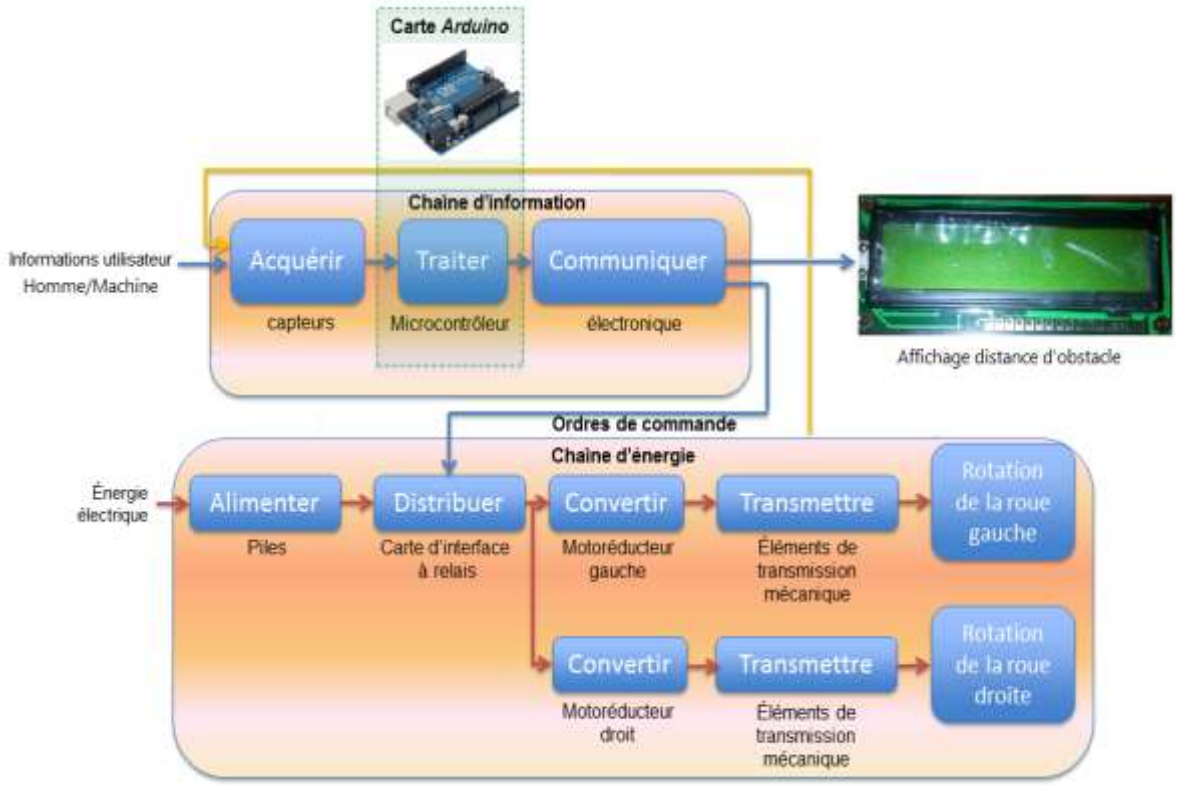
**Fin du programme**

### III.6.3.1 Interprétation :

Le robot se déplace vers l'avant si le capteur a la lecture de moins de 500, il signifie que le robot va considérer qu'il n'a pas d'objet devant lui.
Si la lecture du capteur est inférieure à 500, nous pouvons définir ça en fonction de vos besoins.

Si le capteur se lire supérieur ou égal à 500, alors il va chercher une autre façon dans les étapes suivantes :

- D'abord le robot s'arrête totalement, il déplace un peu vers l'arrière.
- A nouveau le robot s'arrête, le servo se déplace à gauche et le capteur aura une autre lecture et après que le servo se déplacer vers la droite et le capteur sera de nouveau prendre des nouveaux lectures.
- la gauche et la droite sont comparés, Le robot se tourne vers la direction où la lecture est plus grande et il s'arrête.
- A nouveau le même code ou cycle se répète jusqu'à ce que vous éteignez l'alimentation.




**FigureIII.21 :** chaine acquisition.

### III.7 Conclusion :

L'objectif de cette étude était la conception et la réalisation d'un robot mobile autonome à trois roues (deux roues mobiles et une roue fixe). Les résultats expérimentaux obtenus sont satisfaisants sur les trois aspects abordés (mécanique, électronique et informatique).
Nous avons ensuite présenté des outils et méthodes à la disposition du concepteur pour dimensionner correctement ce type de robot, on a assuré l'autonomie.
Comme nous avons pu le constater, ce qui prime dans Arduino, c'est sa simplicité qui permet de mettre en œuvre de nombreux objets numériques à moindre coût sans être un spécialiste du fer à souder ou de la programmation des microcontrôleurs. C'est cette qualité-là qui donne à Arduino le succès planétaire qu'on lui connaît.

# Conclusion générale

L'objectif de notre projet qui consiste à réaliser un robot mobile autonome capable d'éviter les obstacles à l'aide d'un capteur infrarouge de distance permet de contrôler l'environnement.

De plus grâce à l'outil de base de donnée du logiciel réalisé permet de suivre et de contrôler le robot d'une façon permanente. Le logiciel réalisé est souple pratique et convivial vue qu'il offre en premier lieu la possibilité de signaler le changement des données.

Ce modeste travail qui nous avons élaboré nous a permis d'une part d'avoir une idée sur le résultat que peut donner l'association de l'outil informatique et de l'électronique et d'autre part la possibilité offerte à un ordinateur de communiquer avec des périphériques extérieurs.

Ce projet nous a permet de nous familiarisé avec l'utilisation de multiple langage de programmation tel que le C++ ainsi que l'utilisation d'une communication série entre le PC et une carte à base d'Arduino qui intègre un microcontrôleur ATmega328. Cette dernière représente un facteur très important pour l'adapter à d'autres applications, pour cela il suffit de la charger avec un programme adéquat correspond à l'application désirée.

Enfin nous espérons que ce travail qui nous a permet d'élargir nos connaissances servira de base de réflexion pour mieux le développer et l'enrichir dans de prochain projet.

# Perspectives

**Projet future :**

## ‘’Réalisation d’un quadri rotor semi-autonome à base de l’Arduino Uno’’

**Problématique :**

Le domaine des drones est très vaste et dans l’utilisation de ces derniers à basse altitude et dans des situations où on aura une mauvaise visibilité telle que la nuit et le Brouillard pose un risque de crache surtout avec l’existence des obstacles telle que les arbres, les bâtiments, ETC. Ce qui influence sur l’efficacité des drones.

**Composition et fonctionnement :**

Pour construire une quadri rotor qui vous permettra de voler partout (dehors, dedans) et même de faire la mission souhaitable. Son pilotage est relativement simple grâce à sa carte électronique (microcontrôleur et dans notre la carte Arduino Uno) de stabilisation et d’assistance au pilotage. Certain appelle cet engin un Drone ou quadri rotor.

La construction est très simple et peu coûteuse. La structure principale utilise des composants du commerce que l’on peut trouver très facilement dans n’importe qu’elle grande surface de bricolage. Il faut simplement du U d’aluminium (10x10mm), quelques vis/écrou de 3 mm, 2 plaques d’époxy 20/10, un croisillon plastique et des tiges plates de plastiques de 5mm pour sa construire sa structure principale.

Afin d'éviter à l'appareil de tourner sur lui-même sur son axe de lacet, il est nécessaire que deux hélices tournent dans un sens, et les deux autres dans l'autre sens. Pour pouvoir diriger l’appareil.

Les moteur électrique sont plus puissant, la carte Arduino Uno généralement ne fournit pas un courant électrique fore c pour cette raison on utilise les transistors à effet de champ pour résoudre le problème de la puissance, pour faire varier l’altitude ou faire les trois mouvements sur les trois axes on ajuster sur la vitesse de rotation des moteur a courant continue on utilisant le signale PMW.

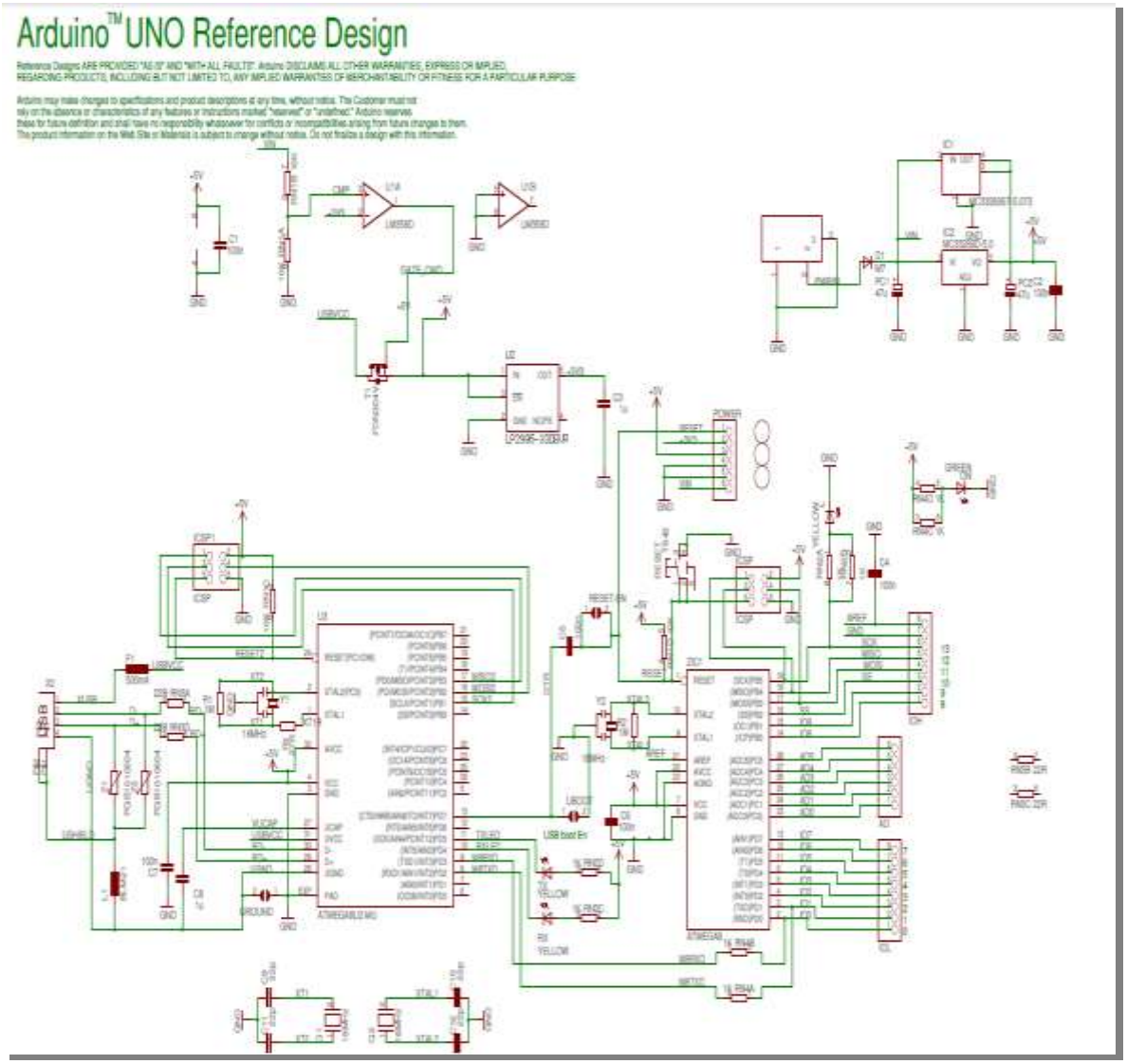
**Résolution de problème :**

Pour éviter les obstacles on utilisant des détecteurs de distance infrarouge puissant qui envoie un signal d’émission et receviez un autre signale réfléchie par l’obstacle. On utilise un servomoteur qui tournera pour la perception de l’environnement. Suite a un programme installé sur notre carte Arduino connecté

à un pc embarqué nous pouvons contrôler les mouvements (latérale et longitudinal) en commandent les moteurs a courant continue.

Donc on aura un quadri-rotor semi-autonome qui a toujours une séparation latérale et longitudinale ce qui permet de protéger notre drone.

# Annexe **A**



# Bibliographique :


[1] Mémoire de Magister: Commande d'un bras de Robot Manipulateur à l'aide d'une carte dSPACE - ds1104.

[2] R. P. Paul, "Robot manipulators: mathematics, programming, and control. The Computer Control of Robot Manipulators", The MIT Press, Cambridge, 1983.

[3] Le site Arduino : http://www.arduino.cc.

[4]Le lien: http://www.louisreynier.com.

[5]Le lien: www.openclassrooms.com

[6]Le lien: http://www.semageek.com/category/electronique/arduino-electronique/

[7] Mémoire Automatique/Présenté par: MECHALIKH Youcef MILOUDDI Ali/Thème: Développement d'algorithmes d'évitement d'obstacles statiques et dynamiques/2011/2012

[8] Ulrich, Borenstein. VFH * : L'action d'éviter locale d'obstacle avec regardent-en avant vérification. Dans IEEE international. Conf. Sur la robotique et l'automation San Francisco, les Etats-Unis, 2000

[9] C. Latombe. Planification de mouvement. Université de Stanford, Etats-Unis, 2004. HTTP : WWW. latombe/cs326/2004/index.htm de //robotics.stanford.edu/.

[10] PROJET. Laumond, éditeur. Mobile de robotique de La. Les sciences du `es de Herm, 2001.

[11] Projet de Fin D'étude : SLAM Visuel 3D pour robot mobile autonome
Romain Drouilly

[12].coure: Robots mobiles autonomes par Alain PRUSKI Professeur à l'Université de Metz.

[13] le site : www. Cite de zéro /ARDUINO.fr.

[14] Projet de Fin D'étude: David Gingras/Université de Sherbrooke/Agence spatiale canadienne/Le 3 décembre 2009.

[15] les liens : http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=main. Débuter réinstallation Windows.
http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=main. Débuter présentation logiciel.

[16] le lien: http://skyduino.wordpress.com/2011/08/06/test-rainbowduino.

[17] le site web: http://skyduino.wordpress.com/2011/08/31/test-olimex-stm32-maple-arduino/

[18] Thèse de Doctorat / l'Institut National Polytechnique de Toulouse /Navigation autonome sans collision pour robots mobiles non holonomes/2006.

[19] le lien: www.Electronique et robotique/ Robots Programmables /Tutoriels en robotique.com  le 31 jan 2014 par Jérôme Laplace.