

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Mention Électronique
Système de Vision et Robotique

présenté par

MERDJ Hamza

&

BOUADANI Mohamed

Conception et réalisation d'un robot mobile joueur de football

Proposé par : M^r KAZED Boualem

Année Universitaire 2016-2017

Avec un énorme plaisir, un cœur ouvert et d'une joie, que je dédie ce modeste travail à mes très chers, respectueux, et magnifiques parents que nulle dédicace ne puisse exprimer mes sincères sentiments pour leur patience illimitée, leur soutien, leur aide tout au long de ma vie.

A mon cher père

A ma chère mère

En témoignage de mon profond amour et respect pour leurs grands sacrifices. Que dieu les accueille dans son vaste paradis.

Mon binôme Bouadani Mohamed pour sa patience et son encouragement pour accomplir ce projet.

A mes chers amis pour leur encouragement et aide au long de mes études, et à toute ma famille et à ceux que j'aime.

MERDJHAMZA...

*Je dédie ce modeste travail et ma profonde gratitude à mes chères parent qui ont cru en moi, m'ont soutenu et m'ont éclairé le chemin par leur conseil judicieux,
Que dieu les gardes éternellement heureux.*

A mes frères et toute ma grande famille en reconnaissance de leurs encouragements.

A mon binôme MERDJ HAMZA avec qui j'ai partagé les peines et les joies de ce projet de fin d'étude.

A mes très chères amies En leur souhaitant le succès dans leur vie aussi bien professionnelle que familiale.

A tous ceux qui me sont chers.

Remerciements

Nous tenons à remercier Allah tout puissant, nous avoir donné la santé, la volonté et l'aide pour réaliser ce travail.

Nous voulons aussi exprimer par ses quelques lignes de remerciement notre gratitude envers tous ceux en qui par leur présence, leur soutien, leur disponibilité et leur conseil, nous avons eu le courage d'accomplir ce projet.

Nous commençons par remercier Mr. KAZEK BOUALEM qui nous a fait l'honneur d'être notre promoteur.

Nous tenons d'autre part à remercier les responsables membres de jury pour bien vouloir nous accorder de leurs temps pour commenter, discuter et juger notre travail.

Enfin, nous ne pouvons achever ce mémoire sans exprimer notre gratitude à tous les professeurs de l'université de Blida 1 « Saad Dahleb » pour leur dévouement et leur assistance tout au long de nos études universitaires.

ملخص:

الهدف من هذا المشروع هو تصميم وإنجاز رجل آلي متحرك مستقل يلعب كرة قدم، مجهز ببوصلة لتحديد اتجاه الهدف المراد الوصول إليه، وكاميرا محمولة لمعالجة الصور. الرجل الآلي يبحث عن الكرة ويأخذها إلى الهدف وذلك بواسطة المحركات الكهربائية ذات تيار مستمر والمحركات المؤازرة. كل عناصر الرجل الآلي متحكم فيها بواسطة لوحة التحكم أردوينو ميغا وأدوينو أونو

كلمات المفاتيح: أردوينو ميغا، وأدوينو أونو، كاميرا

Résumé :

le but de ce projet est de concevoir et réaliser un robot mobile autonome joueur football équipé d'un capteur boussole pour l'orientation du robot par rapport au but voulu, une caméra embarquée CMUCAM5 qui fait le traitement d'image. Le robot cherche la balle et la prendre vers l'objectif. Toutes sont effectuées à l'aide des moteurs à courant continu et des servomoteurs, commandées par la carte arduino Mega 2560 et arduino Uno 328p.

Mots clés : arduino Mega, CMUCAM5, arduino Uno

Abstract :

The goal of this project is to design and realize a mobile robot autonomous player football equipped with a compass sensor for the orientation of the robot in relation to the intended purpose, an embedded CMUCAM5 camera that does the image processing. The robot looks for the ball and takes it towards the target. All are made using DC motors and servo motors, controlled by the arduino board Mega 2560 and arduino Uno 328p.

Keywords : arduino Mega, CMUCAM5, arduino Uno

Conclusion générale

Durant ce projet nous avons pu acquérir une certaine expérience pour la conception et la réalisation des différentes parties du robot mobile autonome.

Nous avons donc pu réaliser l'objectif de notre projet qui est l'utilisation de logiciel et carte électronique arduino pour manipuler un robot mobile a trois roues. Nous avons étudié tout d'abord les principales caractéristiques d'un robot et en particulier le robot mobile. On a ensuite essayé de parler des différents composants utilisés pour construire ce robot: de moteurs vers les capteurs jusqu' arduino et microcontrôleurs pour mieux comprendre son fonctionnement et pouvoir ensuite générer les signaux de commandes qui seront envoyés à partir de la carte électronique.

Ce projet pourra plutôt servir de base pour commander d'autres systèmes ou d'autres robots qu'ils ont des différents taches et applications selon le besoin.

La réalisation de ce type de projet permet de mettre en pratique un grand nombre de connaissance acquises au fil des études et engendre la grande satisfaction de voir son travail se concrétiser réellement. Nous souhaitons beaucoup de courage et réussite à ceux nous suivants dans ce type de projet.

Table des matières

Introduction générale	1
Chapitre1 : généralités	
1.1 Introduction	2
1.2 Les composants de Robots	3
1.2.1. Manipulateur	3
1.2.2. Effecteur finale	3
1.2.3. Actionneurs	3
1.2.4. Capteurs	4
1.2.5. Contrôleur	4
1.2.6. Processeur	4
1.2.7. Logiciel	5
1.3. Les type des robots	5
1.3.1 : les robots manipulateurs	5
1.3.2 Les robots mobiles	5
1.3.2.1 Architecture des robots mobiles	6
1.3.2.2. Les robots à roues	6
1.4 Comparaison des différents types des robots	9
1.5 Conclusion	9
Chapitre 2 : conception mécanique	
2.1 Introduction	10
2.2 Structure	10
2.2.2 Le châssis	10
2.2.4 Les roue	11
2.3. Les actionneurs	12
2.3.1. Moteurs CC (EMG 30)	12
2.2.2 Servomoteur	14
2.5. Fabrication	15

2.6 Conclusion	15
Chapitre 3 : conception électrique	
3.1 Introduction	16
3.2 La carte mère	17
3.3 Les cartes de commandes	17
3.3.1 carte de commande arduino Mega 2560	17
3.3.2 Caractéristique de l'arduino Mega 2560	17
3.3.3 Entrées et sortie numérique	19
3.3.3.1 présentations	19
3.3.3.2 entrées séries asynchrones	19
3.3.3.3 Entrées analogiques	20
3.3.3.4 Fonctionnement et utilisation de l'arduino	20
3.4 Carte d'isolation galvanique	21
3.5 carte de puissance	21
3.5.1 Le but de carte de puissance	21
3.5.2 Le circuit intégré L298N	23
3.5.3 Régulation de la vitesse	25
3.6 Capteur infrarouge	26
3.7 Codeur à effet de HALL	28
3.8 Le Capteur boussole (CMPS03)	28
3.8.1 Caractéristiques techniques	29
3.8.2 Connexion	30
3.9 Module de reconnaissance vidéo CMUCAM05	30
3.9.1Caractéristiques	31
3.9.2 Principe de fonctionnement	32
3.9.3 <i>Distances d'objet et d'image</i>	32
3.9.4 Traitement d'image de la caméra	35
3.9.4.1 Conversion RVB TSL	35
3.9.4.2 : Binarisation de l'image de la teinte	36

3.10 Fabrication	37
3.11 Conclusion	38
Chapitre 4 : partie programmation et gestion des capteurs	
4.1 Introduction	39
4.2 Programmation du l'arduino	39
4.2.1 Langage de programmation	39
4.2.2 Arduino IDE (Environnement de Développement Intégré)	39
4.2.3 Fenêtre général de l'application arduino	40
4.3 Les organigrammes	41
4.3.1 Organigramme de capteur infrarouge	41
4.3.2 Organigramme de robot et la caméra CMUCAM05	42
4.3.3 Organigramme de capteur boussole avec robot	43
4.3.4 Organigramme globale du cycle de robot	44
4.3.4 Organigramme de contrôle de vitesse	45
4.4 Résultats et discussion	46
4.4.1 Réglage des paramètres CMUcam5 Pixy	46
4.4 Conclusion	51
Conclusion générale	52

Liste des figures

Figure 1.1	Un bras manipulateur.....	3
Figure 1.2	Effecteur finale d'un manipulateur.....	3
Figure1.3	Différents actionneurs d'un robot.....	4
Figure 1.4	Un contrôleur utiliser pur commander un robot.....	4
Figure 1.5	Architecture d'un robot mobile.....	6
Figure1.6	Robot de type unicycle.....	7
Figure1.7	Robot de type tricycle.....	7
Figure1.8	Robot de type voiture.....	8
Figure1.9	Robot de type omnidirectionnel.....	8
Figure2.1	schéma du châssis.....	10
Figure2.2	image en position vertical de trois roues du robot.....	11
Figure2.3	les deux types de roues de robot.....	12
Figure2.4	Moteur à CC (EMG 30).....	13
Figure2.5	moteurs placés dans le châssis.....	13
Figure2.6	servomoteur à CC.....	14
Figure 3.1	schéma synoptique général du robot.....	16
Figure3.2	arduino Mega2560.....	18
Figure 3.3	fonctionnement d'un pont en H.....	22
Figure 3.4	le pont en H.....	23
Figure 3.5	schéma interne du L298.....	24
Figure 3.6	schéma développé de la carte de puissance.....	25
Figure 3.7	Exemple de signaux PWM.....	26
Figure 3.8	capteur infrarouge.....	27
Figure 3.9	fonctionnement d'un codeur à effet de HALL.....	29

Figure 3.10 capteur boussole connecté avec l'arduino.....	29
Figure3.11 le module CMUCAM05.....	31
Figure3.12 analyse de la lentille qui montre la relation de l'objet et l'image.....	33
Figure4.1 le compilateur arduino IDE.....	40
Figure 4.2 organigramme de gestion de télémètre.....	41
Figure 4.3 organigramme des décisions à suivre par le robot.....	42
Figure 4.4 organigramme de gestion de capteur boussole.....	43
Figure 4.5 l'organigramme globale.....	44
Figure 4.6 principe du système qui commande la vitesse de moteur en boucle fermée	45
Figure 4.6 robot vision caméra.....	46
Figure 4.7 robot vision caméra avec les deux objets à la fois.....	47
Figure4.8 détection de la balle.....	48
Figure4.9 robot tire la balle par le servomoteur	49
Figure 4.10 robot en face de la barre et tire la balle dans l'objectif.....	50

Liste des tableaux

Tableau 1.1 Les avantages et des inconvénients des différents types de robots à roues	9
Tableau3.1 logique décrivant le fonctionnement du pont en H.....	23
Tableau 3.2 les entrées/sorties du pont en H.....	25
Tableau3.3 les plant RVB	35
Tableau3.4 plat de la teinte.....	36
Tableau4.1 : CMUcam5 Pixy derniers paramètres sélectionnés.....	46

Chapitre 1 : Généralités

1.1 Introduction :

Les robots aujourd'hui ont un impact considérable sur de nombreux aspects de la vie moderne, de la fabrication industrielle aux soins de santé, le transport et l'exploration de l'espace et le profond de la mer. Demain, des robots seront aussi omniprésents et personnelle comme les ordinateurs personnels.

Le rêve de créer des machines qui sont qualifiés et intelligentes a fait partie de l'humanité depuis le début du temps. Ce rêve est en train de devenir une partie de la réalité de notre monde. [1]

Quand les chercheurs ont commencé à réfléchir sur la conception de robots, on appelait « robotique » la science des robots et/ou l'art de concevoir et fabriquer des robots. Les succès des robots industriels, qui travaillaient initialement en poste isolé et qui ont été intégrés à des chaînes de production comme des machines parmi d'autres, ont conduit le public à élargir le sens du terme « robotique » et à le rendre désormais presque synonyme d'automatisation. Le correspondant de robotique devrait naturellement être « automatique » mais ce n'est pas le cas dans le langage courant. Pour compenser ce hiatus se sont créés le mot « robotisation » et le verbe « robotiser » pour désigner la plupart des automatisations et leur mise en œuvre, même en l'absence de véritables robots. Dans certains champs d'applications, des mots calqués sur robotique sont apparus. Il en est ainsi de « domotique » qui désigne tout ce qui concerne l'automatisation dans l'habitat ou « productique » pour tout ce qui a trait aux moyens de production. [2]

Aujourd'hui, la robotique est donc l'art d'automatiser des systèmes plus ou moins complexes mais en s'appuyant sur le savoir-faire acquis par les études sur la conception de robots, savoir-faire issu des développements d'une branche de l'automatique générale. [3]

Chapitre 1 : Généralités

1.2 Les composants de Robots

Un robot, en tant que système, se compose des éléments, qui sont intégrés ensemble pour former un ensemble. La plus part des robots contient les éléments suivants :

1.2.1. Manipulateur : c'est le corps principal du robot qui comprend les jonctions, les articulations, et d'autres éléments de structure du robot. Il convient de noter ici que le manipulateur seul n'est pas un robot. [4]

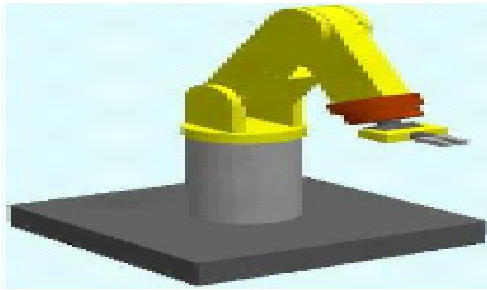


Figure 1.1 Un bras manipulateur

1.2.2. Effecteur finale : cette partie est reliée à la dernière jonction (main) d'un manipulateur qui gère généralement les objets, établit des connexions à d'autres machines ou effectue les tâches requises. [4]

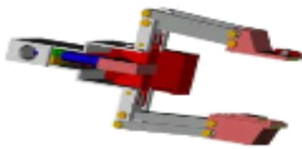


Figure 1.2 Effecteur finale d'un manipulateur

1.2.3. Actionneurs : les actionneurs sont les «muscles» de manipulateurs. Le contrôleur envoie des signaux aux actionneurs, qui, à son tour, déplacent les articulations du robot et des jonctions, les types communs des actionneurs sont les servomoteurs, les moteurs pas à pas, les actionneurs pneumatiques et les vérins hydrauliques. Les actionneurs sont sous le contrôle du contrôleur. [4]

Chapitre 1 : Généralités



Figure 1.3 Différents actionneurs d'un robot

1.2.4. Capteurs : les capteurs sont utilisés pour recueillir des informations sur l'état interne du robot ou pour communiquer avec l'environnement extérieur. Comme chez l'humain, le dispositif de commande de robot doit connaître l'emplacement de chaque lien du robot afin de connaître la configuration du robot. Toujours comme vos principaux sens de la vue, le toucher, l'ouïe, le goût, et la parole, les robots sont équipés de dispositifs sensoriels externes comme un système de vision, le toucher et les capteurs tactiles, synthétiseur de parole, et grâce à eux le robot peut communiquer avec le monde extérieur. [4]

1.2.5. Contrôleur : le contrôleur est plutôt proche de votre cerveau ; même si elle n'a pas la puissance du cerveau ; il contrôle toujours vos mouvements. Le contrôleur reçoit les données de l'ordinateur (le cerveau du système), commande les mouvements des actionneurs, et coordonne les mouvements avec les informations envoyées par les capteurs. [4]

1.2.6. Processeur : le processeur est le cerveau du robot. Il calcule les mouvements des articulations du robot, détermine combien et à quelle vitesse chaque joint doit se déplacer pour atteindre l'emplacement et la vitesse souhaitée, et supervise les actions coordonnées du contrôleur et les capteurs. Dans certains systèmes, le contrôleur et le processeur sont intégrés ensemble en une seule unité, et dans d'autres cas, ce sont des unités séparées. [4]



Figure 1.4 Un contrôleur utilisé pour commander un robot

Chapitre 1 : Généralités

1.2.7. Logiciel : trois groupes de logiciels sont utilisés dans un robot. L'un est le système d'exploitation qui exploite le processeur. Le second est le logiciel robotique qui calcule la motion nécessaire de chaque joint du robot basée sur des équations cinématiques. Ces informations sont envoyées au dispositif de commande. Ce logiciel peut être à différents niveaux, de la langue de la machine aux langues sophistiqués utilisés par les robots modernes. Les troisième groupes est la collection d'application - orientée les routines et les programmes développés pour utiliser le robot ou ses périphériques pour des tâches spécifiques telles que l'assemblage, le chargement de machines, la manutention et les routines de vision. [4]

1.3. Les type des robots

Il existe deux grandes familles de robots sont :

- Les robots manipulateurs.
- Les robots mobiles.

1.3.1 : les robots manipulateurs :

Un robot manipulateur est en forme d'un bras et se compose d'un certain nombre de segments qui est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement en toute sécurité. Les robots manipulateurs sont utilisés dans des applications industrielles pour s'effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage. [5]

1.3.2 Les robots mobiles : Un robot mobile est celui qui peut se déplacer dans son environnement de façon indépendante. Pour ce faire, le robot doit pouvoir naviguer, et la portée et la précision de navigation requise varie en fonction de la taille du robot et du type de sa tâche.

Chapitre 1 : Généralités

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

La structure mécanique et la motricité

Les organes de sécurité

Le système de traitement des informations et gestion des tâches.

Le système de localisation.

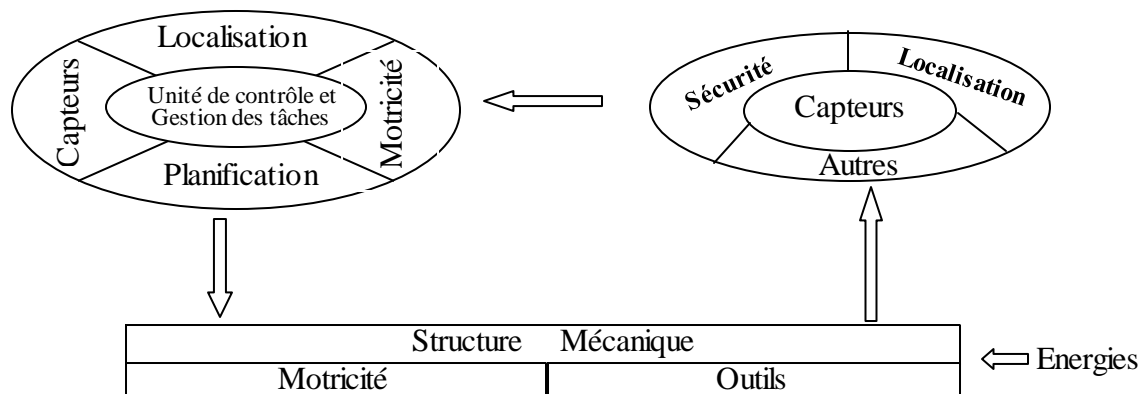


Figure 1.5 Architecture d'un robot mobile

1.3.2.2. Les robots à roues : Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées.

Nous citerons ici les quatre classes principales de robots à roues. [6]

a) **Robot unicycle :** Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

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

Chapitre 1 : Généralités

Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites. [6]

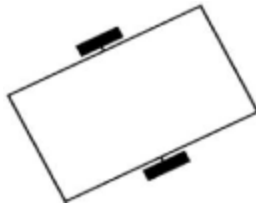


Figure1.6 Robot de type unicycle

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

C'est un robot non-holonyme. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable. [6]

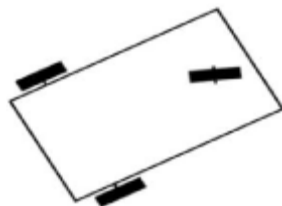


Figure1.7 Robot de type tricycle

c) **Robot voiture** : Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe. [6]

Chapitre 1 : Généralités

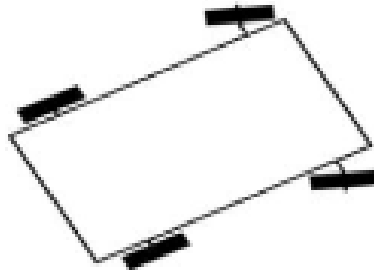


Figure1.8 Robot de type voiture

d) **Robot omnidirectionnel** : Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral. [6]

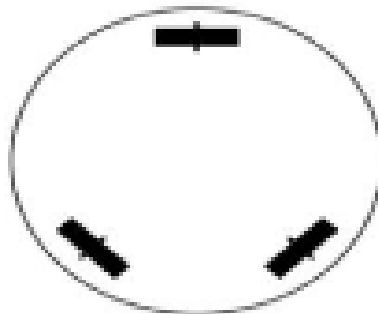


Figure1.9 Robot de type omnidirectionnel

Chapitre 1 : Généralités

1.4 Comparaison des différents types des robots : Nous pouvons observer dans le tableau ci-dessous un récapitulatif des avantages et des inconvénients des différents types de robots à roues :

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

Tableau 1.1 Les avantages et des inconvénients des différents types de robots à roues

1.5 Conclusion : ce chapitre nous permet d'évoquer quelques généralités sur les robots, moyens de perceptions et d'exécutions, classes des robots mobile ainsi que les avantages et les inconvénients de chaque structure.

Le chapitre prochain en va porter sur l'étude du coté mécanique du robot et toutes les parties.

Chapitre 2 : Conception mécanique

2.1 Introduction :

Après avoir vu les composants et les différents types des robots, on entame maintenant la réalisation mécanique du robot. Dans ce chapitre nous allons détailler chaque partie mécanique du robot.

2.2 Structure :

La structure du robot est entièrement composée d'une plateforme en matière d'aluminium. Elle supporte tous les composants du robot.

Le robot possède un corps de hauteur 40 cm et de largeur 30 cm et longueur 25 cm avec trois roues, 2 roues en arrière et 1 roue en avant.

Le système de perception est basé sur une caméra qui détecte les objets, un capteur infrarouge pour l'évitement des obstacles, et 2 encodeurs optiques intégrés dans les moteurs pour la mesure de vitesse, et une boussole qui assure la localisation du robot.

Le système d'exécution compte sur 3 servomoteurs pour tirer la balle, 2 moteurs à courant continu.

2.2.2 Le châssis :

Il porte tous les éléments constituant les systèmes mécaniques et électriques du robot à savoir les moteurs, les roues, la carte de puissance

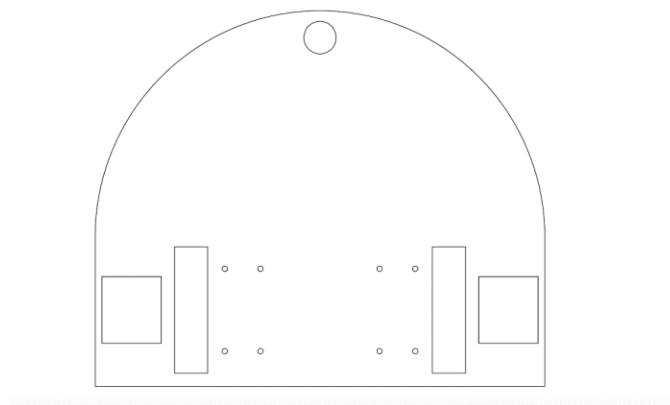


Figure 2.1 schéma du châssis

Chapitre 2 : Conception mécanique

2.2.4 Les roues : Le robot possède trois roues, deux en arrière sont identiques, elles ont été fabriquées en téflon couvertes d'une couche procurant une certaine adhérence et facilitant le déplacement et pour éviter le glissement. Et l'autre en avant pour réaliser l'équilibre du robot mobile.

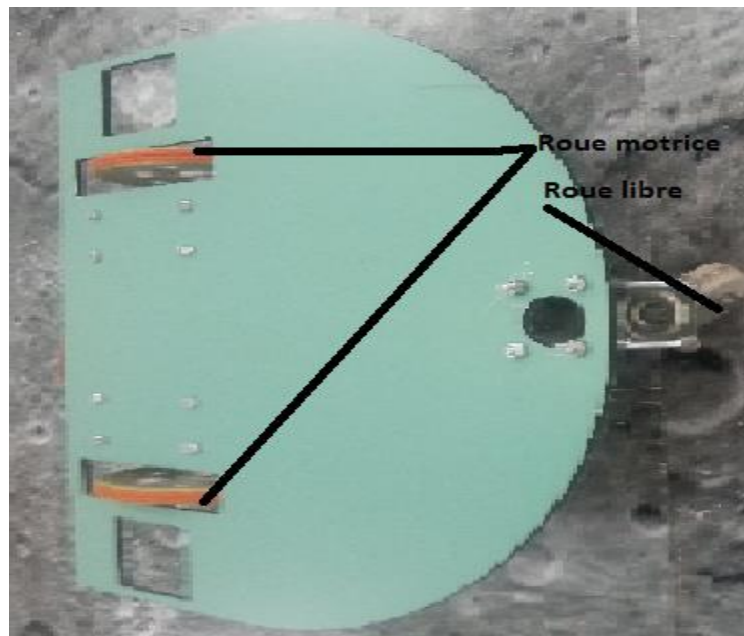


Figure 2.2 image en position verticale de trois roues du robot.

Les roues de l'arrière de diamètre 90mm et 14 mm de largeur, sont reliées au moteur par une bague d'accouplement.

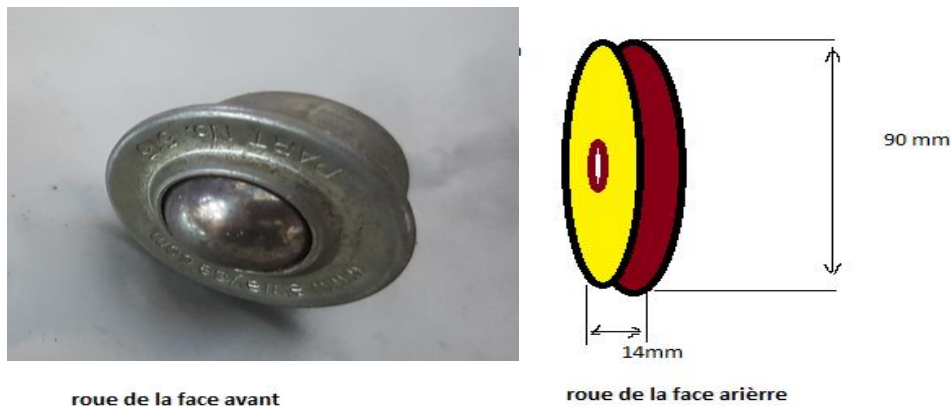


Figure 2.3 les deux types de roues de robot.

2.3. Les actionneurs :

Dans notre projet on a utilisé 2 types des actionneurs :

- Moteurs à courant continu.
- Les servomoteurs.

2.3.1. Moteurs CC (EMG 30) :

L'EMG30 est un motoréducteur 12 Vcc équipé d'un encodeur. Il convient parfaitement aux applications de robotique et est équipé d'un filtre de suppression de bruit. Il est utilisé dans le kit de propulsion RD02.

Alimentation: 12 Vcc

Consommation: 530 mA (150 mA à vide)

Courant de blocage: 2,5 A

Vitesse de rotation: 170 t/min (216 t/min à vide)

Couple: 1,5 kg.cm

Encodeur: 360 impulsions/tour

Dimensions: $\varnothing 28,5 \times 86,6$ mm (axe de 10 mm inclus) [7]

Chapitre 2 : Conception mécanique



Figure2.4 Moteur à CC (EMG 30)

Les moteurs doivent être placés de manière précise dans le châssis pour assurer la mobilité (évités les handicaps du robot).

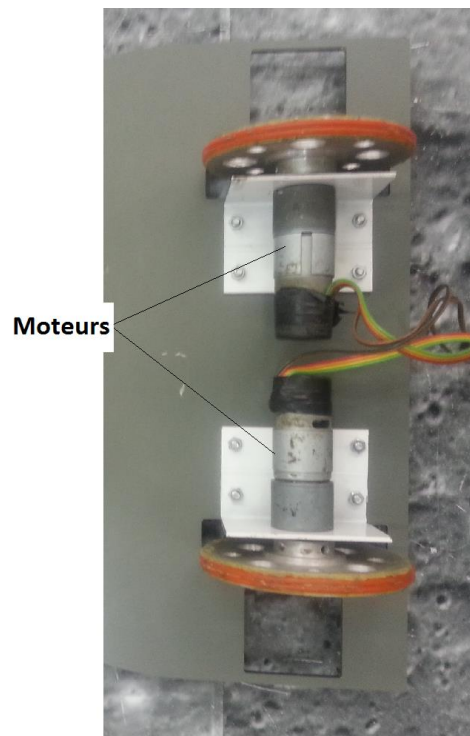


Figure2.5 moteurs placés dans le châssis

2.2.2 Servomoteur :

Les servomoteurs proposent plus d'intelligence et de fonctionnalités qu'un moteur simple, c'est pourquoi ils sont beaucoup utilisés en robotique. Un servomoteur est constitué classiquement d'un moteur. Un servomoteur peut atteindre et maintenir une position définie.

Le système de frappe d'la balle se fait à base des servomoteurs identiques avec un bras chacun. Ils sont fixés horizontalement en avant du châssis. Les servomoteurs à CC utilisent une alimentation de 1.8/6v.

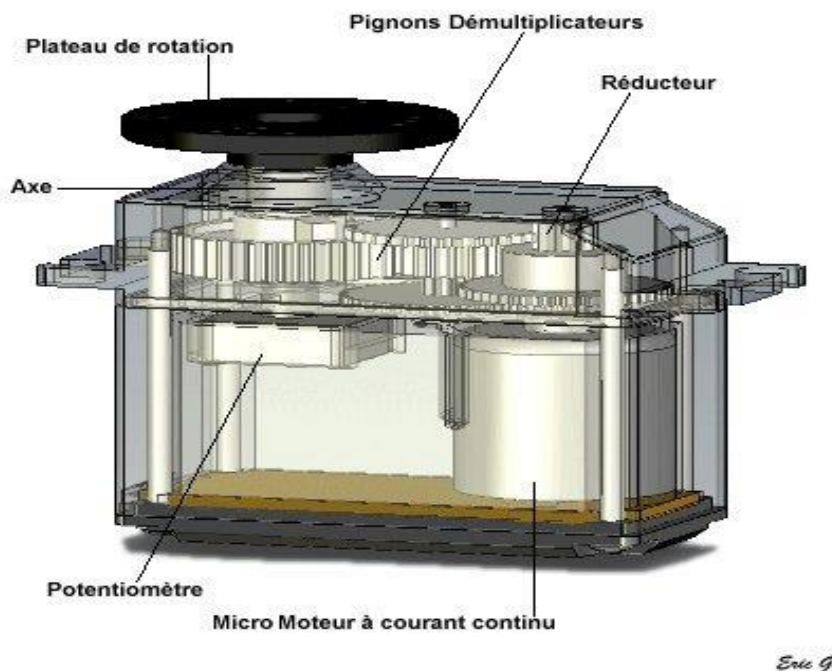


Figure2.6 : servomoteur à CC

2.5. Fabrication :

Étape 1 :

Nous avons commencé par l'assemblage sur le châssis, les moteurs et les servomoteurs que nous avons dû préalablement câbler. Dès lors, nous avons différencié les moteurs en nomment M1 pour le moteur gauche et M2 pour le moteur droite, SVR1 pour le servomoteur gauche et SVR2 pour le servomoteur droite et SVRM pour le servomoteur qui est au milieu par rapport à l'avant de la plateforme. Les moteurs y sont boulonnés.

Étape 2 :

Nous avons vissé sur le châssis la roue libre et aux axes des moteurs les roues motrices. Cela permet d'apporter la notion de mobilité au robot.

2.6 Conclusion :

Dans ce chapitre nous nous sommes intéressés à la partie structure du robot, nous avons présenté l'emplacement des actionneurs du robot mobile ainsi que le rôle que joue chacun pour le fonctionnement.

Dans le prochain chapitre nous présentons la partie électronique de notre robot, avec toutes les cartes réalisées.

Chapitre3 : Conception électrique

3.1 Introduction :

Dans la conception d'un robot, la partie électronique joue un rôle très important dans le fonctionnement du système. En effet pour permettre au robot de se déplacer et de communiquer avec l'extérieur il est nécessaire de réaliser des cartes telles que la carte de commande arduino Mega. La carte d'isolation galvanique et la carte de puissance ainsi que les différents capteurs et la camera. La structure électronique de notre robot mobile peut être représentée par le schéma ci-dessous :

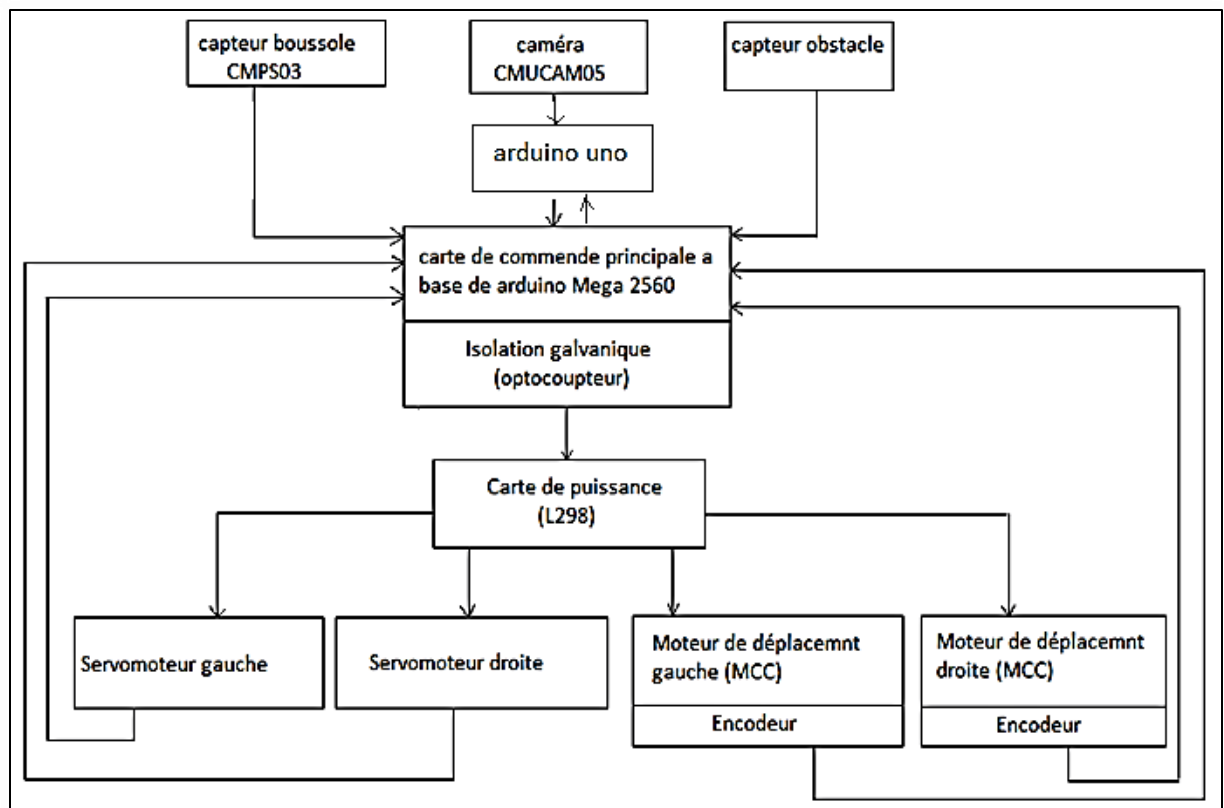


Figure 3.1 schéma synoptique général du robot

Chapitre3 : Conception électrique

3.2 La carte mère :

La carte mère permet le contrôle et la gestion de toutes les autres cartes. C'est-à-dire elle récupère les informations des capteurs et la caméra puis elle effectue leur traitement, en fonction de ces informations, elle délivre les commandes adéquates aux autres cartes pilotant les différents moteurs à courant continu et les servomoteurs.

3.3 Les cartes de commande :

La partie électronique dispose deux types de carte de commande, l'une la carte arduino Mega et la carte arduino Uno.

L'arduino Uno est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur ATmega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C.

Dans ce projet la carte arduino Uno est utilisée comme un composant intermédiaire entre la caméra CMUCAM5 et la carte de commande arduino Mega, elle reçoit les informations traitées par la caméra et l'envoient à la carte arduino Mega.

3.3.1 Carte de commande arduino Mega 2560 :

Le système d'information centrale est basé sur le microcontrôleur ATmega 2560. La carte arduino Mega permet le contrôle et la gestion de toutes les mécaniques

(Moteurs, servomoteurs, capteurs...).

3.3.2 Caractéristique de l'arduino Mega 2560 :

- alimentation : via port USB 7 à 12 V sur connecteur alim
- microprocesseur : ATmega2560
- mémoire flash : 256 KB
- mémoire SRAM : 8 KB
- mémoire EEPROM : 4 KB

Chapitre3 : Conception électrique

- 54 broches d'E/S dont 14 PWM
- 16 entrées analogiques 10 bits
- intensité par E/S : 40 mA
- cadencement : 16 MHz
- 3 ports série
- bus I2C et SPI
- gestion des interruptions
- fiche USB B
- dimensions : 107 x 53 x 15 mm [8]

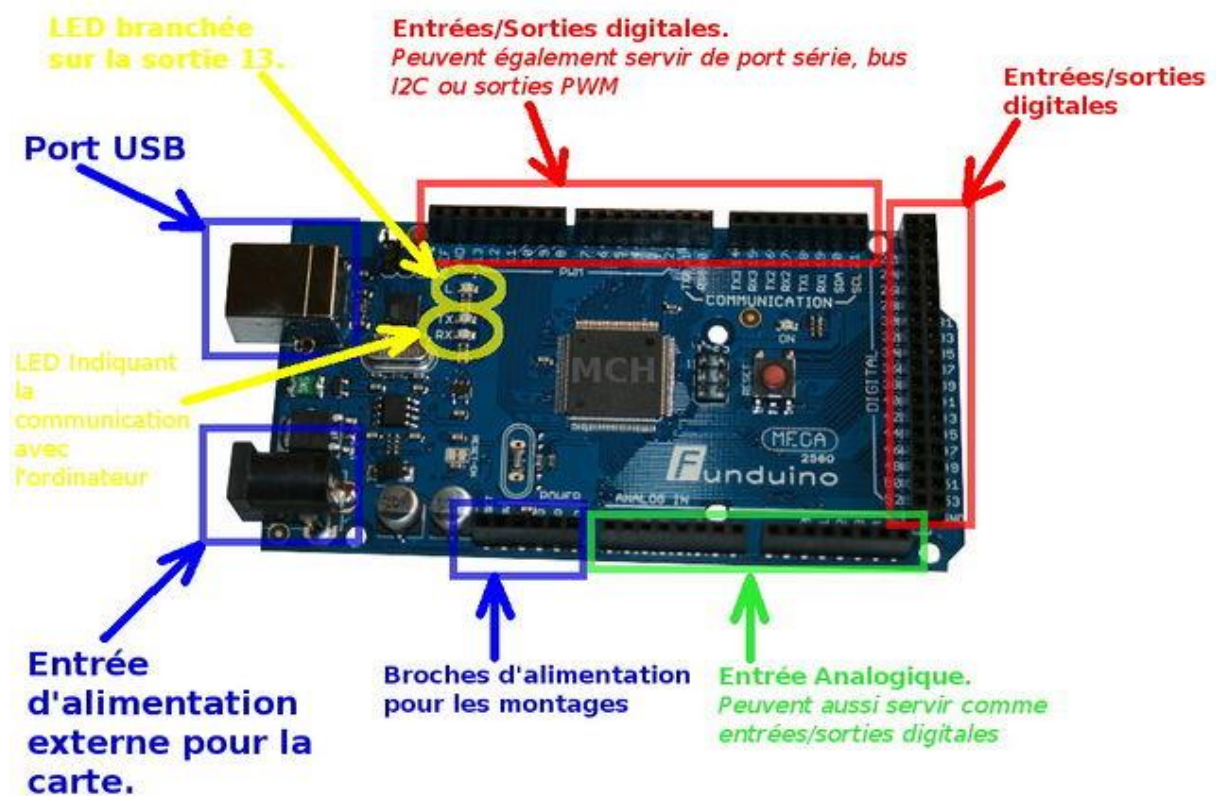


Figure3.2 arduino Mega2560

Chapitre3 : Conception électrique

3.3.3 Entrées et sortie numérique :

3.3.3.1 présentations :

54 entrées / sorties numériques , repérées de 0 à 53; chacune d'entre elles pouvant fonctionner en entrée ou en sortie sous le contrôle du programme et ce sens pouvant même changer de manière dynamique pendant le fonctionnement du programme. Ces entrées / sorties admettent et délivrent des signaux logiques compatible (tension comprise entre 0 et 5V). [8]

3.3.3.2 entrées série asynchrones :

4 entrées (Rx0,Rx1,Rx2,Rx3) et 4 sorties (Tx0,Tx1,Tx2,Tx3) séries asynchrones, partagées respectivement avec les lignes 0,19,17,15 pour la partie Rxn 1,18,16,14 pour ce qui est des Txn.[8]

a) Interruptions externes :

6 entrées d'interruption externes partagées avec les lignes 2, 3, 18, 19, 20,21.

b) Les sorties PMW :

14 sorties PWM (La modulation de la largeur d'impulsion (MLI ou PWM), est une technique pour 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 BAS, 0V et 5V) avec les lignes de 0 à 13. [8]

c) Bus série normalisée SPI :

4 entrées/ sorties pour le bus série normalise SPI (SS -> 53 / MOSI ->51 / MISO -> 50 / SCK -> 52).

Chapitre3 : Conception électrique

L'interface Série pour périphériques (SPI) est un protocole de communication série synchrone utilisé par les microcontrôleurs pour communiquer avec un ou plusieurs composants périphériques rapidement sur de courtes distances. Ce protocole peut aussi être utilisé pour des communications entre deux microcontrôleurs. [8]

d) Entrée/Sortie pour l'interface série I2C :

2 entrées/ sorties d'interface série I2C réparties comme suit SDA -> 20, SCL ->21
I2c est un bus de données qui permet de relier facilement un microprocesseur et différents circuits. [8]

3.3.3.3 Entrées analogiques :

La carte dispose pour cela de 16 entrées, repérées d'A0 à A15, qui peuvent admettre toute tension analogique comprise entre 0 et 5 V. Mais la tension ne doit jamais dépasser 5 Volts sous peine de détruire le microcontrôleur. Pour mesurer des tensions supérieures à 5 Volts il faut équiper l'entrée d'un pont diviseur. Pour mesurer des tensions analogiques les cartes Arduino contiennent un convertisseur analogique-numérique. Ce convertisseur a une résolution de 10 bits, renvoyant des résultats de mesure sous forme d'entiers de 0 à 1023. [8]

3.3.3.4 Fonctionnement et utilisation de l'arduino :

L'utilisation des cartes arduino est très simple : connectez la carte à l'ordinateur (via le câble USB, en général), lancez « Arduino IDE » (c'est le logiciel qui permet de programmer la carte), mettez en place vos composants et reliez-les à la carte, puis codez votre programme. Enfin, chargez-le sur la carte grâce au bouton téléverser. Votre programme s'exécutera ensuite en boucle. [9]

Chapitre3 : Conception électrique

3.4 Carte d'isolation galvanique :

Lorsque les moteurs démarrent ou s'arrêtent, d'importants pics de courant peuvent se produire. Ceux-ci pourraient être destructeurs pour l'électronique de commande. Nous utiliserons donc des optocoupleurs afin d'avoir une isolation électrique parfaite entre les deux parties (commande et puissance).

3.5 carte de puissance :

La carte arduino délivre des signaux qui ne pourront pas mettre en marche les moteurs, c'est pourquoi on utilise une carte de puissance qui réalise cette fonction. La carte de puissance est une interface de puissance entre les signaux venant de la carte de commande et le fonctionnement des moteurs.

3.5.1 Le but de carte de puissance :

Cette carte a deux fonctions principales. Tout d'abord, elle doit fournir aux autres cartes électroniques, leur tension d'alimentation (0/5V stabilisé). Ensuite elle doit alimenter les moteurs (12V, 2A) qui permettent au robot de se déplacer. Elle reçoit donc des signaux de commande via la carte de protection du port parallèle. [10]

Notre carte de puissance est composée de deux ponts en H, ce dernier est constitué de quatre transistors qui travaillent comme des interrupteurs et permettent d'obtenir quatre états selon la façon dont ils sont commandés, ce principe est représenté par la figure fonctionnement d'un pont en H.

Chapitre3 : Conception électrique

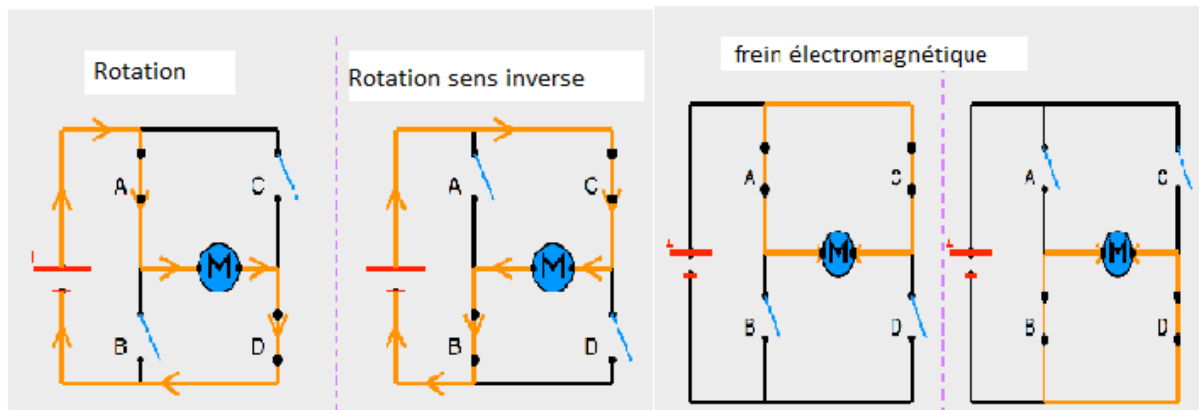


Figure 3.3 fonctionnement d'un pont en H

La roue est libre : tous les interrupteurs sont commandés ouverts, aucune tension n'est appliquée au moteur la tension de la batterie.

La marche avant : deux interrupteurs sont commandés fermés et deux autres ouverts, on applique au moteur la tension de la batterie.

La marche arrière : deux interrupteurs sont commandés fermés et deux autres ouverts, mais, à l'inverse du cas précédent, on applique au moteur la tension de la batterie, avec une polarité inverse.

Le frein : deux interrupteurs de même polarité du pont en H sont commandés fermés, les pôles du moteur sont court-circuités, ce qui freine le moteur. Cet état est rarement utilisé.

Un pont en H est donc simplement un interrupteur qui relie la batterie au moteur. Cet interrupteur est généralement commandé par deux entrées logiques : enb1 et enb2. La figure 3.3 donne le schéma du pont en H.

Chapitre3 : Conception électrique

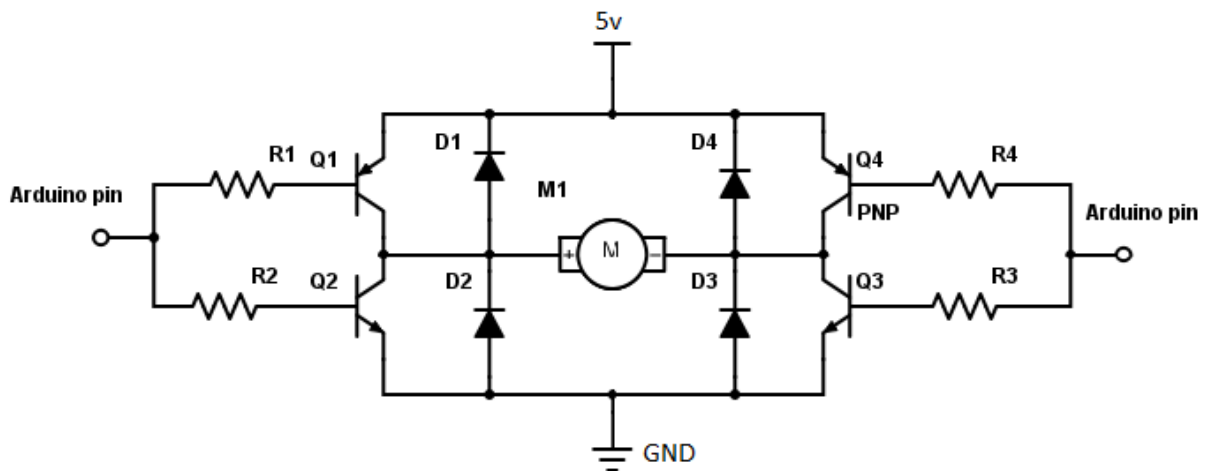


Figure 3.4 le pont en H

Enb1	Enb2	Moteur
0	0	Arrêt
0	1	Sens1
1	0	Sens2
1	1	Arrêt

Tableau3.1 logique décrivant le fonctionnement du pont en H

Les ports en H sont habituellement réalisés à partir de transistors de puissance normaux et les diodes rapides de protection (pour protéger les transistors des surtensions produits par le moteur).

3.5.2 Le circuit intégré L298N :

Cette carte de puissance utilise un driver L298, qui intègre un double pont en H. la carte est l'application direct du datasheet : seulement le L298, les condensateurs de filtrage, les diodes rapide de protection, et enfin le connecteur pour le relier à la carte principale, le connecteur d'alimentation.

Chapitre3 : Conception électrique

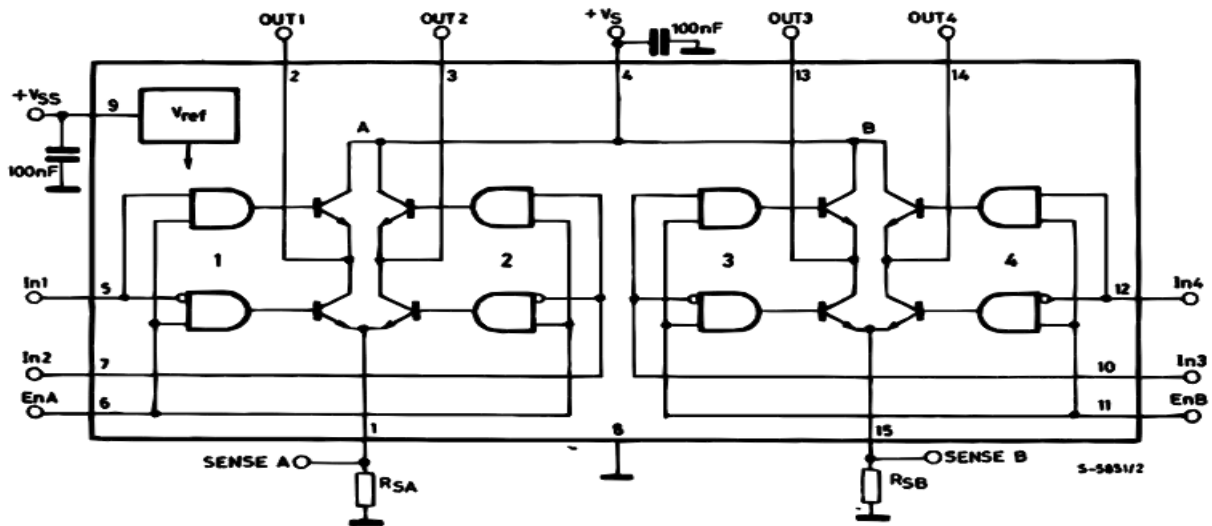


Figure 3.5 schéma interne du L298

Les caractéristiques du L298 sont les suivantes :

- Intensité maximale : 2A par pont ;
- Alimentation de puissance de 5.5V à 50V ;
- Trois entrées par pont : In1, In2, ENABLE
- Description puissance : 25W ;
- Type de boîtier : multiwatt15 ;

Chapitre3 : Conception électrique

Le fonctionnement est comme suit :

Entrées		Sortie	Fonction
ENB=1	In1=1 ; In2=0	Out1=1 ; Out2=0	La marche avant
	In1=0 ; In2=1	Out1=0 ; Out2=1	La marche arrière
	In1=In2	Out1=1 ; Out2=1 ou Out1=1 ; Out2=0	Le frein
ENB=0	In1=X ; In2=X	Rien	Roue libre

Tableau 3.2 les entrées/sorties du pont en H

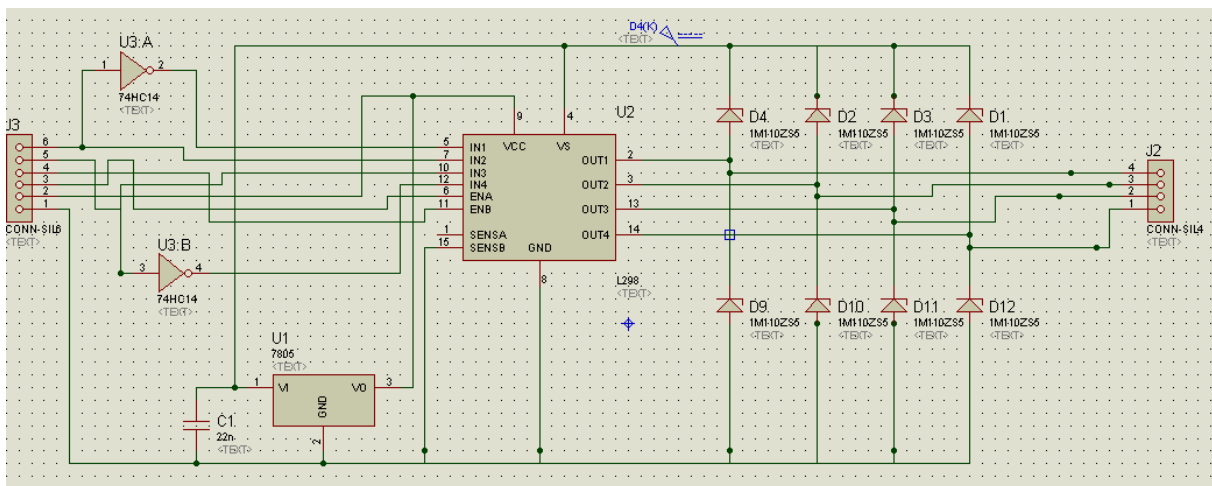


Figure 3.6 schéma développé de la carte de puissance

3.5.3 Régulation de la vitesse :

Le pilotage des moteurs se fait par MLI (Modulation de Largeur d'Impulsion) ou PWM (Pulse Width Modulation).

Chapitre3 : Conception électrique

Le PWM est un signal dont la période est fixe, mais le rapport cyclique varie. En d'autres termes, T_h et T_b varient tout en conservant $T_h + T_b = T_c = \text{constante}$.

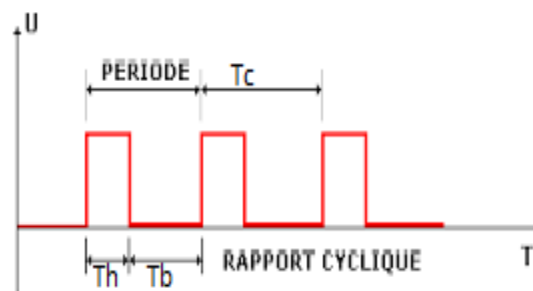


Figure 3.7 Exemple de signaux PWM

La tension moyenne appliquée au moteur est proportionnelle au rapport cyclique selon la formule :

$$V_{\text{moy}} = (T_h * U) / T_c$$

L'association d'un pont en H et du pilotage par MLI permet de piloter la vitesse et le sens de rotation de chaque roue indépendamment.

3.6 Capteur infrarouge :

Un capteur infrarouge donne à un robot programmable la capacité de détecter des obstacles et de mesurer la distance les séparant.

Chapitre3 : Conception électrique

Le capteur infrarouge est constitué d'un récepteur qui détecte l'intensité lumineuse dans la gamme de lumières infrarouges, supérieures à 780 nanomètres (nm) et d'un émetteur de lumière infrarouge. [13]

Le capteur infrarouge peut être utilisé :

- comme capteur de contact en prenant une mesure avec la LED infrarouge éteinte (récepteur) et une avec la LED infrarouge allumée (émetteur) : s'il n'y a aucun obstacle proche, la valeur lue sera la même. Sinon, l'obstacle aura réfléchi la lumière infrarouge et la deuxième mesure donnera un résultat plus élevé.
- pour suivre une ligne noire sur fond blanc (ou l'inverse) : en effet, le noir réfléchit beaucoup moins la lumière infrarouge que le blanc.
- comme capteur de distance en mesurant l'angle avec lequel le rayon réfléchi arrive sur le récepteur : En fonction de la distance entre l'émetteur et le récepteur, on peut en déduire la distance de l'obstacle.

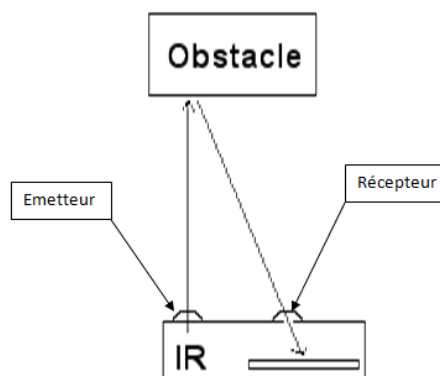


Figure 3.8 un capteur infrarouge

Chapitre3 : Conception électrique

3.7 Codeur à effet HALL :

Les détecteurs à effet Hall peuvent être utilisés comme capteur de position et/ou de vitesse angulaire, ils permettent aussi de connaître le sens de rotation. Le codeur utilise un disque magnétique multipolaire à faible inertie et deux sondes à effet Hall pour générer deux signaux de sortie déphasés de 90°. [14]

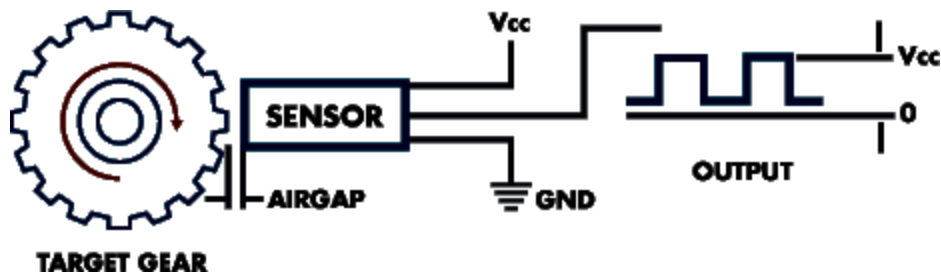


Figure 3.9 fonctionnement d'un codeur à effet HALL

3.8 Le Capteur boussole (CMPS03) :

Ce petit module est capable de détecter le "nord" grâce à l'emploi de 2 capteurs spécialisés montés en angle à 90° et par déduction de vous indiquer son orientation par le biais d'un échange d'information via un bus I2C ou d'un signal "PWM". [10]

Le signal PWM fournit une impulsion comprise entre 1 ms (0°) et 36,99 ms (359,9°) avec une résolution d'environ 100 us / °).

Idéalement conçue pour les applications liées à la robotique ludique cette boussole pourra par exemple être très facilement interfacée avec la plupart des microcontrôleurs.

Chapitre3 : Conception électrique

Pour une plus grande facilité, le bus "I2C" pourra également être utilisé pour récupérer les informations sous forme d'un mot binaire compris entre 0 et 255 ou 0 et 3599 (correspondant ici à 0 et 359,9 °). [10]

Pour ce faire, il suffira d'adresser le module à la manière d'une mémoire série qui vous "répondra" en vous renvoyant la valeur mesurée. Afin d'augmenter la précision de la mesure, le module dispose également d'une procédure de calibrage et d'une intégration de l'influence des perturbations liées aux réseaux 50/60Hz. [10]

3.8.1 Caractéristiques techniques :

- Alimentation : + 5 Vcc.
- Consommation : 20 mA.
- Résolution : 0.1 °.
- Précision : 3 à 4 ° env.
- Dimensions : 32 x 35 mm [10]

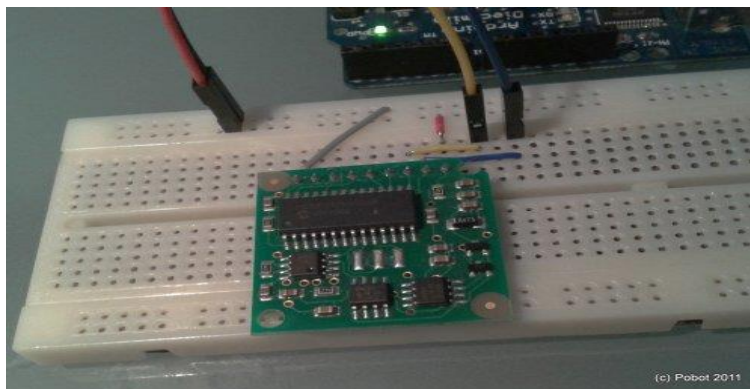


Figure3.10 capteur boussole connecté avec l'arduino

Chapitre3 : Conception électrique

3.8.2 Connexion :

Pin 1 : Alimentation 5V - Consommation 25 m

Pin 2/3 :I2C interface, doivent être pull up au 5V (ex : 47 k) si non utilisé

Pin 4 : PWM Signal représentatif de l'angle, Etat haut varie de 1ms (0°) a 36,99 ms (359.9 °), formule $(100\mu\text{s} * \text{angle}) + 1\text{ms}$, état bas : 65 ms, précision du signal 10 μs

Pin 5 Indicateur de l'état de calibration

Pin 6 : Calibration de la boussole, peut se faire aussi par l'écriture de 0xFF dans le registre de configuration. Active à l'état bas

Pin 7/ 8 Non utilisé

Pin 9 : Masse (0V) [11]

3.9 Module de reconnaissance vidéo CMUCAM05 :

Dans plusieurs applications de robots mobiles, le robot doit pouvoir interagir avec l'environnement tout en s'adaptant à la situation extérieure. Prévion et estimation du mouvement des objets sont la base des tâches d'interaction.

Par exemple, il est très essentiel de détecter et de prédire la position de la balle lorsqu'un robot joue au football. Le mode de suivi de la balle est en détectant sa couleur. Cela peut se faire en utilisant un système de vision approprié.

Un capteur d'image peut détecter pratiquement n'importe quoi par en appliquant le bon algorithme.

La CMUcam5 est un capteur de vision rapide à faible coût. Il peut être connecté directement à l'arduino et à d'autres contrôleurs. Un enseignement rapide peut être effectué dans ce capteur pour détecter des objets.

CMUcam5Pixy traite des problèmes de capteurs de vision en combinant le capteur d'image avec un processeur puissant.

Chapitre3 : Conception électrique

Après avoir traité les images par CMUcam5 Pixy, les seules informations utiles seront envoyées au microcontrôleur, Il le fait très rapidement (fréquence d'image 50 Hz). Une interface, telle qu'UART série, SPI, I2C, sortie numérique ou sortie analogique peut être utilisée pour transmettre cette information. Donc, un bon pouvoir de traitement sera disponible dans Arduino ou autre microcontrôleur lorsqu'il se connecte au CMUcam5 Pixy. Il peut détecter des centaines d'objets avec plusieurs couleurs au même temps. [12]

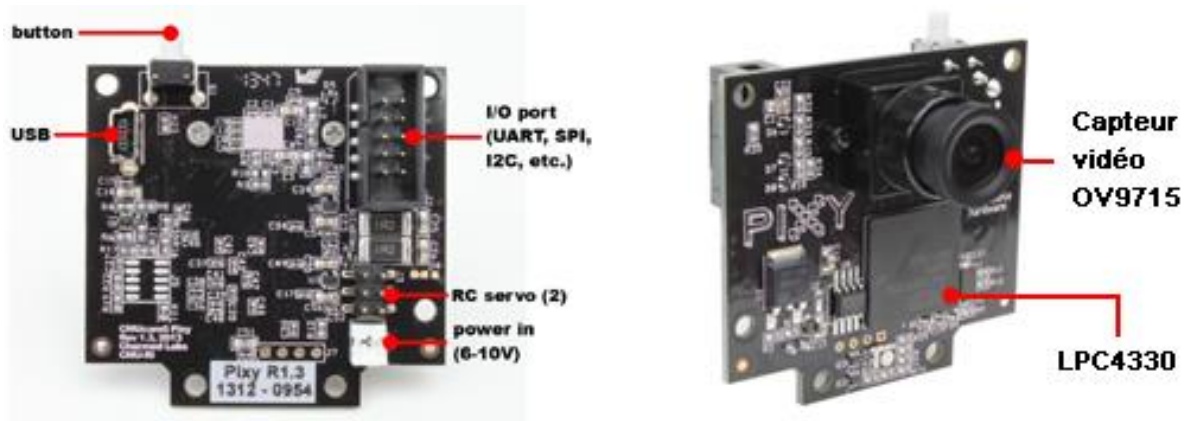


Figure 3.11 le module CMUCAM5

3.9.1Caractéristiques :

- Processeur : NXP LPC4330 - 204 MHz - dual core
- RAM : 264 K bytes
- Flash : 1Mbytes
- Capteur vidéo : Omni vision OV9715 - 1/4" - 1280 x 800 pixels
- Champ de vision de la lentille : 75° horizontal, 47° vertical
- Type de lentille : M12
- Consommation : 140 mA.
- Alimentation : Via USB (5 V - Câble non livré) ou entrée (6 V à 9 V)

Chapitre3 : Conception électrique

- Sorties : UART serial, SPI, I2C, USB, digital, analogue
- Dimensions : 5,3 x 5,1 x 3,6 cm
- Poids : 27 gr [12]

3.9.2 Principe de fonctionnement :

La CMUcam5 Pixy utilise un algorithme de "filtrage de couleur" sur la base de "teintes" pour détecter des objets. La plupart d'entre nous sont familiers avec les notions RVB (rouge, vert et bleu) pour représenter les couleurs. La CMUcam5 Pixy calcule la teinte et la saturation de chaque pixel RGB à partir du capteur vidéo et les utilise en tant que paramètres de filtrage primaire. Dans la majorité des cas, la teinte d'un objet reste inchangée suite à des changements d'éclairage et d'exposition, ce type de phénomène peut toutefois jouer en défaveur de l'algorithme de filtrage. Pour sa part la CMUcam5 Pixy est capable de gérer cette situation avec davantage d'efficacité que les versions précédentes de CMUCAM. [12]

3.9.3 Distances d'objet et d'image :

La lumière peut passer à travers du vide ou des matériaux clairs, mais grâce à un vide, sa vitesse sera plus rapide. La direction d'un rayon lumineux change lorsqu'elle frappe la surface d'interface entre deux supports sous un angle non orthogonal. Ce pli de rayon de lumière est appelé réfraction. Une lentille est un dispositif optique (un morceau de verre) qui réfracte et transmet la lumière, diverge ou converge le faisceau. Une section d'une surface sphérique de taille différente d'un côté et une section de surface sphérique sur l'autre façonnent l'objectif. Ces sections provoquent l'étalement de la lumière soit après que les faisceaux lumineux convergent vers une zone ou immédiatement à partir du verre. Peut être défini à partir des différentes tailles des sphères. CMUcam5 Pixy utilise le type de

Chapitre3 : Conception électrique

lentille M12. La figure 2 illustre l'analyse des lentilles

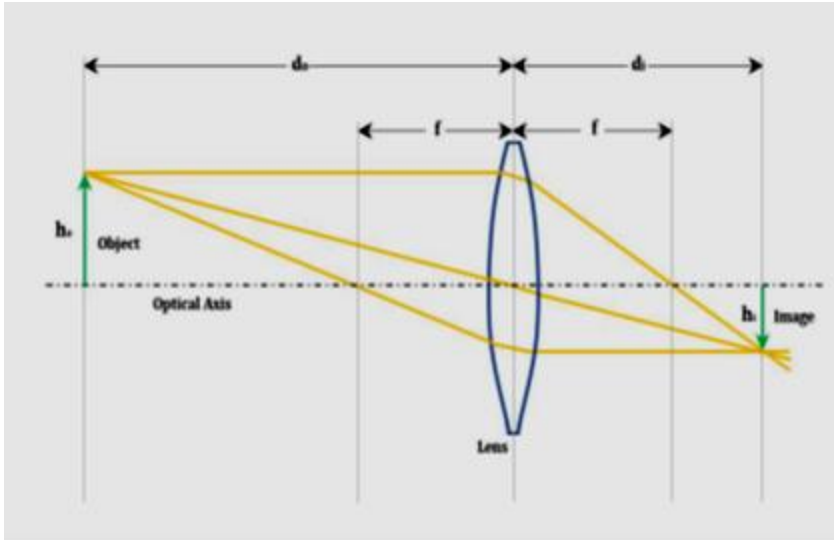


Figure 2 : Analyse de la lentille qui montre la relation de l'objet et de l'image

Où :

d_o : est la distance entre le centre de l'objectif et l'objet (mesuré le long de l'axe).

d_i : est la distance entre le centre de l'objectif et l'image (mesurée le long de l'axe).

F : est la distance focale de l'objectif.

h_o : est la taille de l'objet.

h_i : est la taille de l'image.

La relation quantitative entre la distance de l'image (d_i), la distance de l'objet (d_o) et la distance focale (f) peut être exprimée par l'équation de la lentille. L'équation est indiquée comme suit :

$$\frac{1}{F} = \frac{1}{d_o} + \frac{1}{d_i} \dots \dots \dots (1)$$

L'équation d'agrandissement rapporte le rapport entre hauteur d'image et hauteur d'objet au rapport de distance d'image et distance de l'objet. L'équation d'agrandissement est indiquée comme suit :

Chapitre3 : Conception électrique

$$M = \frac{h_i}{h_o} = \frac{d_i}{d_o} \dots\dots\dots(2)$$

L'équation (1) et l'équation (2) peuvent être combinées pour donner des informations sur la distance de l'objet si l'image

La distance, la hauteur de l'objet et la distance focale sont connues.

L'équation (2) peut être réécrite comme suit :

$$d_i = \frac{h_i \times d_o}{h_o} \dots\dots\dots (3)$$

Sous-équation (3) dans l'équation (1) :

$$\frac{1}{F} = \frac{1}{d_o} + \frac{h_o}{h_i \times d_o}$$

Réorganisez ceci pour donner l'objet à distance comme suit :

$$d_o = F \left(1 + \frac{h_o}{h_i} \right) \dots\dots\dots (4)$$

Chapitre3 : Conception électrique

3.9.4 Traitement d'image de la caméra :

La caméra utilise un algorithme de la conversion {R, V, B} ==> {T, S, L}

3.9.4.1 Conversion RVB TSL :

a) **détection de contour** : la détection de contour se fait manuellement par l'application PixyMoon en sélectionnant l'objet voulu dans l'image

20	20	30	255	60	87	66	90	21	12
20	20	32	94	10	10	65	23	124	100
65	25	100	98	10	10	00	12	140	155
33	50	54	94	200	200	200	40	200	200
66	25	67	96	200	200	200	32	30	20
54	47	91	86	200	200	200	19	30	20
74	50	46	68	200	200	200	76	60	80
98	32	31	38	60	30	98	38	65	65
65	45	28	37	97	30	59	68	94	58
36	98	96	34	36	30	35	91	97	96

Plan rouge

b) **calcul de la teinte** :

Généralement la teinte est calculée par les équations suivantes :

$$T = \frac{V-B}{M-m} + 0 ; \text{si } M = R$$

$$T = \frac{B-R}{M-m} + 2 ; \text{si } M = V$$

$$T = \frac{R-V}{M-m} + 4 ; \text{si } M = B$$

Avec : $M = \max \{R, V, B\}$ et $m = \min \{R, V, B\}$

T : s'exprime en degré ==> $T^\circ = T * 60$

Si $T^\circ < 0$ ==> $T^\circ = 360 + T$

30	32	31	38	60	30	98	21	66	10
32	32	30	20	10	55	90	124	66	30
100	19	30	20	10	95	58	140	80	30
54	76	60	80	30	30	30	200	80	30
67	38	65	65	30	30	30	30	200	200
91	68	94	58	30	30	30	30	100	100
46	91	97	96	30	30	30	60	100	100
31	95	16	90	90	90	90	65	20	25
28	75	14	20	10	55	90	94	10	25
96	30	20	22	200	120	100	97	10	25

Plan vert

Plan bleu

50	100	150	10	95	58	99	10	1	9
50	50	200	25	70	70	80	10	0	9
50	30	20	22	200	120	100	65	0	9
50	30	30	70	80	80	80	30	0	8
50	40	65	60	80	80	80	35	0	8
50	40	13	65	80	80	80	40	0	8
50	70	12	40	80	80	80	45	0	8
50	75	14	20	10	55	90	75	0	8
50	95	16	90	90	90	90	70	0	4
50	96	97	98	19	110	100	90	5	8

Tableau 3.3 : les plants RVB

Chapitre3 : Conception électrique

c) calcule de la luminance :

$$L = \frac{m+M}{2*255}$$

d) calcule de saturation

$$S = \frac{M - m}{2L} * \left(\frac{1}{255}\right) \text{ si } L \leq \frac{1}{2}$$

$$S = \frac{(M - m)}{2 - 2L} * \left(\frac{1}{255}\right) \text{ si } L > \frac{1}{2}$$

$$0 \leq S \leq 1$$

Dans notre projet la saturation de l'objet doit être élevée pour éviter les erreurs entre les couleurs sortant à travers les autres objets.

4.9.4.2 : Binarisation de l'image de la teinte :

255 si $T = \text{seuil}$

0 ailleurs donc en obtient

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Tableau3.4 plant de la teinte

L'application PixyMoon affiche le contour de l'objet avec couleur blanche et suit son déplacement dans l'espace de la scène capturée.

Chapitre3 : Conception électrique

3.10 Fabrication :

Étape1 :

Nous installons la caméra au-dessous de la réglette.

Étape2 :

Nous installons la batterie l'alimentation entre les deux roues

Étape 3 :

Nous avons installé la carte arduino

Étape 4 :

Nous connectons l'arduino Mega et la carte de puissance par un emboîtement des broches dans les pins. Cela permet la connexion et la communication entre les deux composants. Les câbles des moteurs et servomoteurs sont branchés à la carte de puissance..

Étape 5 :

Nous connectons l'arduino uno avec la caméra CMUCAM5 et la carte arduino Mega

Étape 6 :

Nous avons fixé le capteur infrarouge en haut de la réglette dont le but d'augmentation du champ de détection des obstacles.

Étape 7 :

Nous installons sur le haut de la réglette la boussole et le bouton poussoir pour la mise en marche du robot.

Chapitre3 : Conception électrique

3.11 Conclusion :

Dans ce chapitre nous sommes intéressées à la partie électronique du robot mobile, nous avons présenté les différents capteurs utilisés, leurs principes de fonctionnement ainsi que le rôle que joue chacun pour le bon fonctionnement du robot mobile. La gestion des différents systèmes du robot mobile se fait le module de la carte de commande arduino Mega 2560, la commande des moteurs et servomoteurs se fait par la carte de puissance, la protection de la carte de commande se fait par l'isolation galvanique, la localisation du robot se fait par le capteur boussole CMPS03, le traitement d'image se fait par la caméra CMUCAM05.

Dans le chapitre prochain en va porter sur la partie programmation et réalisation du robot.

Chapitre 4 : Partie programmation et gestion des capteurs

4.1 Introduction :

Dans les chapitres précédents nous avons présenté la partie mécanique et électronique du robot, il est clair que la partie programmation est très importante pour la coordination du fonctionnement de toutes les parties du robot mobile.

4.2 Programmation du l'arduino :

4.2.1 Langage de programmation :

On a utilisé le langage C pour programmer la carte arduino à cause de la facilité de programmer, nous utiliserons le logiciel arduino IDE, ce logiciel à une option de nous aider ou cours de la programmation.

4.2.2 Arduino IDE (Environnement de Développement Intégré) :

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java. L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte

C'est le logiciel arduino qui permet :

- D'éditer un programme : des croquis (*sketch* en Anglais),
- De compiler ce programme dans le langage « machine » de l'Arduino,
- De téléversé le programme dans la mémoire de l'Arduino,
- De communiquer avec la carte Arduino grâce au terminal.

Chapitre 4 : Partie programmation et gestion des capteurs

4.2.3 Fenêtre général de l'application arduino :

La fenêtre de l'application Arduino comporte les éléments suivants :

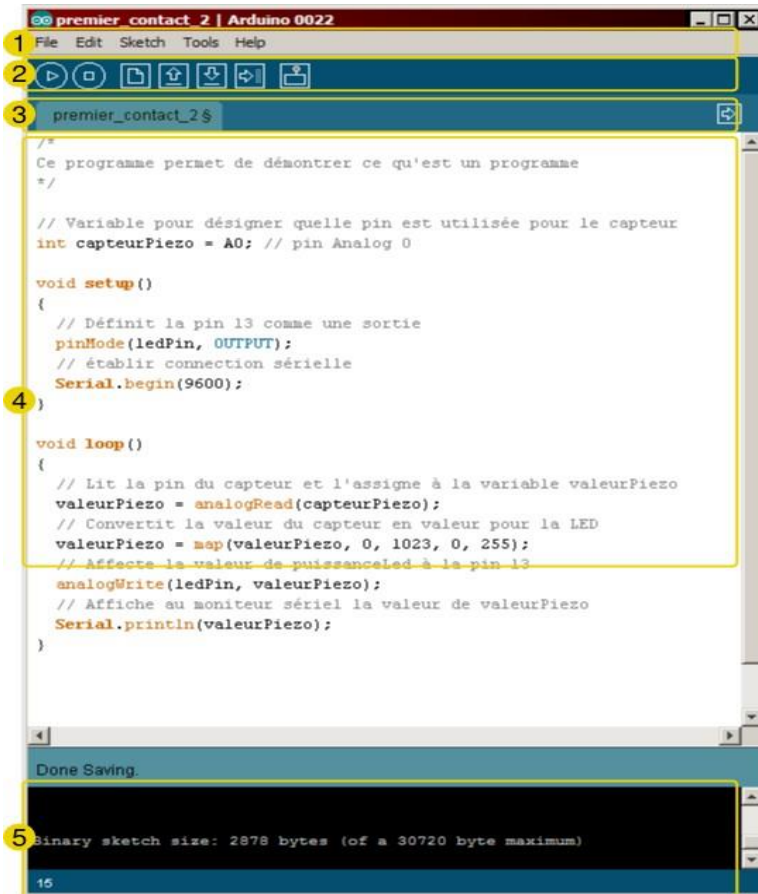


Figure4.1 le compilateur arduino IDE

1. un menu ;
2. une barre d'actions ;
3. un ou plusieurs onglets correspondant aux *sketchs* ;
4. une fenêtre de programmation ;
5. une console qui affiche les informations et erreurs de compilation et de téléversement du programme.

Chapitre 4 : Partie programmation et gestion des capteurs

4.3 Les organigrammes :

Les organigrammes représentent la première étape dans la mise en œuvre de la gestion des tâches. Cela nous a permis d'organiser les tâches, de vérifier l'ordre des instructions et leur interdépendance. Cette étape constitue une transition facilitant et simplifiant la phase de programmation.

4.3.1 Organigramme de capteur infrarouge :

le capteur est fixé sur une hauteur de 50 mm et un angle de 90^0 , il est utilisé afin d'avoir une distance entre le robot et l'obstacle pour la sécurité de la caméra qui est en avant.

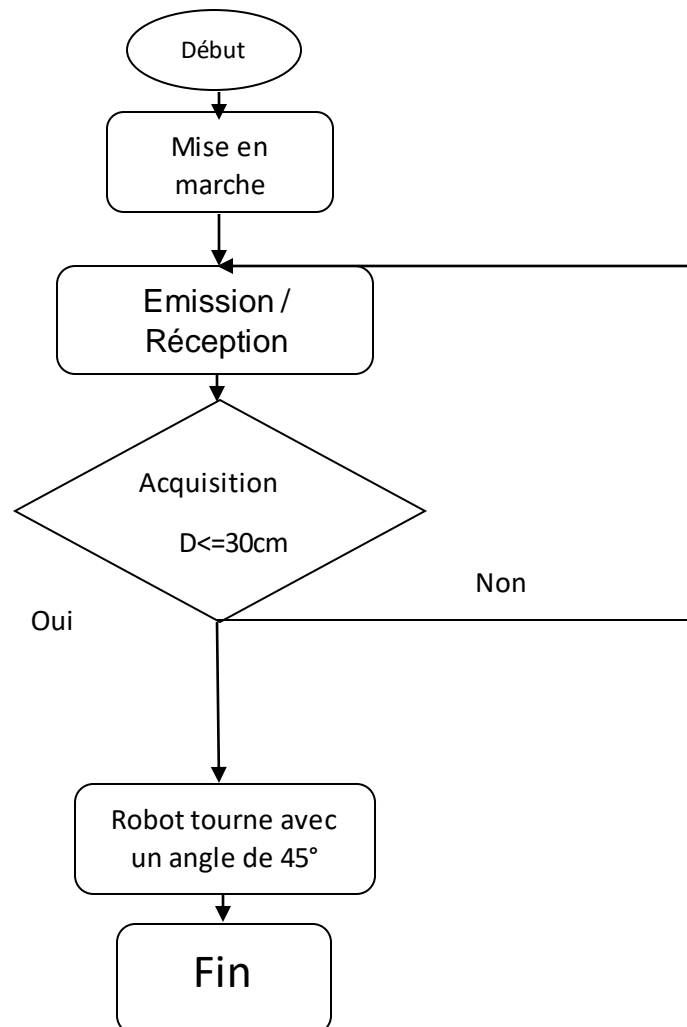


Figure 4.2 organigramme de gestion de l'infrarouge

4.3.2 Organigramme de robot et la caméra CMUCAM05 :

La structure fondamentale de la programmation, y compris les opérations et les décisions à suivre par le robot de façon autonome lorsqu'il détecte et suit la balle est illustré comme un organigramme de la figure suivante :

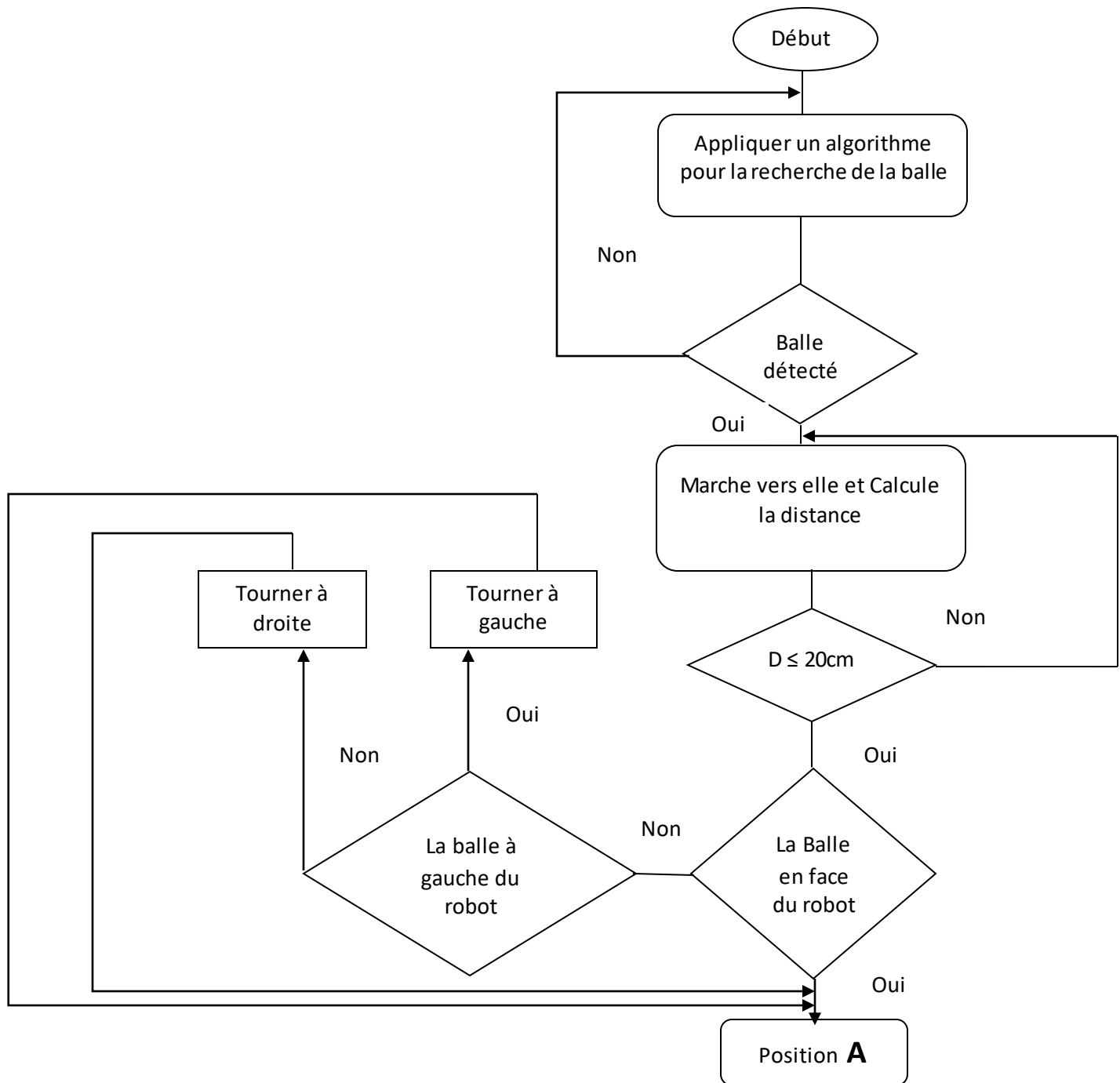


Figure 4.3 organigramme des décisions à suivre par le robot

4.3.3 Organigramme de capteur boussole avec robot :

Ce module de boussole a été spécifiquement conçu pour être utilisé dans les robots comme aide à la navigation. L'objectif était de produire un numéro unique pour représenter la direction à laquelle le robot est confronté.

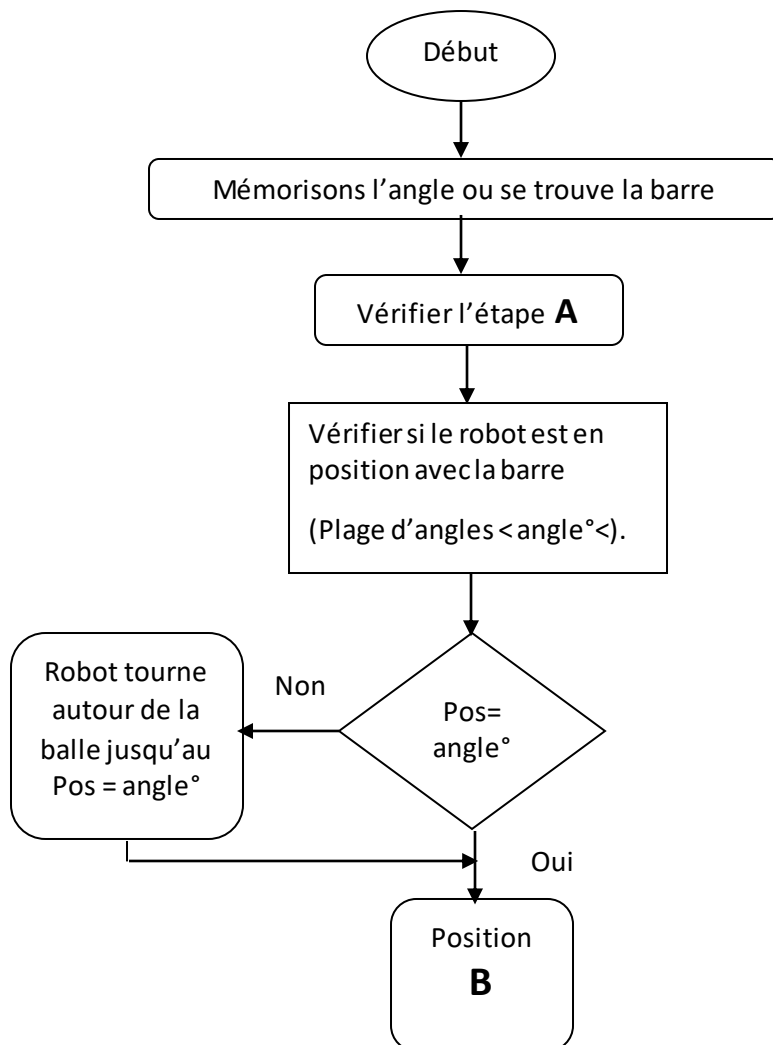


Figure 4.4 organigramme de gestion de capteur boussole

4.3.4 Organigramme globale du cycle de robot :

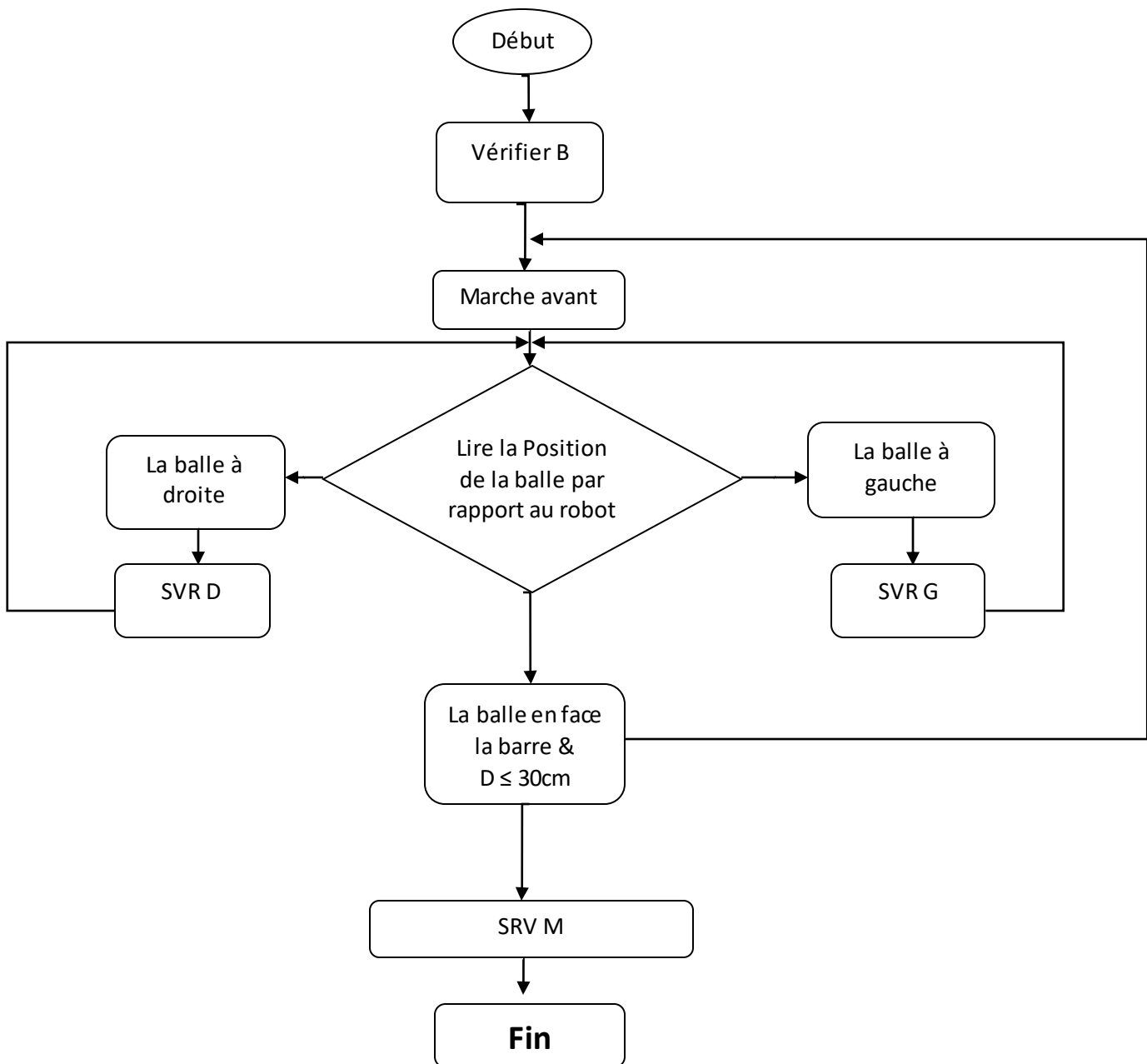


Figure 4.5 l'organigramme globale

Chapitre 4 : Partie programmation et gestion des capteurs

4.3.4 Organigramme de contrôle de vitesse :

Dans le contrôle de la vitesse, l'action de contrôle est calculée en fonction de la différence entre les vitesses de référence ($\phi r1, \phi r2$) et les vitesses angulaires réelles ($\phi1, \phi2$) tel que les nombres (1,2) est la numérotation des roues.

$$e_i = \phi r_i - \phi_i$$

e_i : l'erreur en [rad / s].

La figure 3.7 montre le schéma de principe du système de commande de vitesse du moteur en boucle fermée qui est utilisé dans ce projet pour contrôler la vitesse des roues. Pour atteindre la vitesse requise, la puissance délivrée au moteur est contrôlée par le contrôleur en boucle fermée. Le réglage de l'alimentation électrique fournie au moteur est généralement effectué par la pulse technique de modulation de largeur (PWM).

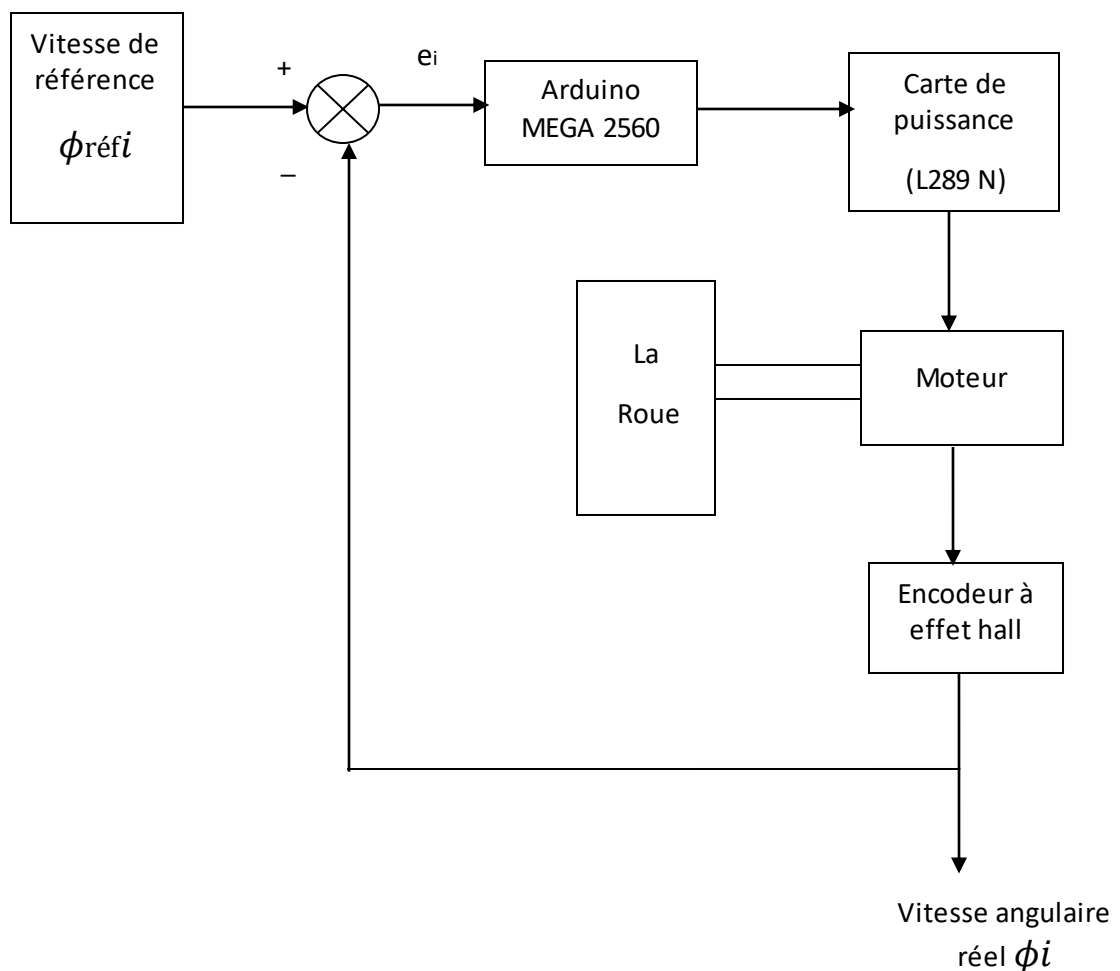


Figure 4.5 principe de système qui commande la vitesse de moteur en boucle fermée

Chapitre 4 : Partie programmation et gestion des capteurs

La vitesse réelle du moteur est mesurée par l'encodeur (capteur effet Hall), puis elle est entrée dans la carte Arduino et comparée avec la vitesse du moteur souhaitée.

L'algorithme de contrôle approprié dans ce projet est l'algorithme de vitesse Proportionnel-Intégral (PI) et écrit dans le microcontrôleur Arduino.

4.4 Résultats et discussion :

Cette section traite de tous les étalonnages, des tests théoriques et pratiques et des évaluations effectuées sur le Robot.

4.4.1 Réglage des paramètres CMUcam5 Pixy

Un enseignement de robustesse pour les paramètres CMUcam5 Pixy est ajusté en utilisant le logiciel PixyMoon pour détecter la balle dans l'environnement d'exploitation désiré. Les derniers paramètres sélectionnés sont indiqués dans le tableau 1.

Paramètre	valeur
Blocs maxi par châssis	1
Blocs maxi par couleur	2
Saturation minimale	10
Port de données (I2C)	1
Luminosité de la caméra	120

Tableau 4.1 : CMUcam5 Pixy derniers paramètres sélectionnés

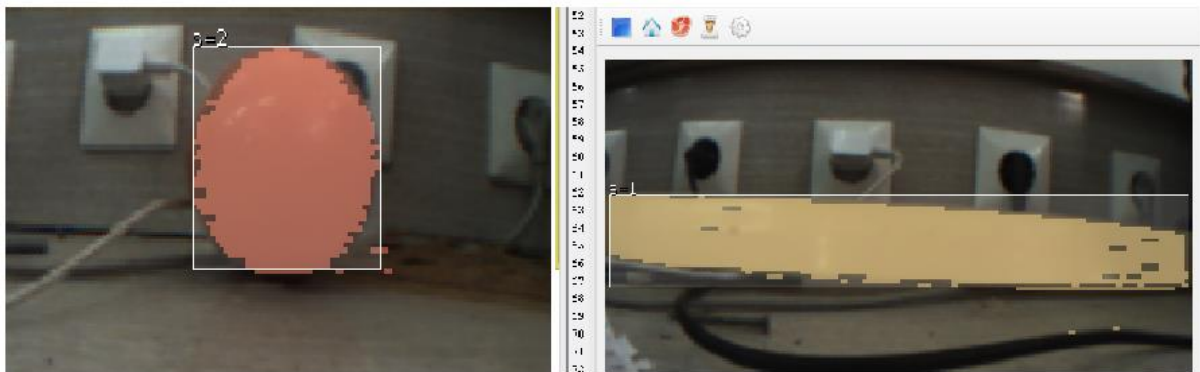


Figure 4.6 robot vision caméra

Le résultat de ce que le robot voit dans l'environnement de travail après avoir ajusté les paramètres CMUcam5 Pixy est illustré à la figure 4.6

Chapitre 4 : Partie programmation et gestion des capteurs

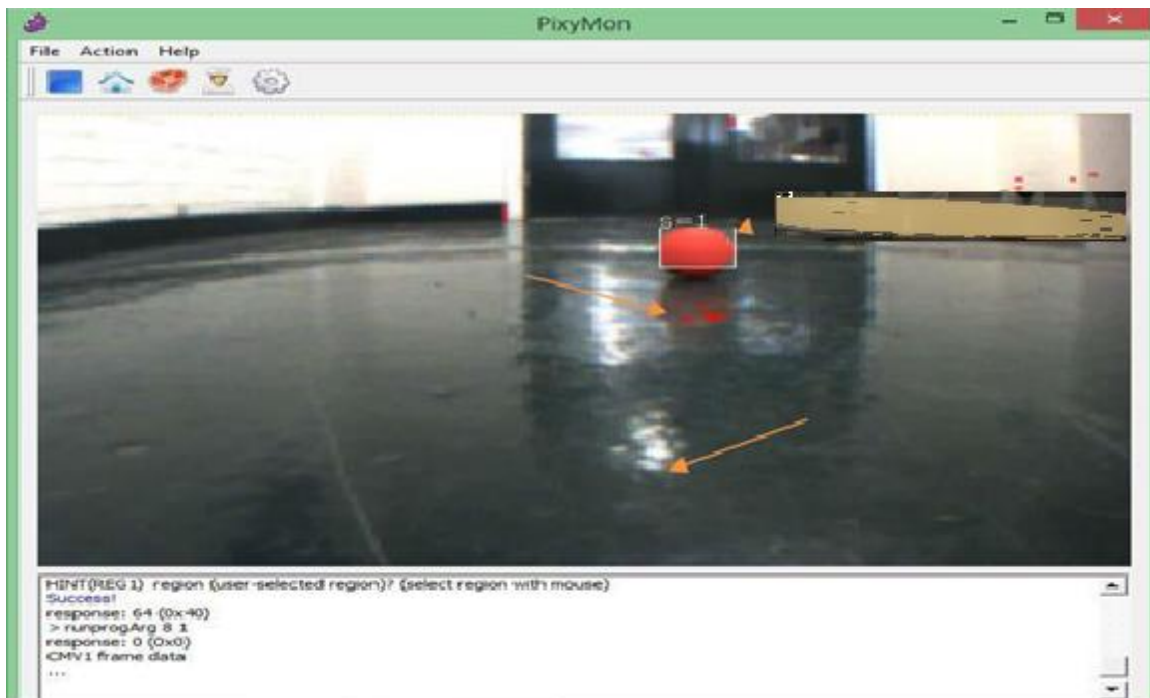


Figure 4.7 robot vision caméra le deux objets à la fois

À partir de la figure 4.7, il est conclu que pour détecter les objets (balle, barre) est nécessaire pour éviter toute interférence de couleur. Bien que l'algorithme de filtrage des couleurs dans le CMUcam5 Pixy soit positif dans le sens où il peut traiter un cadre entier très vite, la discrimination peut parfois être un problème.

Généralement, deux types d'erreurs peuvent survenir, le premier Lorsque CMUcam5 Pixy considère que la balle correspond à une couleur particulière, mais ce n'est pas l'objet correct, c'est un objet avec une teinte similaire. Le deuxième type d'erreur est lorsque CMUcam5 Pixy n'a pas réussi à détecter l'objet qui est requis pour détecter une couleur particulière. Ces erreurs peuvent être éliminées en ajustant plusieurs paramètres dans le CMUcam5 Pixy tel que la luminosité, la saturation Min...

Chapitre 4 : Partie programmation et gestion des capteurs



Figure4.8 détection de la balle

Chapitre 4 : Partie programmation et gestion des capteurs



Figure4.9 robot tire la balle par le servomoteur

Chapitre 4 : Partie programmation et gestion des capteurs



Figure 4.10 robot en face de la barre et tire la balle dans l'objectif

4.4 Conclusion

Dans ce chapitre nous avons présenté les programmes de gestion de robot mobile sous forme des organigrammes suivant la structure mécanique et électronique. Ces organigrammes décrivent la stratégie que suit le robot sur l'air du jeu afin de réaliser sa tâche. Pour le traitement d'image nous avons utilisé la caméra CMUCAM5 qui travail particulièrement avec son logiciel PixyMoon qui fait sélectionner les objets ciblés. Pour la programmation nous avons utilisé le langage C qui est un langage évolué très puissant et simple à manipuler pour la transformation de l'organigramme en instructions.

Listes des acronymes et abréviations

I²C : (signifie : Inter-Integrated Circuit, en anglais) est un bus informatique qui a émergé de la « guerre des standards » lancée par les acteurs du monde électronique. Conçu par Philips pour les applications de domotique et d'électronique domestique, il permet de relier facilement un microprocesseur et différents circuits, notamment ceux d'un téléviseur moderne : récepteur de la télécommande, réglages des amplificateurs basses fréquences, tuner, horloge, gestion de la prise péritel.

AVR : est le terme utilisé par Atmel pour désigner le cœur du processeur et la famille de microcontrôleurs qui le mettent en œuvre.

SRAM : La mémoire vive statique (ou SRAM pour l'anglais *Static Random Access Memory*) est un type de mémoire vive utilisant des bascules pour mémoriser les données. Mais contrairement à la mémoire dynamique, elle n'a pas besoin de rafraîchir périodiquement son contenu. Comme la mémoire dynamique, elle est volatile : elle ne peut se passer d'alimentation sous peine de voir les informations effacées irrémédiablement.

RAM : La mémoire vive ou mémoire à accès direct (traduction de l'anglais *Random Access Memory* (RAM)), par opposition à un accès séquentiel, est la mémoire informatique dans laquelle sont stockées les informations (données, instructions, etc.) traitées par un appareil (ordinateur, console de jeux vidéo, guichet automatique bancaire, téléphone mobile, oscilloscope numérique, etc.).

EEPROM : La mémoire EEPROM (Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et programmable) (aussi appelée E2PROM ou E²PROM) est un type de mémoire morte. Une mémoire morte est une mémoire utilisée pour enregistrer des informations qui ne doivent pas être perdues lorsque l'appareil qui les contient n'est plus alimenté en électricité.

SPI : Une liaison SPI (pour *Serial Peripheral Interface*) est un bus de données série synchrone baptisé ainsi par Motorola, qui opère en mode full-duplex. Les circuits communiquent selon un schéma maître-esclaves, où le maître contrôle la communication. Plusieurs esclaves

peuvent coexister sur un même bus, dans ce cas, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée *Slave Select (SS)*.

SS, MOSI, MISO, SCK : interface série synchrone SPI.

TTL : Transistor-Transistor Logic ou TTL est une famille de circuits logiques utilisée en électronique inventée dans les années 1960. Cette famille est réalisée avec la technologie du transistor bipolaire et tend à disparaître du fait de sa consommation énergétique élevée (comparativement aux circuits CMOS).

PWM : La modulation de largeur d'impulsions (MLI ; en anglais : *Pulse Width Modulation*, soit *PWM*), est une technique couramment utilisée pour synthétiser des signaux continus à l'aide de circuits à fonctionnement tout ou rien, ou plus généralement à états discrets.

Le principe général est qu'en appliquant une succession d'états discrets pendant des durées bien choisies, on peut obtenir *en moyenne sur une certaine durée* n'importe quelle valeur intermédiaire.

RVB : Rouge, vert, bleu, abrégé en RVB ou en RGB, de l'anglais « *Red, Green, Blue* » est, des systèmes de codage informatique des couleurs, le plus proche du matériel. Les écrans d'ordinateurs reconstituent une couleur par synthèse additive à partir de trois couleurs primaires, un rouge, un vert et un bleu, formant sur l'écran une mosaïque trop petite pour être aperçue. Le codage RVB indique une valeur pour chacune de ces couleurs primaires.

Pixel : Le pixel (souvent abrégé px) est l'unité de base permettant de mesurer la définition d'une image numérique matricielle. Son nom provient de la locution anglaise *picture element*, qui signifie « élément d'image ».

IDE : En programmation informatique, un environnement de développement est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction. Certains environnements sont dédiés à un langage de programmation en particulier.

PID : Le régulateur PID, appelé aussi correcteur PID (proportionnel, intégrateur, dérivateur ou proportionnel, intégral, dérivé) est un système de contrôle, permettant d'effectuer un asservissement en boucle fermée d'un système industriel ou « procédé ». C'est le régulateur le plus utilisé dans l'industrie, et il permet de régler un grand nombre de grandeurs physiques.

IR : Le rayonnement infrarouge (IR) est un rayonnement électromagnétique d'une longueur d'onde supérieure à celle de la lumière visible mais plus courte que celle des micro-ondes.

LED : Une diode électroluminescente (abrégé en DEL en français, ou LED, de l'anglais : *light-emitting diode*), est un dispositif optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. Une diode électroluminescente ne laisse passer le courant électrique que dans un seul sens (le sens passant, comme une diode classique, l'inverse étant le sens bloquant) et produit un rayonnement monochromatique ou polychromatique non cohérent à partir de la conversion d'énergie électrique lorsqu'un courant la traverse.

Introduction générale

L'homme a toujours souhaité se libérer du travail dans ce qu'il a d'astreignant, de fatigant, d'inintéressant. La robotique tire ses origines des civilisations les plus antiques. Il est peut-être même vrai que toutes les autres sciences ont existé juste pour permettre à la robotique de se développer afin d'aider l'Homme à créer ce « esclave » qui appliquera ses ordres au doigt et à l'œil et qui le libérera à jamais du travail.

Les générations de robots mobiles se sont vite succédé offrant à chaque fois plus de performances en matière de perception, d'autonomie et de pouvoir décisionnel. Ces progrès étaient un résultat inévitable des développements qu'ont connus la mécanique, la microélectronique et l'informatique. C'est pour cela qu'en très peu de temps, les générations de robots se succèdent pour voir naître une branche de la robotique visant de plus amples horizons : c'est l'ère de la robotique mobile.

Objectif :

L'objectif principal de notre travail est de réaliser un robot mobile en utilisant deux cartes électroniques Arduino et une caméra pixy cam. La tâche principale du robot est la détection de la balle et la prendre à l'objectif même il peut éviter les obstacles.

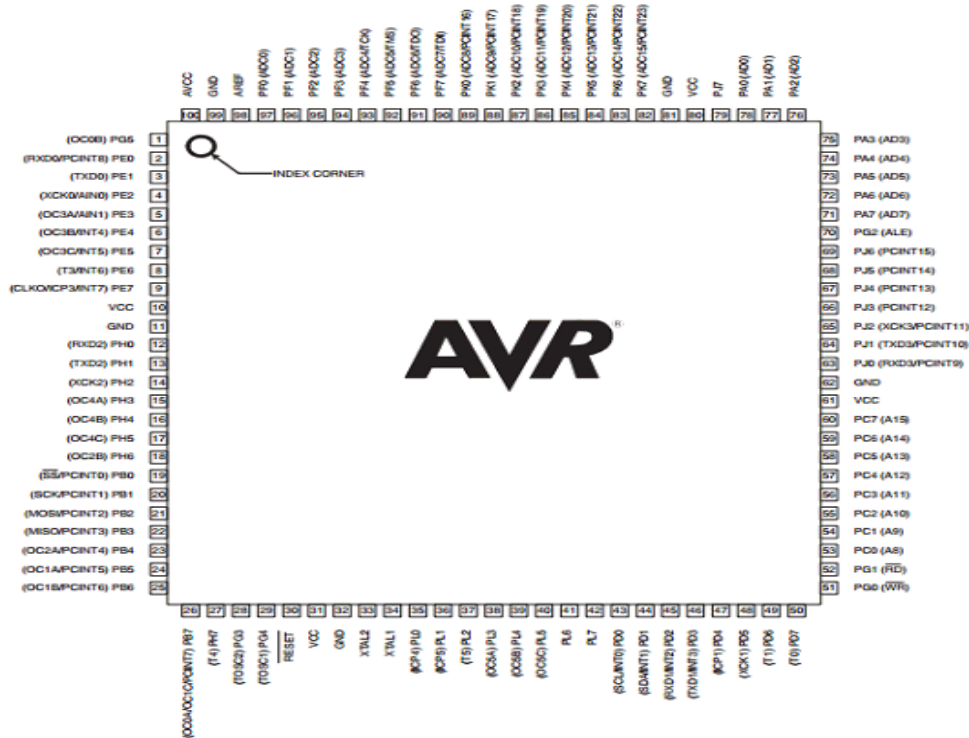
Ce travail est alors organisé en quatre chapitres de la manière suivante : Le premier chapitre présente certaines généralités concernant la robotique et les différents types des robots et robots mobiles. Le second chapitre introduit la conception mécanique du robot mobile. Le troisième chapitre est consacré à la conception électrique. Le quatrième chapitre traite la partie pratique et programmation.

Bibliographie

- [1] Auteur: Bruno Siciliano, Oussama Khatib : Springer Handbook of Robotics
- [2] Auteur : Philippe Coiffet : ' 10 questions à Philippe Coiffet sur les robots ', Editions Le Manuscrit, 2009 ,
- [3] Auteur: Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo
Robotics: ' Modelling, Planning and Control '
- [4] Introduction to robotics: analysis, control, applications; Auteur: Saeed B. Niku
- [5] Auteur: V.S. Bagad : ' Mechatronics ', Edition 2008 , ISBN 9788184314908, .
- [6] Belkhadria Khemisti, "commande d'un robot mobile par réseaux de neurones artificiels"
Mémoire en vue de l'obtention du diplôme de magister en électronique. Option : Robotique.
Université El Hadj Lakhdar Batna
- [7] <https://www.gotronic.fr/art-motoreducteur-encodeur-emg30-866.htm>
- [8] <http://www.redohm.fr/2014/12/arduino/>
- [9] <https://infouelec.wordpress.com/2013/03/24/arduino-quest-ce-que-cest/>
- [10] P1773-module-boussole-cmps03.html
- [11] sbarthelemy.com/électronique/docs-1/cmps03---module-boussole
- [12] <https://www.lextronic.fr/P30095-module-de-reconnaissance-vido-pixy.html>
- [13] http://intelligence-artificielle.livehost.fr/?page_id=254

MICROSHIP Features

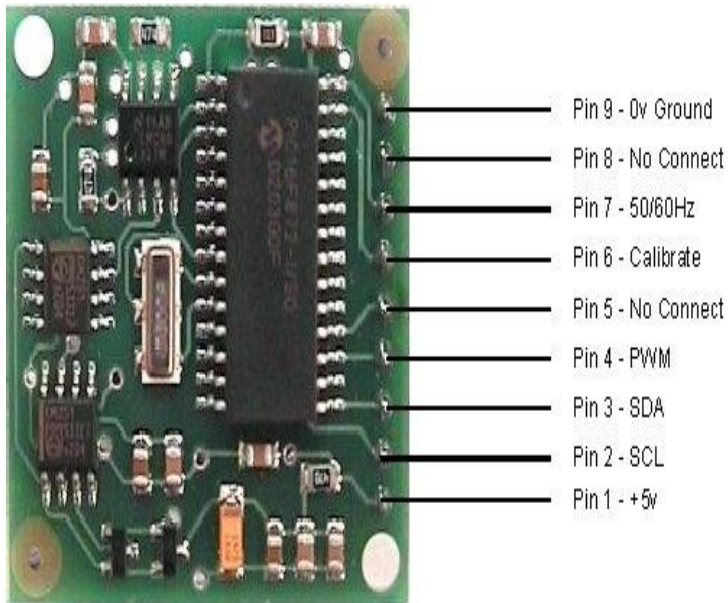
Figure 1-1. TQFP-pinout ATmega640/1280/2560



- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 64K/128K/256K Bytes of In-System Self-Programmable Flash
 - 4Kbytes EEPROM
 - 8Kbytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85° C/ 100 years at 25° C
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
 - Endurance: Up to 64Kbytes Optional External Memory Space
 - Atmel® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels

- JTAG (IEEE® std. 1149.1 compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four 8-bit PWM Channels
 - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits
 - (ATmega1281/2561, ATmega640/1280/2560)
 - Output Compare Modulator
 - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
 - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
 - Master/Slave SPI Serial Interface
 - Byte Oriented 2-wire Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby,

CMPS03 - Robot Compass Module



This compass module has been specifically designed for use in robots as an aid to navigation. The aim was to produce a unique number to represent the direction the robot is facing. The compass uses the Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the Earth's magnetic field. The output from two of them mounted at right angles to each other is used to compute the direction of the horizontal

component of the Earth's magnetic field. Connections to the compass module

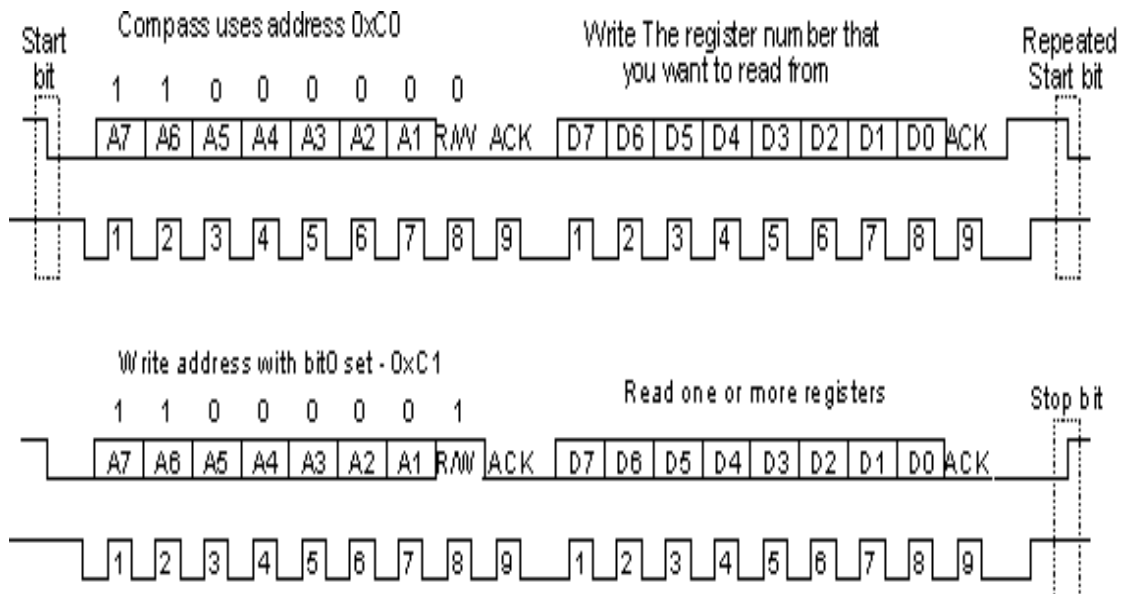
The compass module requires a 5v power supply at a nominal 15mA.

There are two ways of getting the bearing from the module. A PWM signal is available on pin 4, or an I2C interface is provided on pins 2,3.

The PWM signal is a pulse width modulated signal with the positive width of the pulse representing the angle. The pulse width varies from 1mS (0°) to 36.99mS (359.9°) – in other words $100\mu\text{S}/^\circ$ with a +1mS offset. The signal goes low for 65mS between pulses, so the cycle time is 65mS + the pulse width - ie. 66ms-102ms. The pulse is generated by a 16 bit timer in the processor giving a 1 μS resolution, however I would not recommend measuring this to anything better than 0.1 $^\circ$ (10 μS). Make sure you connect the I2C pins, SCL and SDA, to the 5v supply if you are using the PWM, as there are no pull-up resistors on these pins.

Pin 2,3 are an I2C interface and can be used to get a direct readout of the bearing. If the I2C interface is not used then these pins should be pulled high (to +5v) via a couple of resistors. Around 47k is ok

Annexes



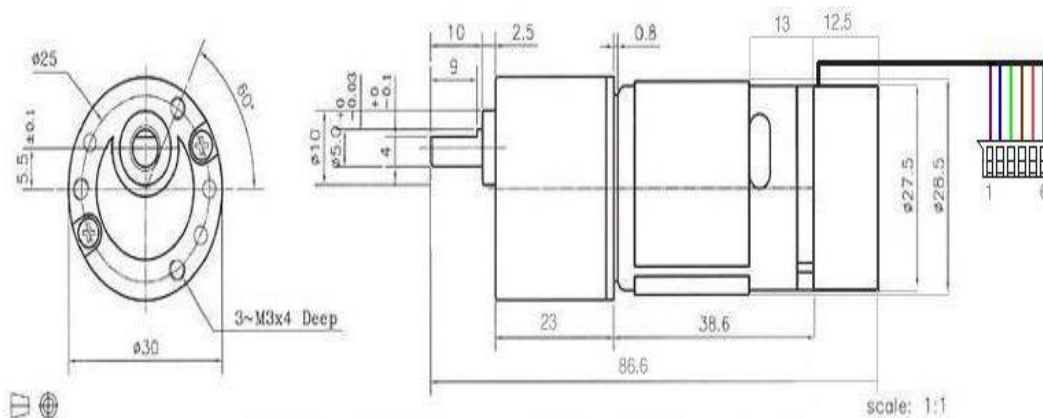
First send a start bit, the module address (0XC0) with the read/write bit low, then the register number you wish to read. This is followed by a repeated start and the module address again with the read/write bit high (0XC1). You now read one or two bytes for 8bit or 16bit registers respectively. 16bit registers are read high byte first. The compass has a 16 byte array of registers, some of which double up as 16 bit registers as follows; The I2C interface does not have any pull-up resistors on the board, these should be provided elsewhere, most probably with the bus master. They are required on both the SCL and SDA lines, but only once for the whole bus, not on each module. I suggest a value of 1k8 if you are going to be working up to 400KHz and 1k2 or even 1k if you are going up to 1MHz. The compass is designed to work at up to the standard clock speed (SCL) of 100KHz, however the clock speed can be raised to 1MHZ providing the following precaution is taken;

At speeds above around 160KHz the CPU cannot respond fast enough to read the I2C data. Therefore a small delay of 50uS should be inserted either side of writing.

EMG30, mounting bracket and wheel specification

The EMG30 (encoder, motor, gearbox 30:1) is a 12v motor fully equipped with encoders and a 30:1 reduction gearbox. It is ideal for small or medium robotic applications, providing cost effective drive and feedback for the user. It also includes a standard noise suppression capacitor across the motor windings.

Measurements



Connector

The EMG30 is supplied with a 6 way JST connector (part no PHR-6) at the end of approx 90mm of cable as standard.

The connections are:

Wire colour	Connection
Purple (1)	Hall Sensor B Vout
Blue (2)	Hall sensor A Vout
Green (3)	Hall sensor ground

Brown (4)	Hall sensor Vcc
Red (5)	+ Motor
Black (6)	- Motor

On the MD25 they are powered from 12v and pulled up to 5v for the signals.

specification

Rated voltage	12v	Rated current	530mA
Rated torque	1.5kg/cm		
Rated speed	170rpm		

DUAL FULL-BRIDGE DRIVER L298N



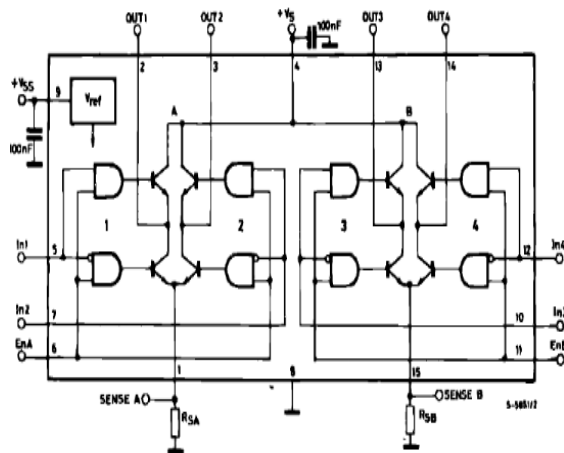
Multiwatt15

ORDERING NUMBERS : L298N (Multi watt Vert.)

L298HN (Multi watt Horiz.)

L298P (PowerSO20)

BLOCK DIAGRAM



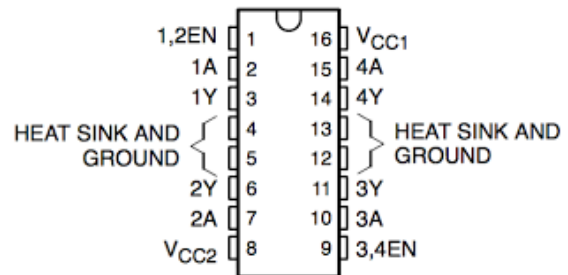
■ ■ OPERATING SUPPLY VOLTAGE UP TO

46 V ■ TOTAL DC CURRENT UP TO 4 A

■ LOW SATURATION VOLTAGE

■ OVERTEMPERATURE PROTECTION

LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V
(HIGH NOISE IMMUNITY)



DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-

lead Multiwatt and PowerSO20 packages. It is a

high voltage, high current dual full-bridge driver designed to

accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and

stepping motors. Two enable inputs are provided to

enable or disable the device independently of their input

signals. The emitters of the lower transistors of each bridge are connected together and the

corresponding external terminal can be used for the connection of an external sensing resistor.

An additional

supply input is provided so that the logic works at a

lower voltage.

