

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Projet de Fin d'Études

présenté par

MADDI Dalila

&

OUDAI Laalia

pour l'obtention du diplôme de Master option Systèmes de Vision et Robotique

Thème

# Conception d'un robot mobile muni d'une caméra embarquée pour la poursuite d'objets de couleur prédéfinie

Proposé par : KAZED Boualem

Année Universitaire 2011-2012

*En témoignage de nos profonds sentiments de respect et de reconnaissance, nous tenons à remercier Monsieur KAZED boualem , notre promoteur, pour avoir accepté de nous encadrer sur un sujet intéressant et d'actualité, pour ses conseils précieux, ses recommandations, ses remarques constructives et surtout pour sa disponibilité.*

*Nos remerciements vont également aux membres du jury pour l'honneur qu'ils nous font en acceptant d'évaluer ce travail.*

*Nous tenons à remercier nos enseignants qui ont su nous apprécier l'électronique, qui ont donné de leurs mieux pour notre formation.*

*Nous remercions, tous ceux et celles qui ont contribué de près ou de loin à la finalisation de ce travail.*

---

**Résumé :** Dans ce projet nous allons concevoir et réaliser un robot mobile capable de reconnaître et poursuivre un objet de couleur particulière. Pour ce faire nous allons exploiter une caméra de type CMUCam qui sera embarquée sur notre robot et permettra au module de commande de gérer les déplacements du robot en fonction de la position de l'objet recherché. Le traitement bas niveau des images captées par la caméra s'effectue au sein même de la CMUCam grâce à un microcontrôleur SX28 cadencé à 75 MHz. La partie commande de notre robot sera basée sur le module ARDUINO Méga 2560, ce dernier sera en charge d'interroger et d'interpréter les données issues de la CMUCam.

**Mots clés :** Robot Mobile ; CMUCam ; Arduino.

---

**Abstract :** This project aims at designing a vision based navigation robot which will be able to track an object according to its particular color. This will be achieved using the CMUCam module placed on board of our robot. The low level image processing routines will be executed within the CMUCam module which is based on the SX28 microcontroller paced at 75 MHz. The overall operations of the robot will be carried out on an ARDUINO Mega 2560 module which subsequently takes care of the data exchange with the CMUCam.

**Keywords :** Mobile Robot ; CMUCam ; Arduino

---

## **Listes des acronymes et abréviations**

CMUCam : Carnegie Mellon University Camera

PWM : Pulse Width Modulation

# Table des matières

Introduction générale.....	1
Chapitre 1 Généralité .....	3
1.1 Généralité .....	3
1.2 Un bref aperçu historique .....	4
1.3 Exemples d'applications .....	6
Chapitre 2 mécaniques du robot .....	8
2.1 Introduction.....	8
2.2 Réalisation mécanique du châssis .....	8
2.2.1 Les roue .....	9
2.2.2 Les moteurs à courant continu .....	9
2.3 Description du robot .....	11
2.3.1 Compartiment de commande .....	11
2.3.2 Compartiment de puissance .....	12
2.3.3 Le couvercle.....	12
Conclusion.....	13
Chapitre 3 électronique du robot .....	14
3.1 La carte électronique.....	14
3.1.1 Introduction .....	14
3.1.2 Principe de fonctionnement de la commande du robot .....	14
<i>a. élément du robot.....</i>	<i>14</i>
<i>b. description du robot .....</i>	<i>15</i>
<i>c. le pont h .....</i>	<i>15</i>

<i>d. exemple du circuit L 298.....</i>	<i>17</i>
<i>e. schéma électronique.....</i>	<i>17</i>
<i>f. modulation de la largeur d'impulsion .....</i>	<i>17</i>
<b>3.2 Les capteur du robot .....</b>	<b>18</b>
<b>3.2.1 Capteur de couleur.....</b>	<b>19</b>
<i>a. la CMUcam .....</i>	<i>19</i>
<i>b. caractéristique .....</i>	<i>19</i>
<i>c. fonctionnalité.....</i>	<i>20</i>
<i>d. multi seuillage couleur et le suivi d'obje .....</i>	<i>20</i>
<b>3.2.2 Détecteur d'obstacle .....</b>	<b>21</b>
<i>a. Capteur infrarouge SHARP GP2D15.....</i>	<i>21</i>
<i>b. présentation .....</i>	<i>21</i>
<i>c. principe de fonctionnement.....</i>	<i>22</i>
<i>d. Schéma de branchement .....</i>	<i>22</i>
<i>e. Montage avec une carte Arduino.....</i>	<i>22</i>
<b>3.3. La carte Arduino .....</b>	<b>23</b>
<b>3.3.1 La carte arduino méga 2560.....</b>	<b>23</b>
<b>3.3.2 Présentation.....</b>	<b>23</b>
<b>3.3.3 Synthèse des caractéristiques.....</b>	<b>24</b>
<b>3.3.4 Alimentation .....</b>	<b>25</b>
<b>3.3.5 espace de développement intégré (EDI) Arduino.....</b>	<b>25</b>
<b>3.4 Servomoteur .....</b>	<b>28</b>

3.5 Les encodeurs optiques .....	29
Chapitre 4 programmations du robot.....	31
4.1 Introduction .....	31
4.2.1 La première approche .....	31
4.2.2 La deuxième approche.....	35
4.3 Les algorithmes.....	36
4.3.1. Algorithme de la première approche.....	36
4.3.2 La deuxième approche .....	37
<i>a. algorithme avance arrière.....</i>	<i>37</i>
<i>b. algorithme droite a gauche.....</i>	<i>38</i>
<i>c. algorithme servomoteur .....</i>	<i>38</i>
4.4 Modélisation <i>des</i> <i>moteurs</i> .....	<i>40</i>
4.5 Conclusion.....	45
Conclusion générale.....	46

## Liste des figures :

Figure 1. Schéma des interactions d'un robot avec son environnement.....	3
Figure 2. La tortue de Grey Walter.....	4
Figure 3. Robot Beast (1960), robot Shakey (1970) .....	5
Figure 4. Robot LAAS .....	5
Figure 5. Genghis, développé par Rodney Brooks .....	6
Figure 6. Exemples de robots commerciaux ou de recherche.....	7
Figure 7. Roue du robot.....	9
Figure 8. Source d'énergie.....	10
Figure 9. Schéma de machine à courant continu vue en coupe latérale.....	10
Figure 10. Châssis du robot .....	11
Figure 11. Support de la caméra .....	12
Figure 12. Bouton d'arrêt d'urgence .....	12
Figure 13. Photo du robot.....	13
Figure 14. La carte électronique .....	14
Figure 15. Pont h(L298) .....	15
Figure 16. Schéma illustrant le principe de fonctionnement du pont H.....	16
Figure 17. Montage en pont .....	16
Figure 18. Schéma interne du l298.....	17
Figure 19. Schéma électronique du l298 .....	17
Figure 20. Modulation de la largeur d'impulsion .....	18
Figure 21. . La CMUCam .....	19
Figure 22. Télémètre infrarouge SHARP GP2D15 .....	21
Figure 23. Schéma de branchement .....	22
Figure 24. Schéma du branchement du capteur avec la carte.....	23
Figure 25. Cartes Arduino 2560 .....	23
Figure 26. Espace de développement de la carte Arduino .....	26

Figure 27. La barre des boutons .....	27
Figure 28. Terminal série .....	27
Figure 29. Servomoteur .....	28
Figure 30.principe de fonctionnement des encodeurs.....	29
Figure 31. Schéma électrique d'encodeur optique .....	29
Figure 32.trajectoire du robot sur la table .....	32
Figure 33 schéma d'un entrainement avec une mcc a excitation indépendante.....	40
Figure 34.schéma du courant .....	43
Figure 35.schéma de la vitesse .....	44

# Introduction générale

---

La capacité d'autonomie d'un robot mobile repose sur ses aptitudes à interpréter et s'adapter à l'environnement dans lequel il évolue. Dans le cadre de ce projet, notre but sera de réaliser un robot mobile qui répond à un cahier de charges limité, ceci étant donné la durée ainsi que les moyens dont nous disposons. Afin de rester dans le contexte de la spécialité du master pour lequel nous allons soumettre ce travail, nous avons choisi de consacrer nos efforts à étudier et exploiter le module de vision CMUCam, qui sera monté à bord de notre robot. Le but principal auquel nous voulons aboutir étant de donner à notre robot la capacité à reconnaître et suivre un objet de couleur prédéfinie, ce suivi devra s'effectuer aussi bien sur l'axe horizontal que vertical. Le choix de la couleur se fera au niveau de la carte de commande principale sous forme de 3 valeurs numériques représentant les proportions des 3 couleurs fondamentales (Rouge, Vert, Bleu).

La gestion globale de notre robot sera assurée par un programme qui sera chargé sur un module de type ARDUINO Méga 2560, cette carte comportera entre autres les circuits de puissance sous forme de ponts en H ainsi que des connecteurs adaptés pour relier les différents capteurs et actionneurs constituant ce robot. Ce programme se présente sous forme de plusieurs routines chacune dédiée à une tâche bien précise, telles que l'envoi et la réception des données issues de la CMUCam, ou encore la commande des deux moteurs de déplacement à travers des ponts en H.

Pour ce qui concerne la partie mécanique, bien que celle-ci ne présente aucune contrainte dans le cadre de ce travail, cette dernière a été réalisée en

fonction d'une stratégie assez particulière que nous avons adopté lors de notre participation au concours national de robotique « Eurobot Algeria » en Avril 2012.

Ce mémoire est organisé en quatre chapitres ; dans le premier nous donnerons quelques généralités sur la robotique, le second chapitre sera consacré à la description de la partie mécanique du robot, dans le troisième chapitre nous aborderons les parties électroniques, le dernier chapitre sera réservé à la partie intelligence du robot c'est-à-dire le coté programmation de la stratégie et la gestion de tous les capteurs et actionneurs présents sur le robot. Nous terminerons par une conclusion générale de ce travail.

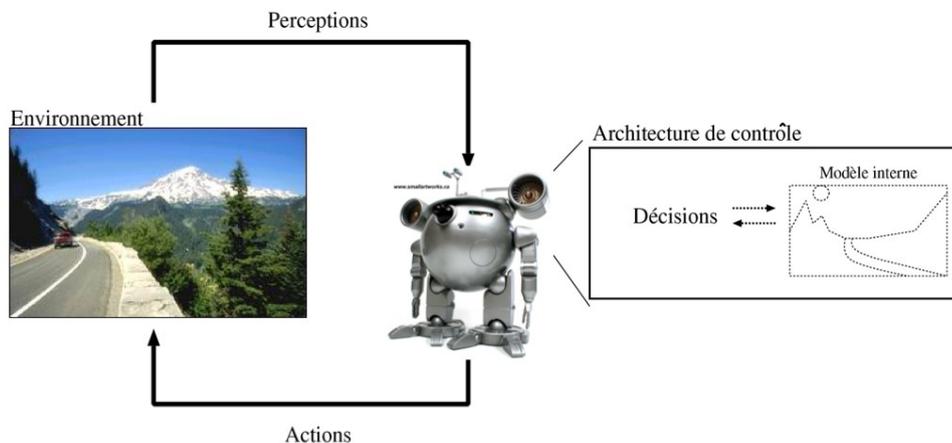
# Chapitre 1 Généralité sur la robotique

---

## 1.1 Généralité :

La robotique est un très bon exemple de domaine pluridisciplinaire qui implique de nombreuses thématiques telles que la mécanique, l'électronique, l'automatique, l'informatique ou l'intelligence artificielle. En fonction du domaine d'origine des auteurs, il existe donc diverses définitions du terme robot, mais elles tournent en général autour de celle-ci :

Un robot est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a.



**Figure 1.** Schéma des interactions d'un robot avec son environnement.

Cette définition s'illustre par un schéma classique des interactions d'un robot avec son environnement

(Figure .1). Les différentes notions que nous présenterons sont essentiellement issues de cette vision de la robotique, très orientée sur l'Intelligence Artificielle, qui place au centre des préoccupations l'enchaînement de ce cycle Perception/Décision/Action. La manière dont un robot gère ces différents éléments

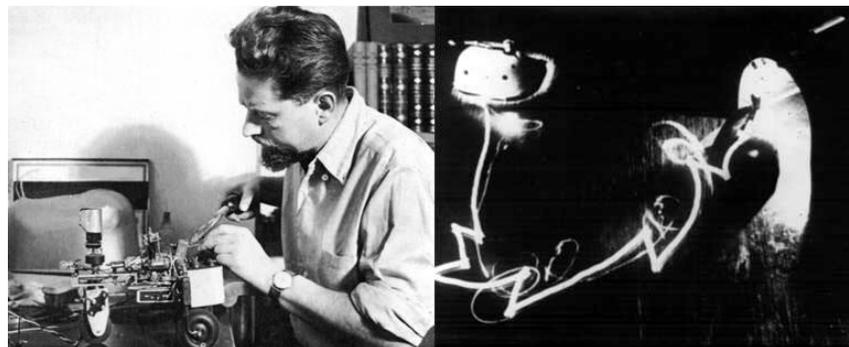
est définie par son architecture de contrôle, qui la plupart du temps va faire appel à un modèle interne de l'environnement qui lui permettra de planifier ses actions à long terme.

## 1.2 Un bref aperçu historique :

Le terme de robot apparaît pour la première fois dans une pièce de Karel Capek en 1920 Rossum's Universal Robots. Il vient du tchèque 'robota' et présente une vision des robots comme serviteurs dociles et efficaces pour réaliser les tâches pénibles mais qui déjà vont se rebeller contre leurs créateurs.

La Tortue construite par Grey Walter dans les années 1950 (Figure.2), est l'un des premiers robots mobiles autonomes. Grey Walter n'utilise que quelques composants analogiques, dont des tubes à vide, mais son robot est capable de se diriger vers une lumière qui marque un but, de s'arrêter face à des obstacles et de recharger ses batteries lorsqu'il arrive dans sa niche.

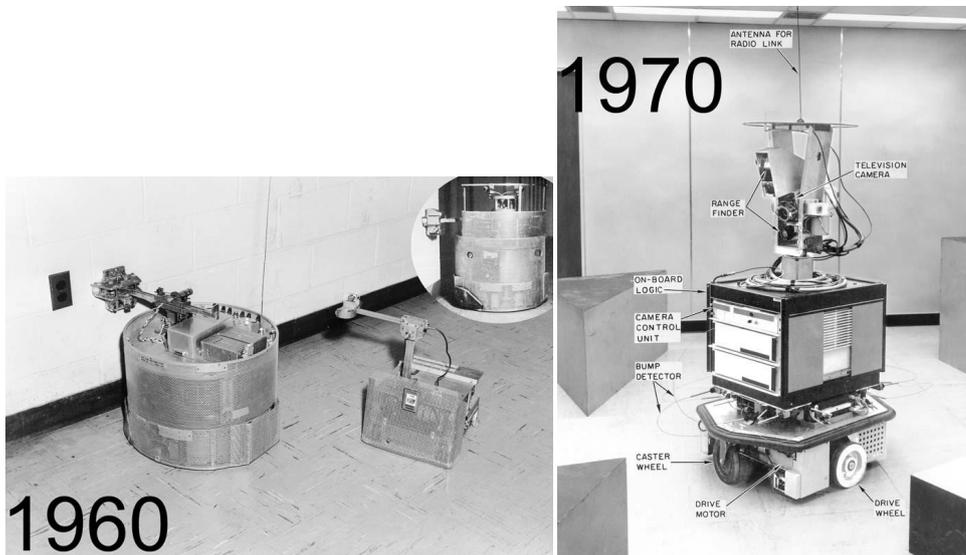
Toutes ces fonctions sont réalisées dans un environnement entièrement préparé, mais restent des fonctions de base qui sont toujours des sujets de recherche et de développement technologiques pour les rendre de plus en plus génériques et robustes.



*Figure 2.* La tortue de Grey Walter

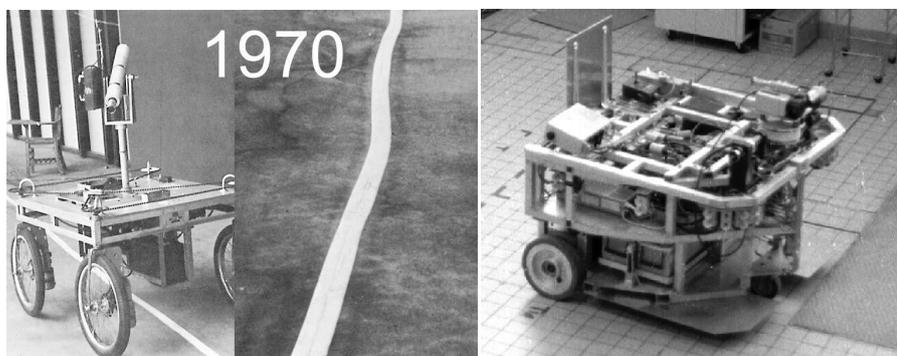
Dans les années 60, les recherches en électronique vont conduire, avec l'apparition du transistor, à des robots plus complexes mais qui vont réaliser des tâches similaires. Ainsi le robot "Beast" (Figure.3) de l'université John Hopkins est capable de se déplacer au centre des couloirs en utilisant des capteurs ultrason, de chercher des prises électriques (noires sur des murs blancs) en utilisant des photodiodes et de s'y recharger.

Les premiers liens entre la recherche en intelligence artificielle et la robotique apparaissent à Stanford en 1969 avec Shakey (Figure 3). Ce robot utilise des télémètres à ultrason et une caméra et sert de plate-forme pour la recherche en intelligence artificielle, qui à l'époque travaille essentiellement sur des approches symboliques de la planification. La perception de l'environnement, qui à l'époque est considérée comme un problème séparé, voire secondaire, se révèle particulièrement complexe et conduit là aussi à de fortes contraintes sur l'environnement. Ces développements se poursuivent avec le Stanford Cart à la fin des années 1970, avec notamment les premières utilisations de la stéréovision pour la détection d'obstacles et la modélisation de l'environnement.



**Figure 3.** Robot Beast (1960), robot Shakey (1970)

En France, le robot Hilare est le premier robot construit au LAAS, à Toulouse (Figure 4).



**Figure 4.** Robot LAAS

Le Stanford Cart date de la fin des années 1970. Le robot Hilare du LAAS a été construit en 1977.

Une étape importante est à signaler au début des années 1990 avec la mise en avant de la robotique réactive, représentée notamment par Rodney Brooks. Cette nouvelle approche de la robotique, qui met la perception au centre de la problématique, a permis de passer de gros robots très lents à de petits robots (Figure 5), beaucoup plus réactifs et adaptés à leur environnement.

Ces robots n'utilisent pas ou peu de modélisation du monde, problématique qui s'est avérée être extrêmement complexe



*Figure 5.* Genghis, développé par Rodney Brooks

Ces développements ont continué et l'arrivée sur le marché depuis les années 1990 de plates formes intégrées telles que le pioneer de la société Mobile Robots a permis à de très nombreux laboratoires de travailler sur la robotique mobile et à conduit à une explosion de la diversité des thèmes de recherche. Ainsi, même si les problèmes de déplacement dans l'espace et de modélisation de l'environnement restent difficiles et cruciaux, des laboratoires ont pu par exemple travailler sur des approches multi-robot, la problématique de l'apprentissage ou sur les problèmes d'interactions entre les hommes et les robots.

### **1.3 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", mais également pour des applications ludiques ou de service, comme l'assistance aux personnes âgées ou handicapées.

Parmi les domaines d'applications possibles de la robotique, nous 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).

A cela, s'ajoute à l'heure actuelle des nombreuses plates-formes conçues essentiellement pour les laboratoires de recherche. (Figure 6) montre quelques exemples de robots existants [1].



**Figure 6.** Exemples de robots commerciaux ou de recherche.

# Chapitre 2 mécanique du robot

---

## 2.1 Introduction :

Notre robot mobile a été réalisé dans le cadre de la compétition européenne de robotique Eurobot 2012. Cette coupe constitue un défi ludique et technique proposé aux participants avec un règlement différent chaque année. L'étude minutieuse du règlement permet de connaître les caractéristiques limitées autorisées ainsi de choisir un mécanisme qui permet de concevoir le robot le plus performant possible tout en restant conforme au règlement.

La réalisation du robot nous a amené à une conception mécanique relativement simple mais efficace permettant d'atteindre le but recherché

## 2.2 Réalisation mécanique du châssis :

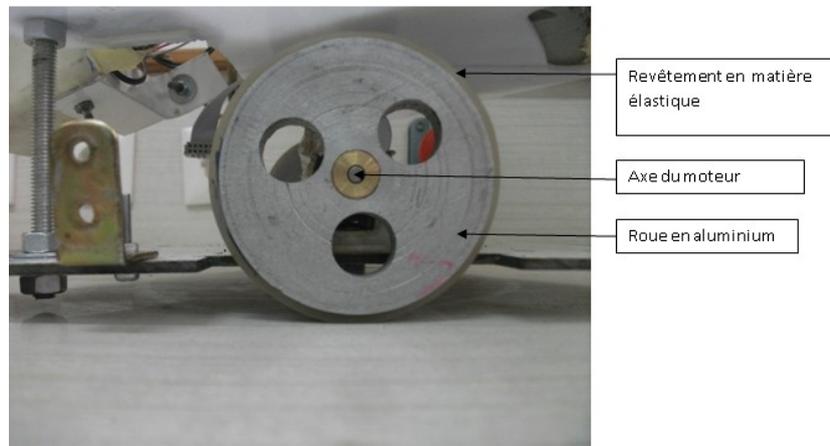
La construction d'un robot débute par le choix de la forme. Les robots mobiles à deux roues peuvent prendre plusieurs formes, nous citons comme exemple la forme circulaire, carré....etc.

Nous avons opté pour une forme triangulaire car comme nous l'avons mentionné dans l'introduction, nous avons participé au concours de eurobot 2012, et nous avons trouvé que cette forme nous permet le ramassage de plusieurs objets au même temps.

Nous avons commencé la construction du châssis par le découpage d'une plaque de résine en un triangle de 32 cm de chaque côté, puis nous avons fixé les moteurs sur cette dernière à l'aide des attaches tuyaux.

### 2.2.1 Les roues:

Les roues sont fixées à l'axe des moteurs, elles ont un diamètre de 8 cm (7,5 cm la roue en aluminium + 0,5 cm du revêtement en matière élastique figure 7).



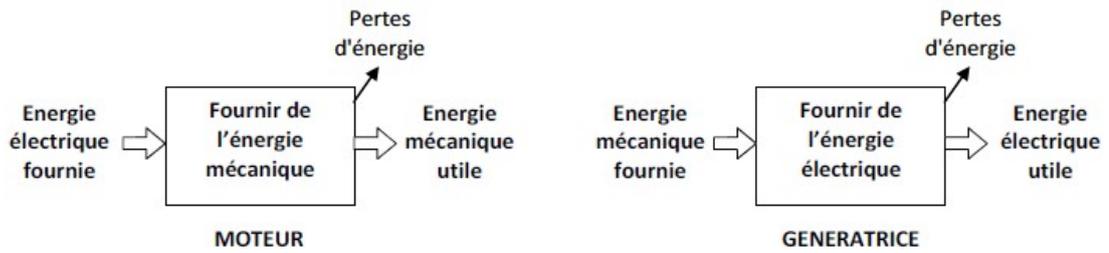
*Figure 7.* Roue du robot

Pour avoir un équilibre entre l'avant et l'arrière on a ajouté deux roues à billes comme le montre la figure 10

### 2.2.2 Les moteurs à courant continu :

Ce sont des machines électriques il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique; selon la source d'énergie

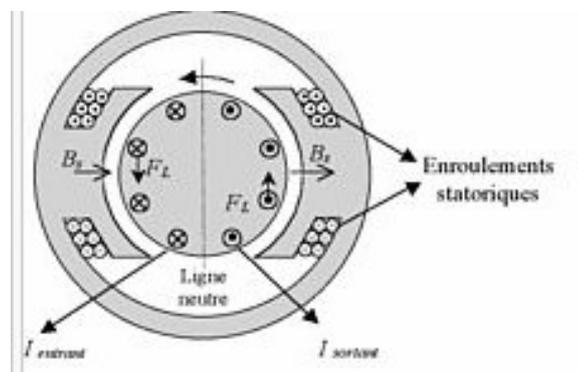
- ✓ En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- ✓ En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique



**Figure 8.** Source d'énergie

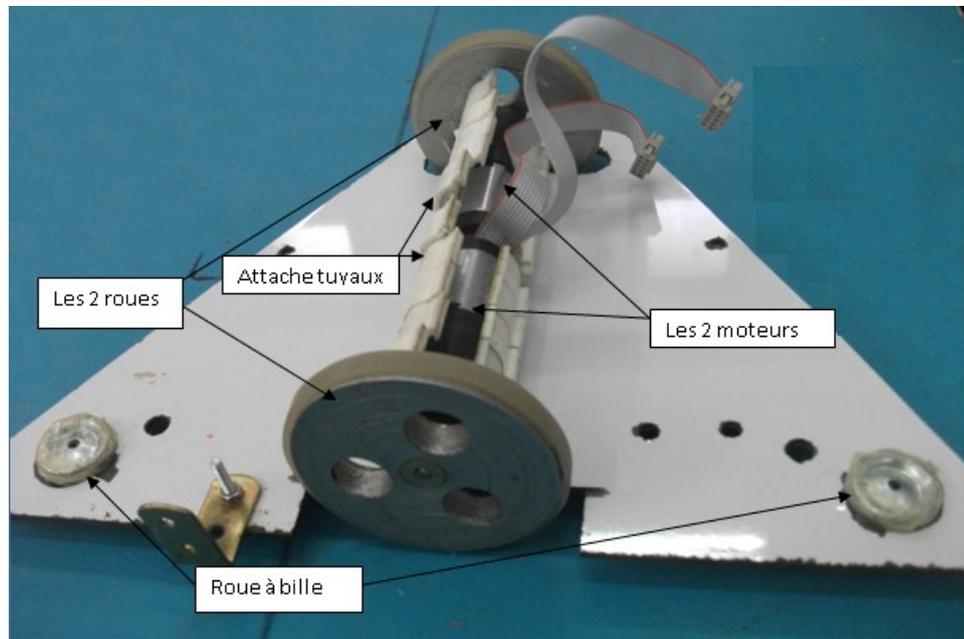
Dans ce cas elle est aussi appelée dynamo. Ils sont constitués d' :

- ✓ Un stator qui crée un flux magnétique fixé soit par des enroulements statoriques (bobinage) soit par des aimants permanents. Il est aussi appelé « inducteur » en référence au fonctionnement en génératrice de cette machine.
- ✓ Un rotor bobiné relié à un collecteur rotatif inversant la polarité de chaque enroulement rotorique, au moins une fois par tour, de façon à faire circuler un flux magnétique transversal en quadrature avec le flux statorique. Les enroulements rotoriques sont aussi appelés enroulements d'induits, ou communément « induit » en référence au fonctionnement en génératrice de cette machine



**Figure 9.** Schéma de machine à courant continu vue en coupe latérale.

La partie ronde au centre est le rotor, qui tourne dans le sens anti-horaire. L'autre partie constitue le stator.



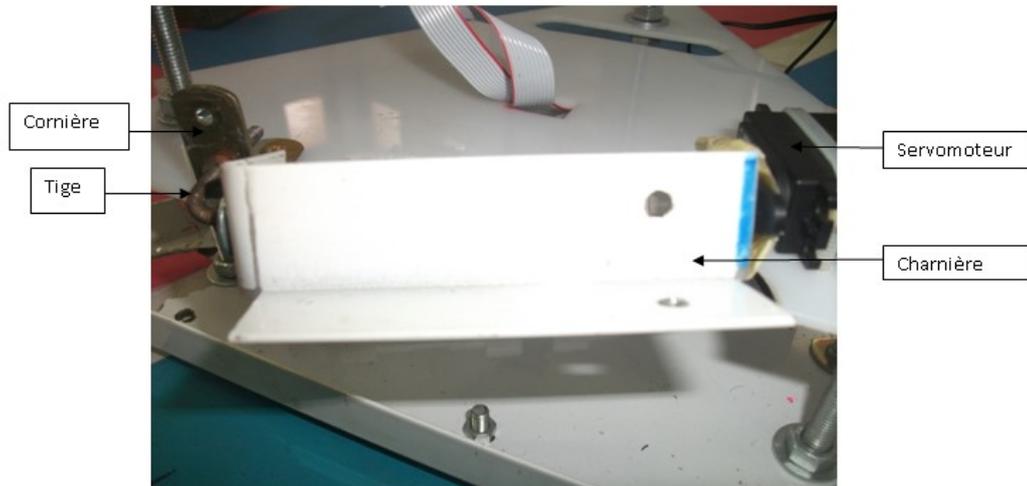
*Figure 10.* Châssis du robot

## 2.3 Description du robot :

Or le châssis, le robot est construit d'un compartiment de commande, d'un compartiment d'alimentation et d'un couvercle.

### 2.3.1 Compartiment de commande :

C'est un triangle en plexiglas de 32cm de chaque côté, sur lequel, on a prévu une ouverture pour permettre le mouvement de la caméra, cette dernière est fixée sur une charnière dont on a découpé et retourné de 90° les deux extrémités (figure 11).



**Figure 11.** Support de la caméra

On a fixé l'extrémité à la tige, et l'autre extrémité au servomoteur ce la afin de faire pivoter la caméra vers le haut et vers le bas.

La carte de puissance est aussi mise dans cet étage.

### **2.3.2 Compartiment d'alimentation :**

Triangle en résine de 32cm de chaque coté, sur le quel on a mi une batterie de 12V /7A.

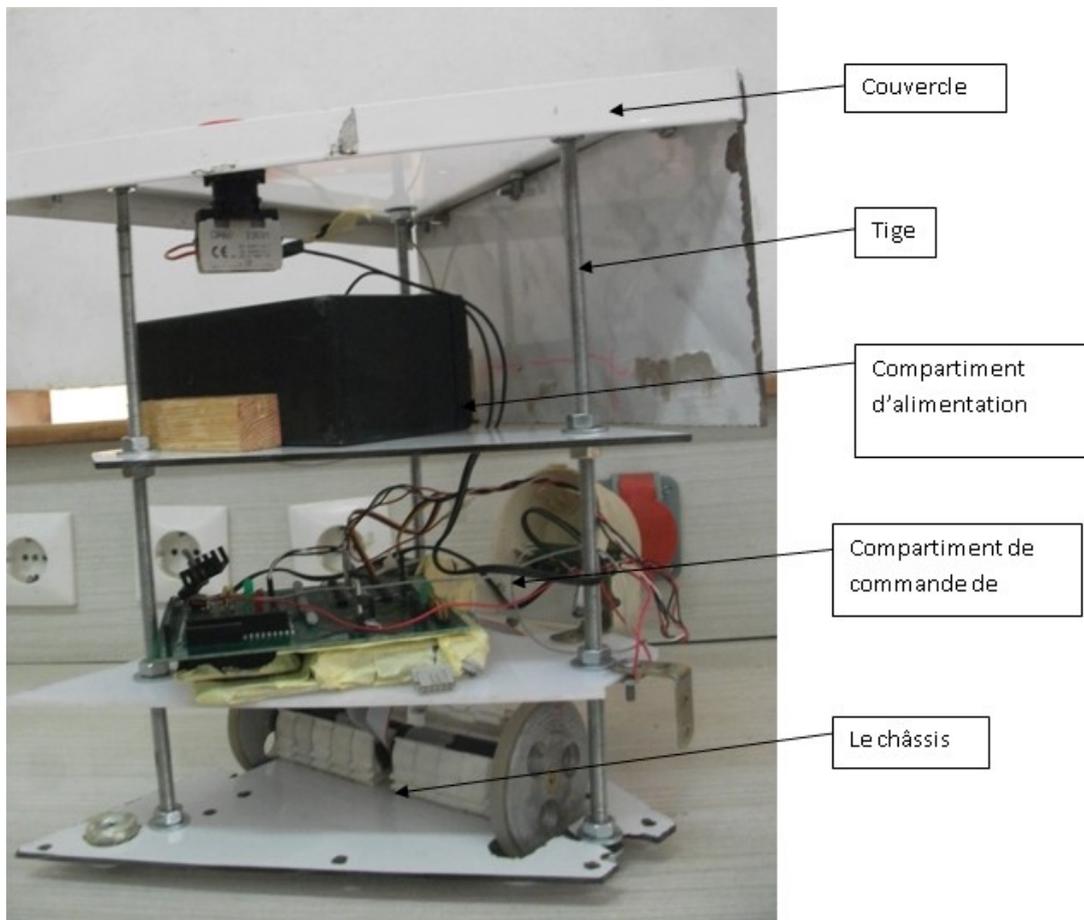
### **2.3.3 Le couvercle :**

Comme les étages précédemment défini, c'est un triangle en résine de 32 cm, ou on a placé le bouton d'arrêt d'urgence.



**Figure 12.** Bouton d'arrêt d'urgence

Les quatre étages du robot sont maintenus parallèlement superposés à l'aide de trois tiges de 34cm de hauteur comme le montre la figure13.



**Figure 13.** Photo du robot.

**Remarque :**

A travers les tests qu'on a effectués, nous avons réalisé la nécessité de l'utilisation d'encodeur externe, que nous avons ajouté.

**Conclusion :**

La partie mécanique du robot est la partie la plus difficile, qui prend le plus de temps, et qui demande beaucoup de précision.

# Chapitre 3 Électronique du robot

---

## 3.1 la carte électronique :

### 3.1.1 Introduction :

Pour la mise en marche de tous les systèmes mécaniques vus dans le chapitre précédent, notre robot doit disposer d'une carte électronique qui assure le bon fonctionnement de celui ci. Nous disposons d'une carte de puissance (figure 14) à double face qui assure la marche des moteurs à courant continue, les capteurs du robot et le servomoteur.



*Figure14.* La carte électronique.

### 3.1.2 Principe de fonctionnement de la commande du robot :

#### *a* **Éléments du robot :**

- ✓ Deux moteurs à courant continue
- ✓ Carte principale : c'est la carte arduino mega 2560.
- ✓ Un télémètre

- ✓ La CMUcam
- ✓ Servomoteur
- ✓ Un carte de puissance.

***b Description fonctionnelle de la carte électronique :***

Le schéma synoptique de l'électronique du robot montre les éléments électroniques principaux conçus pour la commande des éléments mécaniques du robot.

Notre carte de puissance intégrée avec la carte de commande principale (arduino mega 2560) qui sert à :

- ✓ Commander les deux roues des moteurs a courant continue servant aux déplacements du robot
- ✓ Commander un servomoteur pour la rotation de la CMUcam.

Et la carte principale de commande (arduino mega 2560) qui reçoit des informations sert a :

- ✓ La détection du robot adverse grâce à un télémètre infra rouge
- ✓ Une CMUcam qui détecte la couleur de l'objet

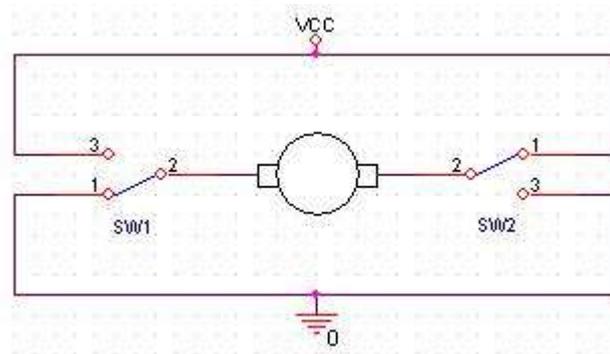
***c Le pont h***



***Figure 15.*** Pont h(L298)

C'est un montage quasiment indispensable pour les « robots à roulettes ». Lorsqu'on pilote un moteur "à courant continu" ou même "pas à pas", il faut changer

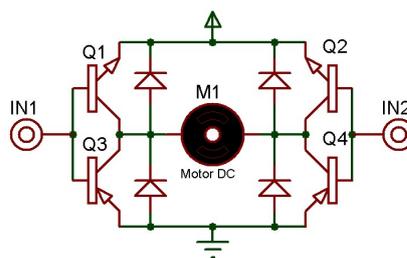
la polarité pour inverser le sens de rotation ou envoyer un train d'impulsions (PWM) pour faire varier la vitesse. La solution à ces deux problèmes se trouve dans le montage que l'on appelle le pont en H. Son principe de base est schématisé (figure16) ici avec des interrupteurs :



**Figure16.** Schéma illustrant le principe de fonctionnement du pont H

Pour utiliser réellement ce principe, on peut remplacer les interrupteurs par des transistors. Les montages réels prévoient des sécurités supplémentaires qui évitent les courts-circuits si les signaux de commande sont mal synchronisés.

Attention aussi aux courants importants qui vont traverser les transistors si les moteurs sont bloqués. Le moteur est alors équivalent à une résistance très faible et le courant important risque de griller le circuit de commande. Afin de protéger les transistors, on utilise des diodes de roue libre avec le pont h (figure17).

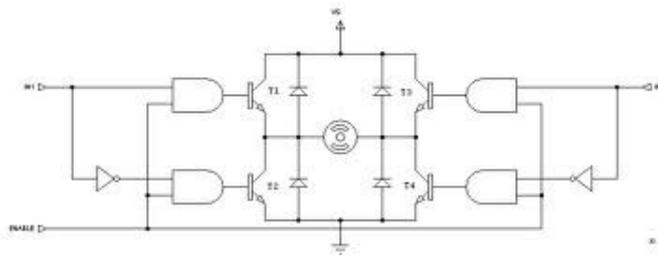


**Figure 17.** montage en pont

Le schéma ci-dessus représente le schéma interne du pont en H. Quand l'entrée In 1 est active et l'entrée In2 est inactive, le transistor Q1 est passant et le transistor Q3 est bloqué. Quand In 1 est actif, le pôle 1 du moteur est au potentiel Vcc, et lorsque, In 1 est à l'état bas, le pôle 1 du moteur est à la masse.

**d Exemple du circuit L298 :**

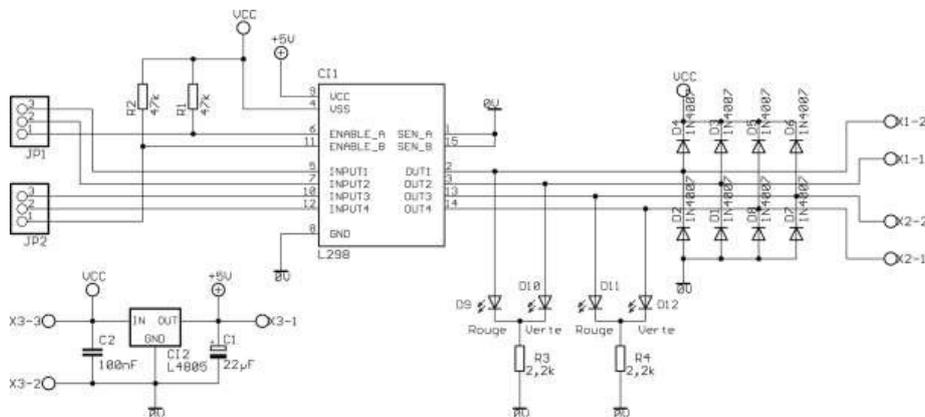
La figure (figure18) montre une partie du schéma interne du L298



**Figure 18.** Schéma interne du L298

**e schéma électronique :**

La figure (figure 19) montre schéma électronique L298

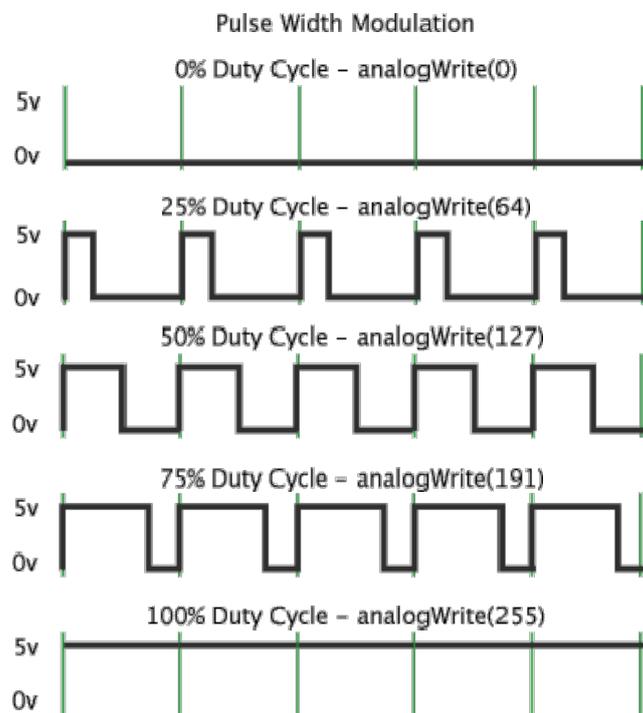


**Figure 19.** Schéma électronique du L298

**f Modulation de la largeur d'impulsion :**

La modulation de la largeur d'impulsion (MLI ou PWM), est une technique qui permet d'obtenir des effets d'allure analogique avec des broches numériques. Le contrôle numérique est utilisé pour créer une onde carrée, un signal basculant entre un niveau HAUT et un niveau BAS, 0V et 5V. Cette succession de niveaux HAUT/BAS peut simuler des tensions entre le niveau HAUT (5 Volts) et le niveau BAS (0 Volts) en faisant varier la proportion du temps où le signal est HAUT sur la proportion de temps où le signal est BAS. La durée du temps du niveau HAUT est appelé largeur d'impulsion, ou encore "duty cycle" (cycle de travail en français). Pour obtenir une variation analogique, il suffit de changer ou de modifier cette largeur d'impulsion.

Dans le graphique ci-dessous (figure 20), les lignes vertes représentent une période de temps régulière. Cette durée, ou période, est l'inverse de la fréquence du signal PWM (qui est une onde carrée). En d'autres termes, avec une fréquence PWM générée par la carte Arduino de l'ordre de 500 Hz, l'espace entre 2 lignes mesure 2 millisecondes. Un appel de la fonction `analogWrite(valeur)` utilise une valeur comprise entre 0 et 255, tel que `analogWrite(255)` utilise 100% du cycle (toujours au niveau HAUT), et `analogWrite(127)` utilise 50% du cycle de travail (la moitié du temps) par exemple. La valeur 0 correspond ainsi à 0% du cycle de travail (duty cycle).



**Figure 20.** Modulation de la largeur d'impulsion

### 3.2 Les capteurs du robot :

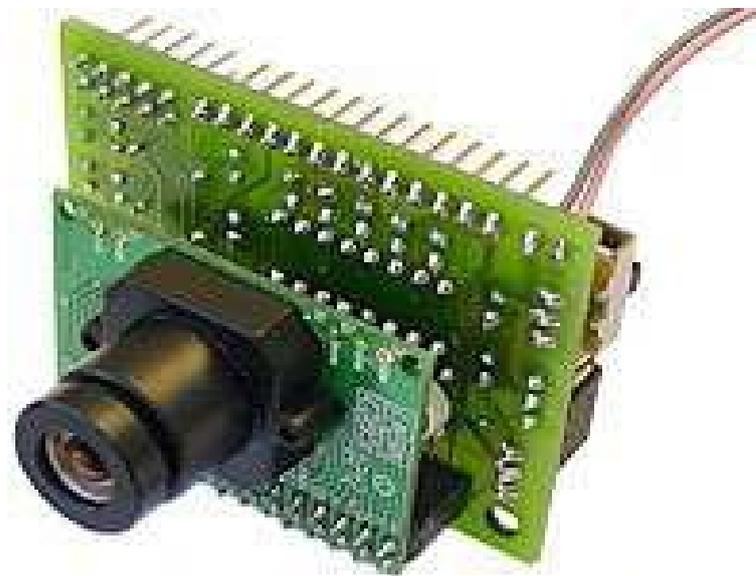
Les capteurs ont une place importante dans le système de traitement d'un robot. Ils peuvent à la fois informer le robot sur le milieu extérieur et l'informer sur ses propres actions en vérifiant l'état de ses actionneurs. Ils sont donc l'élément indispensable à un robot autonome pour savoir ce qu'il fait, ce qui se passe et prendre les bonnes décisions en conséquence.

### **3.2.1 Capteur de couleur :**

Nous avons d'abord pensé à utiliser le matériel disponible, c'est pourquoi nous nous sommes lancés dans l'étude de la CMUCam afin de détecter la couleur de l'objet

#### ***a La CMUcam1 :***

La CMUCam (figure21) est un module utilisé en robotique lorsque le robot a besoin de repérer un objet de couleur donnée. Ce module est une caméra qui intègre un microcontrôleur avec des algorithmes de traitement d'images déjà implantés



***Figure 21.*** La CMUCam

***b Caractéristiques :***

- ✓ Dimensions: 57 x 49 x 50 mm.
- ✓ Alimentation: 6 à 7 Vcc.
- ✓ - Consommation: 200 mA.
- ✓ - Port RS-232 configurable de 9600 à 115.200 bds.
- ✓ - RS-232 avec niveau logique 0-5 V et +/- 10 V pour interfaçage direct par PC.
- ✓ - Capteur vidéo CMOS couleur "OV6620" (Omnivision)
- ✓ - Micro-contrôleur: SX28AC (Ubicom), fréquence 75 Mhz, 2 Ko de mémoire Flash, 136 octets de RAM
- ✓ - Suivie de couleur à 17 images par secondes
- ✓ - Résolution maximale gérée par le microcontrôleur : 143 x 80 pixels.
- ✓ - Sortie pour commande d'un servomoteur (livrable en option).
- ✓ - Led de visualisation [2]

***c Fonctionnalités :***

La CMUcam permet diverses opérations:

- ✓ Suivre la position d'un objet coloré
- ✓ Mesurer les statistiques RGB d'une portion d'image
- ✓ Se "caler" et suivre le premier objet qu'il voit
- ✓ Suivre "physiquement" le déplacement d'un objet en utilisant l'axe d'un servomoteur comme support
- ✓ Récupérer une image au travers du port série.

***d Multi-seuillage couleur et le suivi d'objets :***

Cette partie est la plus importante du logiciel puisqu'elle permet le suivi d'objets via leur couleur. Plusieurs méthodes ont été réalisées :

- ✓ Détection couleur au centre et suivi.
- ✓ Suivi couleur.
- ✓ Choix de couleur à suivre.

### 3.2.2 Détecteur d'obstacle :

Il existe plusieurs moyens pour détecter les obstacles par exemple la famille des micro-switches, les détecteurs infrarouges et aussi les détecteurs à ultrasons. Dans notre cas nous avons utilisé le télémètre infrarouge SHARP GP2D15 .

#### *a Capteur infrarouge SHARP GP2D15 :*



*Figure 22.* Télémètre infrarouge SHARP GP2D15

#### *b Présentation :*

Le capteur GP2D15 est un capteur de distance :

- ✓ opérant dans une plage de mesure de 10 à 80 cm.
- ✓ infrarouge (utilise un système optique lumineux dans une longueur d'onde invisible à l'œil nu)
- ✓ non-linéaire (la tension de sortie n'est pas directement proportionnelle à la distance)
- ✓ fonctionnant sous 5V

**c 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.

**d Schéma de branchement :**

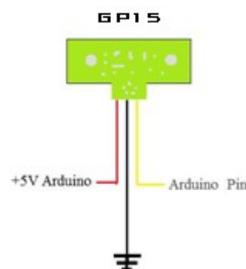
Le SHARP contient trois pins : (Figure 23)

-VCC : entrée alimentation 5V

-GND : la masse

-V0 tension de sortie : elle est à 0v sauf quand elle reçoit un signal

Nous avons branché V0 dans une entrée d'interruption du arduino (pin) quand elle détecte un obstacle cette entrée passe à l'état haut arduino et détecte le changement d'état et va arrêter les moteurs.



**Figure 23.** Schéma de branchement

### e Montage avec une carte Arduino :

La figure 24 montre schéma de branchement avec la carte

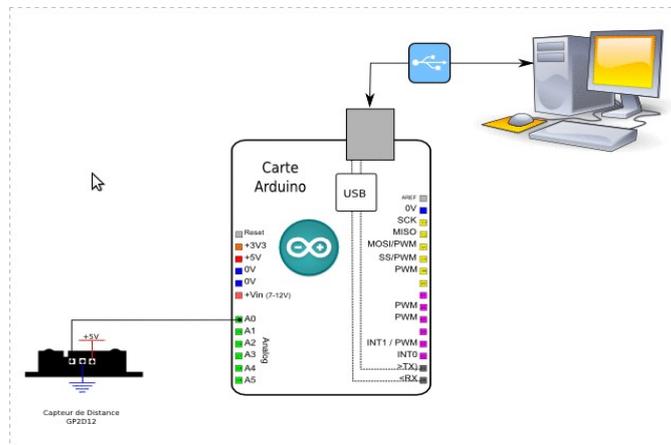


Figure24. Schéma du branchement du capteur avec la carte

## 3.3 la carte Arduino :

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. La carte Arduino repose sur un circuit intégré (un mini ordinateur appelé également microcontrôleur) associée à des entrées / sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes.

Il existe plusieurs variétés de cartes Arduino et nous nous sommes lancé dans l'étude de la carte Arduino mega2560

### 3.3.1 La carte Arduino méga 2560 :



Figure25. Cartes Arduino méga 2560

### 3.3.2 Présentation :

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560

Cette carte dispose :

- ✓ de 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- ✓ de 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- ✓ de 4 UART (port série matériel),
- ✓ d'un quartz 16Mhz,
- ✓ d'une connexion USB,
- ✓ d'un connecteur d'alimentation jack,
- ✓ d'un connecteur ICSP (programmation "in-circuit"),
- ✓ et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB

### 3.3.3 Synthèse des caractéristiques :

- ✓ Microcontrôleur : AT mega2560
- ✓ Tension de fonctionnement : 5 V
- ✓ Tension d'alimentation (recommandée) : 7-12 V
- ✓ Tension d'alimentation (limites) : 6-20V
- ✓ Broches E /S numériques : 54(dont 14 disposent d'une sortie PWM)
- ✓ Broches d'entrées analogiques : 16(utilisable en broches E/S numériques)
- ✓ Intensité maxi disponible par broche E/S (5V) : 40 mA
- ✓ Intensité maxi disponible pour la sortie 3.3 V : 50 mA
- ✓ Intensité maxi disponible pour la sortie 5 V : Fonction de l'alimentation utilisé
- ✓ Mémoire programme flash : 256 KB dont 8 KB sont utilisés par le boot loader

- ✓ Mémoire SRAM (mémoire volatile) : 8KB
- ✓ Mémoire EEPROM (mémoire non volatile) : 4KB
- ✓ Vitesse d'horloge 16 MHz

### 3.3.4 Alimentation

La carte Arduino Mega 2560 peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

La carte peut fonctionner avec une alimentation externe de 6 à 12 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte.

Les broches d'alimentation sont les suivantes :

- ✓ **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée).
- ✓ **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte
- ✓ **3V3** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega)
- ✓ **GND** : Broche de masse (0V).

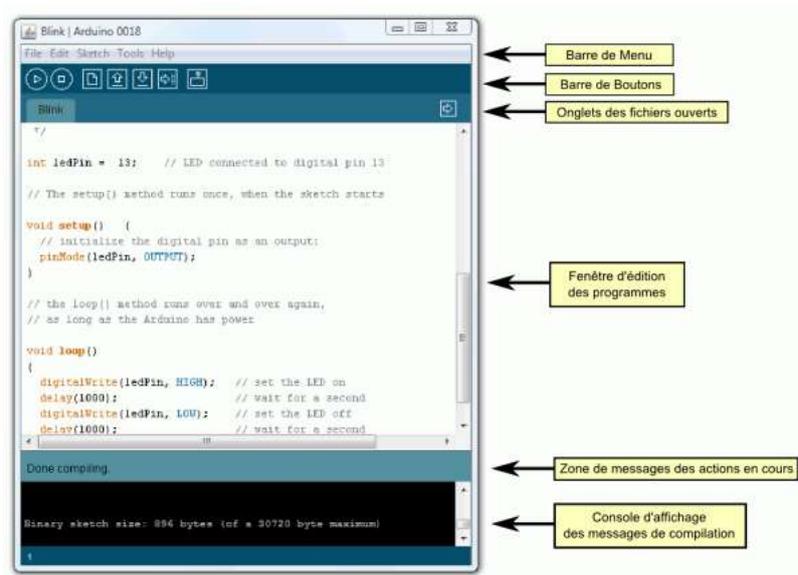
### 3.3.5 Espace de développement intégré (EDI) Arduino :

Le logiciel Arduino a pour fonctions principales :

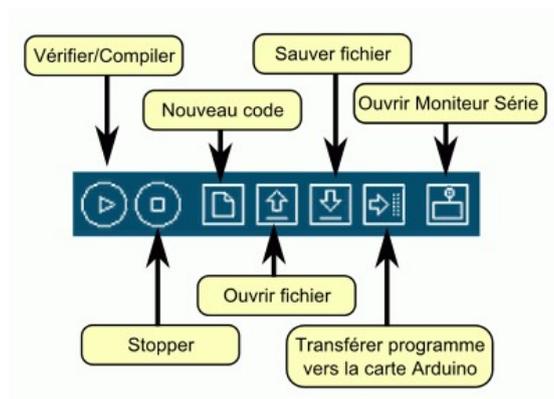
- ✓ de pouvoir écrire et compiler des programmes pour la carte Arduino
- ✓ de se connecter avec la carte Arduino pour y transférer les programmes
- ✓ de communiquer avec la carte Arduino

Cet espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino comporte :

- ✓ une **BARRE DE MENUS** comme pour tout logiciel une interface graphique (GUI),
- ✓ une **BARRE DE BOUTONS** Figure27 qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation,
- ✓ un **EDITEUR** pour écrire le code du programme, avec onglets de navigation,
- ✓ une **ZONE DE MESSAGES** qui affiche indique l'état des actions en cours,
- ✓ une **CONSOLE TEXTE** qui affiche les messages concernant le résultat de la compilation du programme



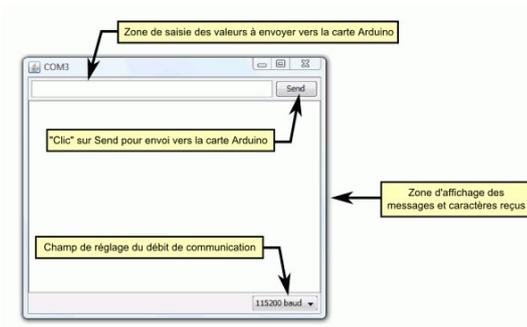
**Figure 26.**espace de développement des cartes Arduino



**Figure27.** la barre des boutons

Le logiciel Arduino intègre également :

- ✓ un **TERMINAL SERIE** (figure 28) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.



**Figure 28.** Terminal série

### 3.4 servomoteur :

Un servomoteur (figure 29) est un moteur conçu pour générer un mouvement précis selon une commande externe. D'habitude, son mouvement est limité à un arc de 180 degrés.



*Figure 29.* Servomoteur

Les servomoteurs ont 3 fils de connexion :

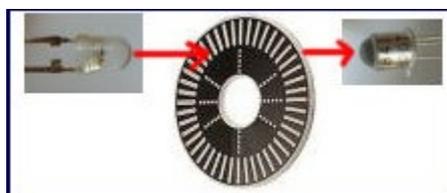
- ✓ le + .
- ✓ le - (ou la masse) .
- ✓ la commande du servomoteur : une impulsion modulée en largeur (PWM) qui définit la position de l'axe du servomoteur.

Dans cas de l'utilisation d'un servomoteur standard :

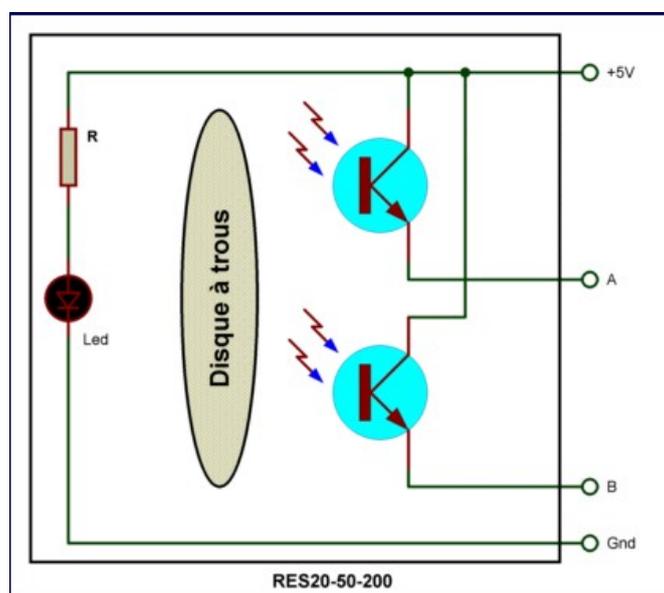
- ✓ Le fil du +, est typiquement rouge, et doit être connecté à la broche 5V de la carte Arduino.
- ✓ Le fil de la masse est typiquement noir et doit être connecté au 0V (= à la masse) de la carte Arduino.
- ✓ Le fil de l'impulsion de commande est typiquement jaune et doit être connecté à une broche numérique de la carte Arduino.

### 3.5 Les encodeurs optiques :

Les encodeurs optiques sont les capteurs les plus utilisés pour mesurer la vitesse et la position angulaire de l'axe d'un moteur ou celle d'un système de guidage , le principe de fonctionnement de ces encodeurs est basé sur l'utilisation de photodiodes ou phototransistor qui deviennent " passants " lorsqu'ils sont soumis à une certaine intensité lumineuse pour transférer la vitesse ou la position d'un rotor en une donnée exploitable par un processeur. Une source de lumière fixe est altérée par des zones opaques et transparentes d'un disque solidaire de ce rotor la figure illustre ce principe de fonctionnement .

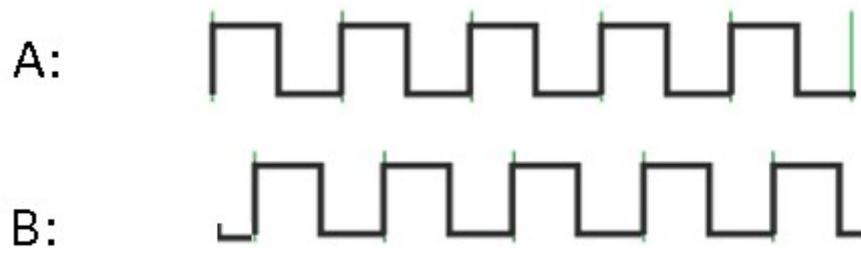


**Figure 30.** Principe de fonctionnement des encodeurs



**Figure31.** Schéma électrique d'encodeur optique

- ✓ Etant donné leurs positions vis-à-vis de la source lumineuse, les récepteurs reçoivent les signaux A et B suivant :



- ✓ L'acquisition des signaux A et B nous permet d'obtenir la vitesse des rotors ( $f = \frac{1}{T}$ ) ainsi que le sens de rotations du moteur.

# Chapitre 4 Programmation du robot

---

## 4.1 Introduction :

Pour atteindre l'objectif fixé au préalable, nous avons opté pour deux approches différentes

La première approche consiste à faire déplacer le robot mobil vers un objet fixe.

La deuxième approche consiste reconnaître et poursuivre l'objet.

## 4.2 Programmation :

### 4.2.1 La première approche :

Toute la première approche a été basée sur la supposition que nous connaissons l'endroit de l'objet vers le quel le robot devra se déplacer, de ce fait et à l'aide d'encodeur et de capteur infrarouge, nous avons essayé de diriger notre robot.

L'objet a été mis à 700 mm du robot en ligne droite, mais impossible de se diriger tout droit à cause de la barrière en bois, du coup le déplacement du robot se fera en trois étapes :

a. La première étape :

Le robot fait une ligne droite de 600 mm

b. La deuxième étape :

Le robot tourne à droite avec un angle  $135^\circ$

c. La troisième étape :

Le robot fait une deuxième ligne droite de 870 mm



Ce qui veut dire que quand les moteurs des roues font un tour, le robot se déplacera de 25.12cm, et l'encodeur délivre 512 impulsions.

Donc :

25,12 cm       $\longrightarrow$       512 impulsions

60 cm       $\longrightarrow$       X impulsions

$$x = \frac{512 * 60}{25,12}$$

$$x = 1222 \text{ impulsions}$$

Lors de la programmation du robot, nous avons programmé les PWM a une puissance maximale, mais le robot a dépassé 60 cm, nous avons répété l'expérience plusieurs fois mais à chaque fois nous avons eue le même résultat.

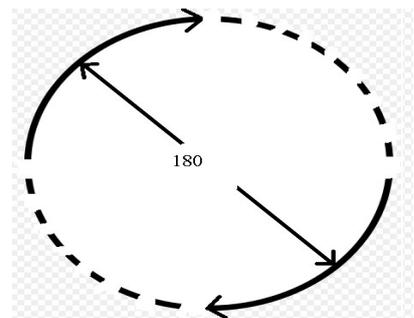
Alors nous avons pensé a diminuer le PWM et nous avons remarqué qu'à chaque fois, la distance s'approche de la distance voulue, et à un PWM égal 128 nous avons pus obtenir les 60 cm, mais a un PWM égal a128, on perd beaucoup de puissance des moteurs et donc beaucoup de temps.

Nous avons programmé notre robot de telle sorte, que quand il arrive au nombre d'impulsion voulue, il s'arrête et passe à l'instruction suivante. Dans ce cas, il tourne à droite, et en cas d'obstacle observé par le capteur infrarouge, il marque un stop.

## Etape2 : Faire une rotation de 135° à gauche

Pour faire tourné le robot à gauche, on a décidé de faire tourné la roue droite en arrière et la roue gauche en avant, ce qui nous donne un cercle de diamètre 18cm (la distance séparant les deux roues).

Le calcul du nombre d'impulsion de l'encodeur :



Pour cela on doit calculer la circonférence du cercle causé par le mouvement de deux roues,

La circonférence C du cercle causé pas le mouvement des roues :

$$c = \pi * d$$

$$c = 3,14 * 18$$

$$c = 56,52 \text{ cm}$$

Chacune des roues fera 3/8 du cercle, ce qui donne 21,195 cm

25,12cm       $\longrightarrow$  512 impulsions

21,195cm       $\longrightarrow$  X impulsions

$$x = \frac{512 * 21,195}{25,12}$$

$$x = 432 \text{ impulsions}$$

Etape3 : Faire marcher le robot en ligne droite de 880 mm

Comme la première étape, il faut calculer le nombre d'impulsion équivalent à 880 cm

Calcul du nombre d'impulsion de l'encodeur :

25,12cm       $\longrightarrow$  512 impulsions

88 cm       $\longrightarrow$  X impulsions

$$x = \frac{512 * 88}{25,12}$$

$$x = 1794 \text{ impulsions}$$

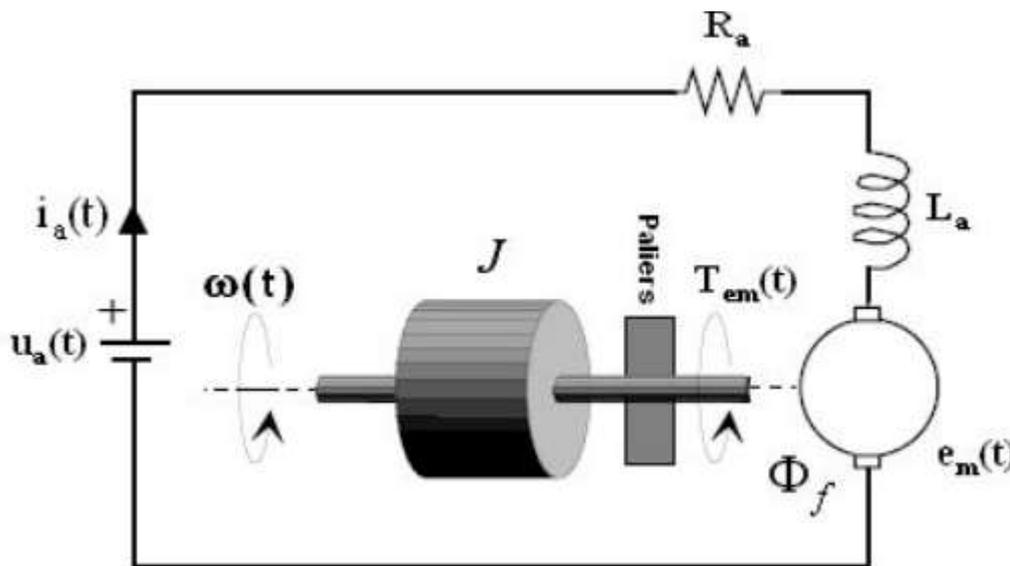
#### **4.2.2 La deuxième approche :**

Toute la deuxième approche a été basée sur le fait que l'objet n'est pas fixe, donc c'est la CMU CAM qui doit le reconnaître (reconnaître la couleur) le localiser, et nous donner des informations sur sa position, sa taille, et à travers ces informations, nous avons programmé le robot comme le montre les algorithmes.

#### 4.4 Modalisation du moteur courant continu :

la machine à courant continu se définit par un circuit d'induit et un circuit inducteur ; l'induit du moteur à courant continu peut être vu comme une résistance  $R$  et une inductance  $L$  en série avec une source de tension commandée  $e(t)$  et Proportionnelle à la vitesse  $\Omega(t)$ .

Le schéma technologique d'une MCC est représenté sur la figure suivante :



**Figure 33.** Schéma d'un entraînement avec une MCC à excitation indépendante.

Les signaux y intervenant sont les suivants :

- La tension aux bornes de l'induit  $u(t)$ .
- le circuit électrique de l'induit, faisant apparaître :
  - La résistance de l'induit  $R$ .
  - L'inductance de l'induit  $L$ .
- Une tension  $e_m(t)$  appelée f.e.m. (force électromotrice), proportionnelle à la vitesse angulaire  $\Omega(t)$ .
- Le courant traversant le circuit d'induit  $I(t)$  ;
- Le couple électromagnétique instantané  $T_{em}(t)$  produit ;

- L'inducteur, fixé au stator, créant un flux magnétique d'excitation  $\Phi_f$  ;
- La charge mécanique, dépendante de l'application (inertie J, frottement visqueux, élasticité de la transmission, etc.) ;

La vitesse  $\omega(t)$  du rotor du moteur.

$$\begin{cases} T_{em}(t) = \Phi * i = J\ddot{\omega} = J\dot{\omega} \\ e = \Phi\dot{\omega} \\ E = Ri + L\frac{di}{dt} + e \end{cases}$$

$$\text{Ou bien } \begin{cases} \frac{d\omega}{dt} = \frac{\Phi}{J} i(t) \\ \frac{di}{dt} = \frac{1}{L}(E - Ri(t) - \Phi\omega(t)) \end{cases}$$

Ainsi nous obtenons un système de deux équations différentiels couplées, afin de résoudre ce système d'équation nous pouvons prendre l'intégrale des 2 termes de ces équations et utiliser un logiciel de simulation tel que MATLAB pour obtenir les réponses du système en fonction d'un signal d'entrée E pour compléter le modèle précédent et s'approcher de système réel, il est nécessaire de tenir compte des frottements existants au niveau des contacts mécanique, pour cela l'équation donnant le couple s'exerçant sur l'axe du rotor de vient

$$T_{em}(t) = J\dot{\omega} + f\omega$$

F étaient le coefficient du frottement (amortissement) ainsi notre système d'équation doivent :

$$\begin{cases} \Phi i(t) = J\dot{\omega} + f\omega \\ \frac{di}{dt} = \frac{1}{L}(E - Ri(t) - \Phi\omega(t)) \end{cases}$$

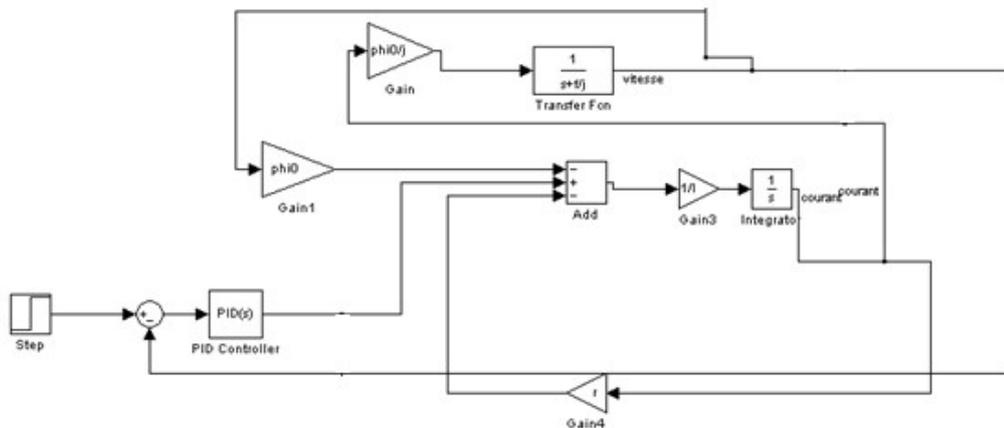
En utilisant la transformer de la place et en conditions initiales nulles  $i(0)=0$  et  $\omega(0)=0$  nous avons :

$$\begin{cases} \phi = J\omega(s) + f\omega(s) \\ sI(s) = \frac{1}{L}(E(s) - Ri(s) - \phi\omega(s)) \end{cases}$$

$$\Rightarrow (Js+f)\omega(s) = \phi I(s) \Rightarrow$$

$$\begin{cases} \omega(s) = \frac{\phi}{J} \left( \frac{1}{s + \frac{f}{J}} \right) I(s) \\ I(s) = \frac{1}{s} \left( \frac{1}{L} \left( E(s) - Ri(s) - \frac{\phi\omega(s)}{s} \right) \right) \end{cases}$$

Ce system peut être résolu par le diagramme bloc simulink suivant :



Résultat de simulation pour un moteur maxon référence 418537

Les données du constructeur sont :

$$R=6.01\Omega$$

$$L=0.362\text{ H}$$

$$\phi = 10.9 * 10^{-3} \text{ Nm/A}$$

$$J=4.29*10^{-7} \text{ Kg m}^2$$

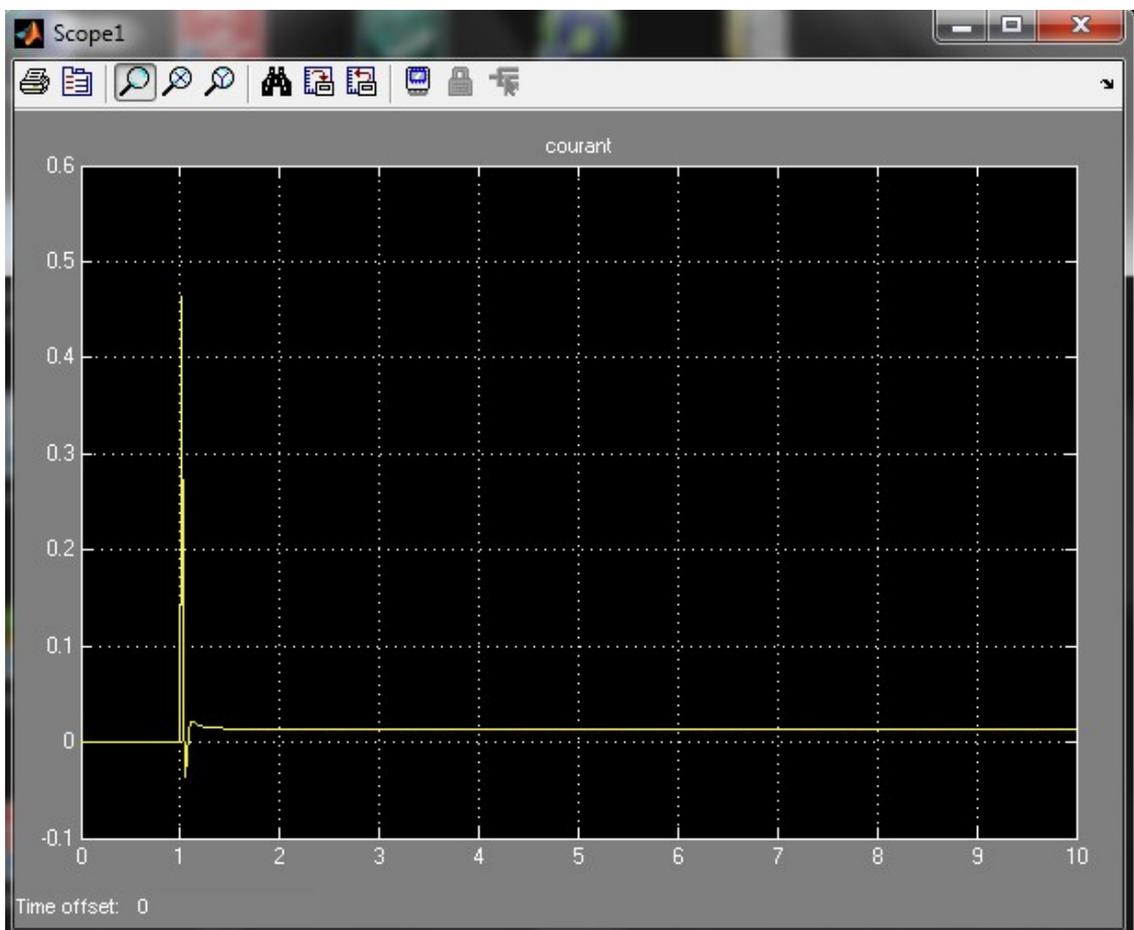
$$F=7.75* 10^{-7} Nm/rad$$

Sur la notice du conducteur sont donnés aussi d'autres caractéristiques telles que la puissance du moteur ainsi que la vitesse maximale à vide

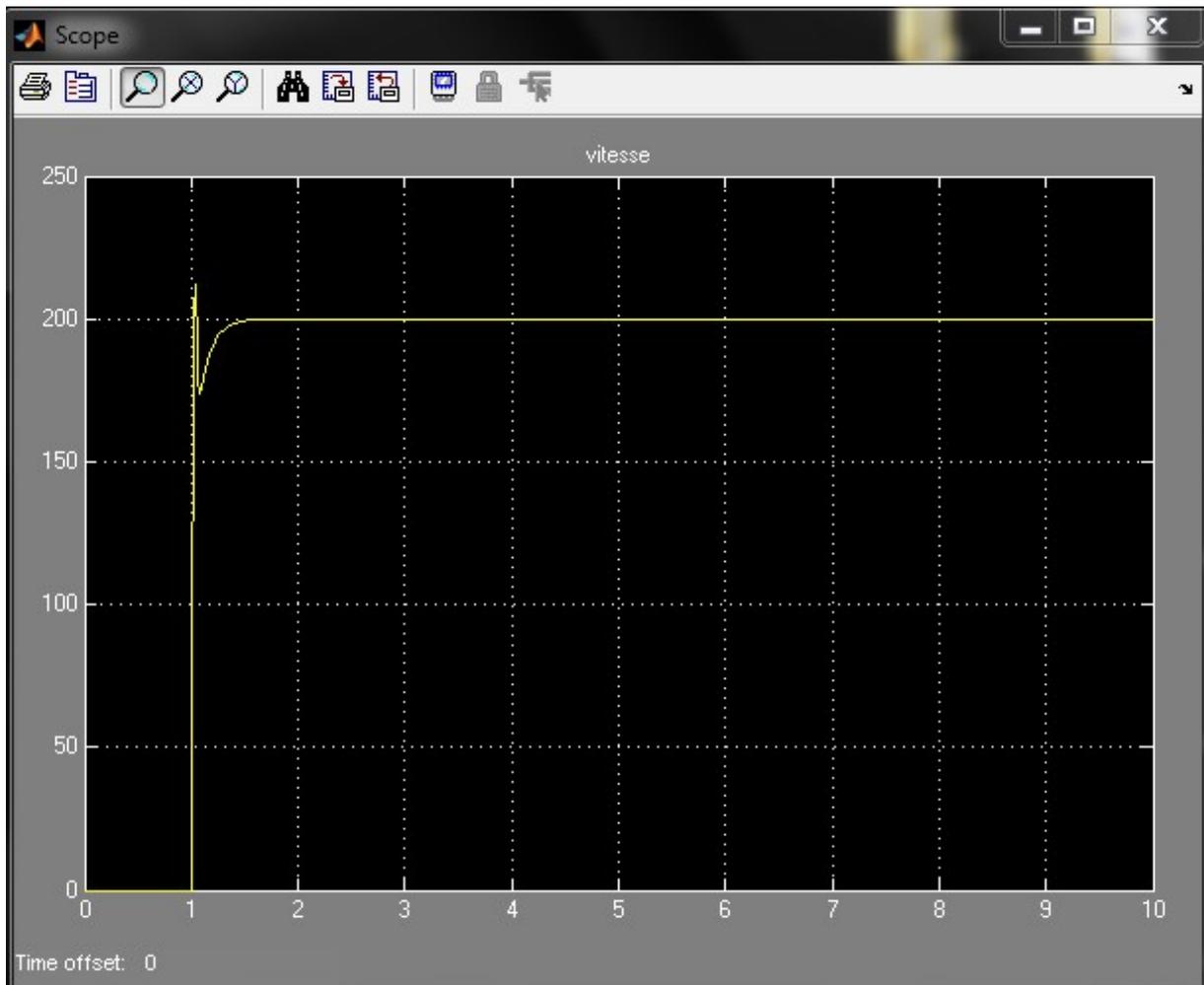
Pour ce type de moteur la vitesse à vide est  $w_0$

$w_0=8430 \text{ tour}/\text{min}=882 \text{ rad}/\text{sec}$  pour obtenir cette vitesse le coefficient de frottement est déterminé en effectuant plusieurs simulations

Figure . Résultats de la simulation des maxon 418537 avec  $E=10V$

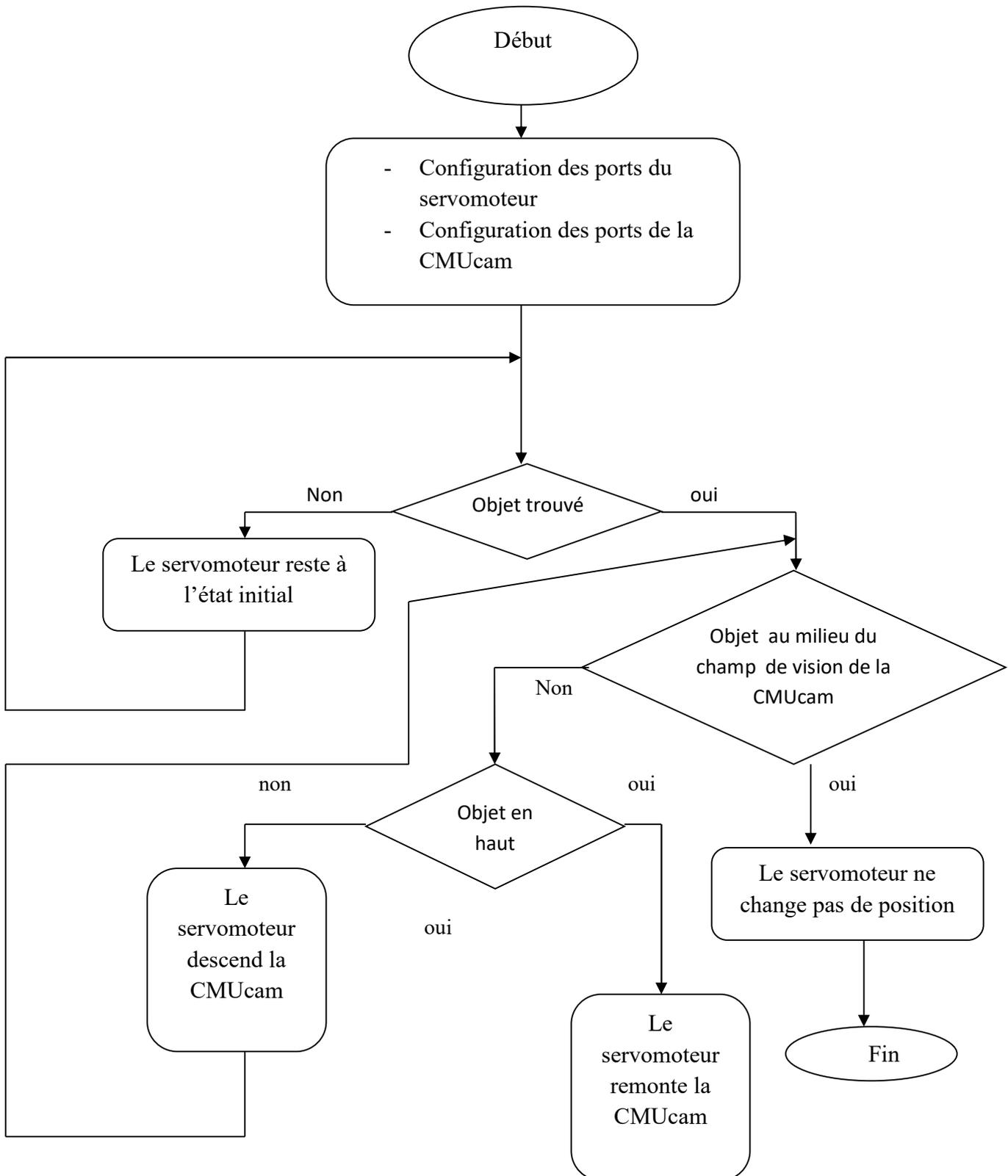


**Figure 34.** Schéma du courant

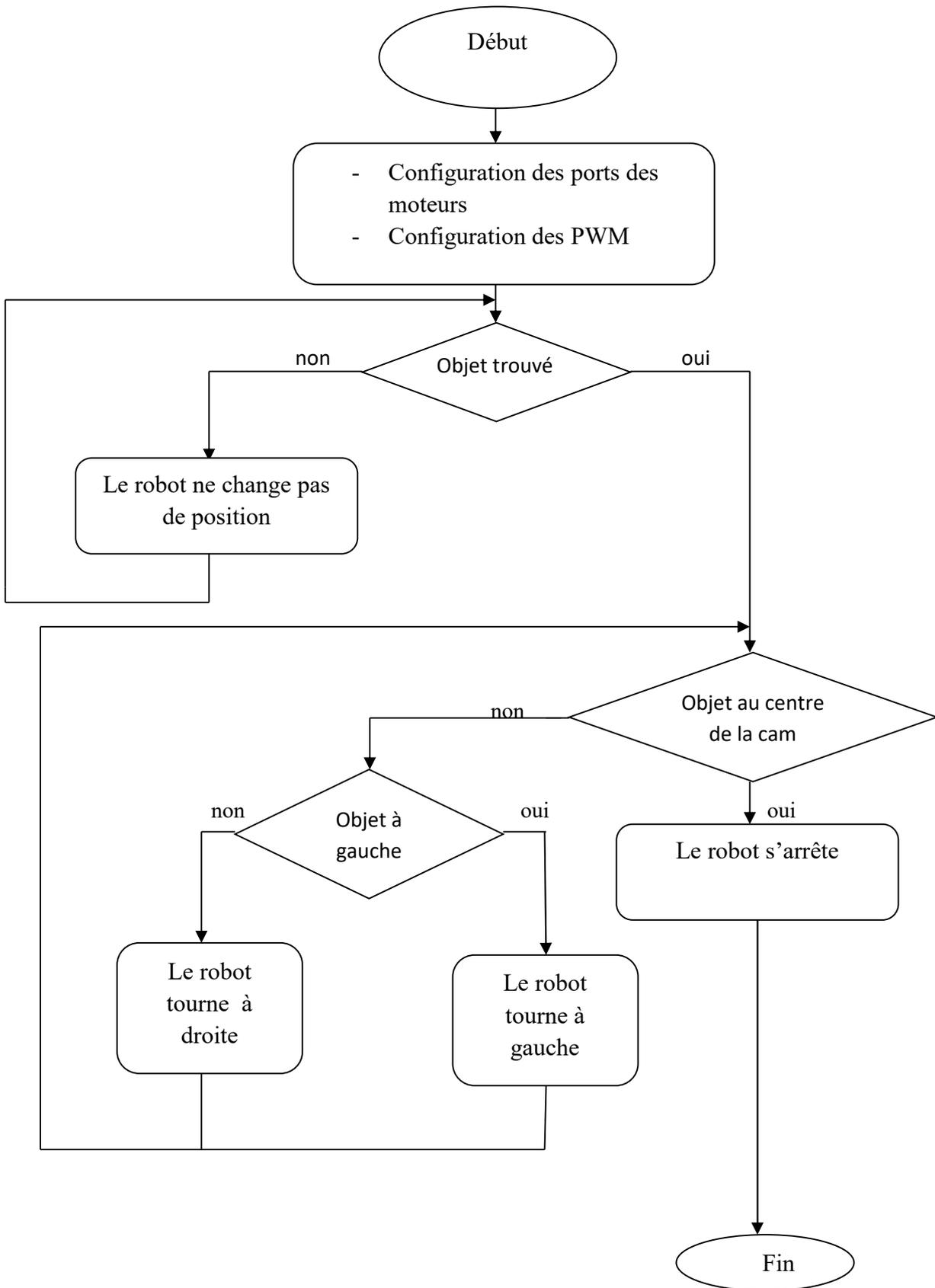


*Figure 35.* Schéma de la vitesse

**C** *algorithme servomoteur :*

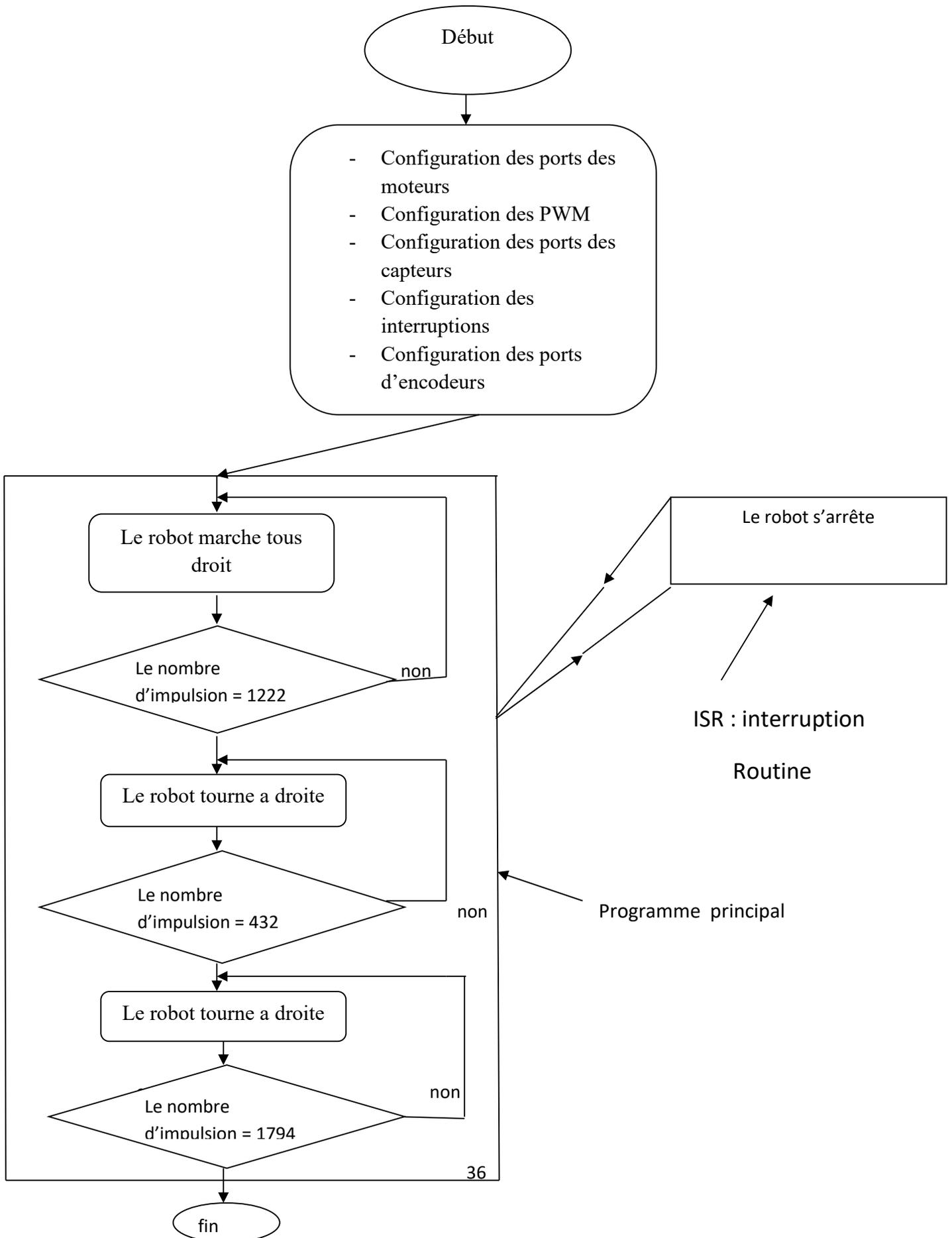


**b** *algorithme droite gauche :*



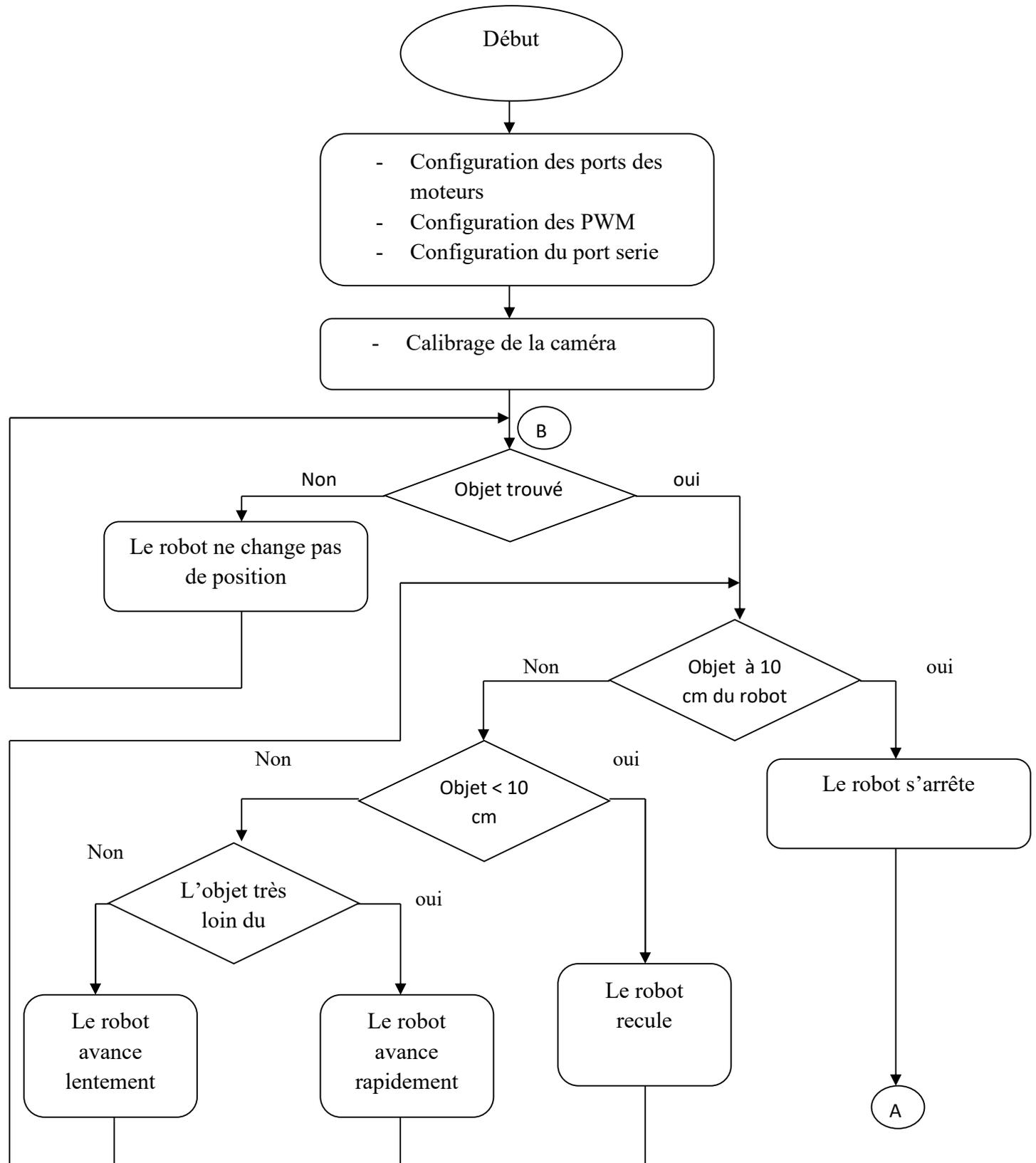
## 4.3 Les algorithmes :

### 4.3.1 Algorithme de la première approche :



### 4.3.2 La deuxième approche :

*a algorithme avant arrière*



## Bibliographie

---

[1] Auteur1. David FILLIAT : Robotique Mobile, École Nationale Supérieure de Techniques Avancées ParisTech,

[2] Auteur1 Hakim LOUAHAB ; Geoffroy VILMOT. Suivi temps réel d'objets par caméra intelligente.2006-2007

## **4.5 Conclusion:**

Dans ce chapitre nous avons présenté le programme de gestion de robot mobile sous forme des algorithmes suivant la structure mécanique et électronique. Ces algorithmes décrivent la stratégie que suit le robot sur l'air du jeu afin de réaliser sa tâche. Pour la programmation nous avons utilisé le langage arduino

## Conclusion générale

---

La robotique est un domaine pluridisciplinaire. Durant ce projet nous avons pu acquérir une certaine expérience pour la conception et la réalisation des différentes parties d'un robot mobile autonome tout en respectant un cahier des charges très particulier, notamment en termes de dimensions et de fonctionnalités. Ce robot a été réalisé dans un premier temps pour participer au concours Eurobot Algérie, qui s'est déroulé les 15 et 16 Avril 2012 à l'université de Blida. Après ce concours et dans le cadre du projet de fin d'études nous avons eu comme complément de travail d'ajouter à ce robot des fonctions de guidage lui permettant d'aller chercher les objets de couleur particulière. Pour ce faire nous avons étudié les possibilités offertes par le module de vision CMUCam qui nous a paru être très adapté à notre conception. Les principales caractéristiques de ce module peuvent se résumer en sa capacité à extraire les caractéristiques essentielles concernant les couleurs fondamentales de l'image acquise par son capteur intégré. Parmi les requêtes acceptées par ce module nous nous sommes particulièrement intéressées à la commande « TC » ( Track Color) , qui nous permet d'obtenir les coordonnées et la taille de l'objet dont la couleur appartient à l'intervalle indiquée avec la commande « TC ». Bien que cela semble évident d'un point vue mise en œuvre, nous avons trouvé énormément de difficultés s'agissant de décoder les trames de données issues de la CMUCam. En effet étant donné la taille variable de chacune des valeurs délivrées par la caméra il fallait concevoir un algorithme capable de décoder et séparer les trames successives envoyées par la CMUCam et ce d'une manière suffisamment souple pour être ensuite traduite en langage C sous l'environnement de développement ARDUINO. Une fois cette partie du travail achevée il suffisait de compléter le programme principal avec les routines nécessaires à faire tourner les deux moteurs de déplacement du robot en fonction de la position de l'objet dans le

champ de vision de la caméra, le positionnement vertical de la caméra est assuré par un servomoteur, vu que l'angle maximal pour ce déplacement est limité.

En termes de difficultés rencontrées au cours de la réalisation de ce projet nous citerons le problème de luminosité nécessaire au bon fonctionnement de la CMUCam, ceci étant l'une des limitations de ce module. La mise en œuvre du travail demandé a nécessité beaucoup d'efforts de notre part, en particulier la construction mécanique du châssis et tout le système de ramassage des objets initialement prévu pour le concours cité précédemment. Bien que nous ayons eu recours à des ateliers de mécanique pour la réalisation des différentes pièces requises, nous nous sommes rendu compte des difficultés et surtout de l'importance qu'il faut accorder à la qualité des pièces mécaniques lorsqu'il s'agit de robotique.

La réalisation d'un robot du genre décrit dans ce mémoire ne peut pas être possible sans la partie électronique indispensable pour acheminer les signaux électriques provenant des différents capteurs et attaquant les actionneurs présents sur le robot. Pour notre cas cette partie a été largement simplifiée grâce à l'utilisation d'un module programmable intégrant suffisamment de fonctionnalités pour gérer toutes les entrées et sorties vers les éléments constituant le robot. La programmation de ce module s'effectue par l'intermédiaire d'un logiciel spécifique fourni par les concepteurs et qui permet un traitement simplifié des entrées et sorties du module utilisant un langage C muni de plusieurs fonctions permettant d'interagir avec les interfaces liées au module ARDUINO utilisé.

Avant de terminer ce mémoire nous devons signaler que malgré toutes les difficultés rencontrées nous avons beaucoup apprécié ce genre de travail puisque cela nous a permis, même si cela reste limité, de mettre en pratique un certain nombre de connaissances acquises durant notre cursus universitaire. Comme tout projet de ce type il nous est impossible d'affirmer que nous avons terminé le travail tel que nous l'aurions souhaité puisque, dans ce domaine, il y a tellement de choses à améliorer ou ajouter que le projet est encore loin d'être achevé. Nous espérons toutefois que ce modeste travail servira comme base d'inspiration pour d'autres travaux dans ce domaine.