

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

جامعة سعد دحلب البليدة

UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك

FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE PROJET DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT

EN ELECTRONIQUE

OPTION : CONTRÔLE

**CONTRÔLE DE LA TRAJECTOIRE D'UN SYSTEME
MOBILE**

Présenté par :

❖ ZEMOUCHI ALI
❖ EZZEROUG YASSINE

Propose et dirigé par :

❖ MR : KAZED
BOUALAM

PROMOTION: 2012/2013

Session: juillet 2013

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

جامعة سعد دحلب البليدة

UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك

FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE PROJET DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT

EN ELECTRONIQUE

OPTION : CONTRÔLE

**CONTRÔLE DE LA TRAJECTOIRE D'UN SYSTEME
MOBILE**

Présenté par :

❖ ZEMOUCHI ALI
❖ EZZEROUG YASSINE

Propose et dirigé par :

❖ MR : KAZED
BOUALAM

PROMOTION: 2012/2013

Session: juillet 2013

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

جامعة سعد دحلب البليدة

UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك

FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE PROJET DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT

EN ELECTRONIQUE

OPTION : CONTRÔLE

**CONTRÔLE DE LA TRAJECTOIRE D'UN SYSTEME
MOBILE**

Présenté par :

❖ ZEMOUCHI ALI
❖ EZZEROUG YASSINE

Propose et dirigé par :

❖ MR : KAZED
BOUALAM

PROMOTION: 2012/2013

Session: juillet 2013

CHAPITRE V : IMPLEMENTATIO N SUR LE ROBOT ET RESULTATS.

CHAPITRE V : IMPLEMENTATIO N SUR LE ROBOT ET RESULTATS.

CHAPITRE V : IMPLEMENTATIO N SUR LE ROBOT ET RESULTATS.

V.4. 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 .

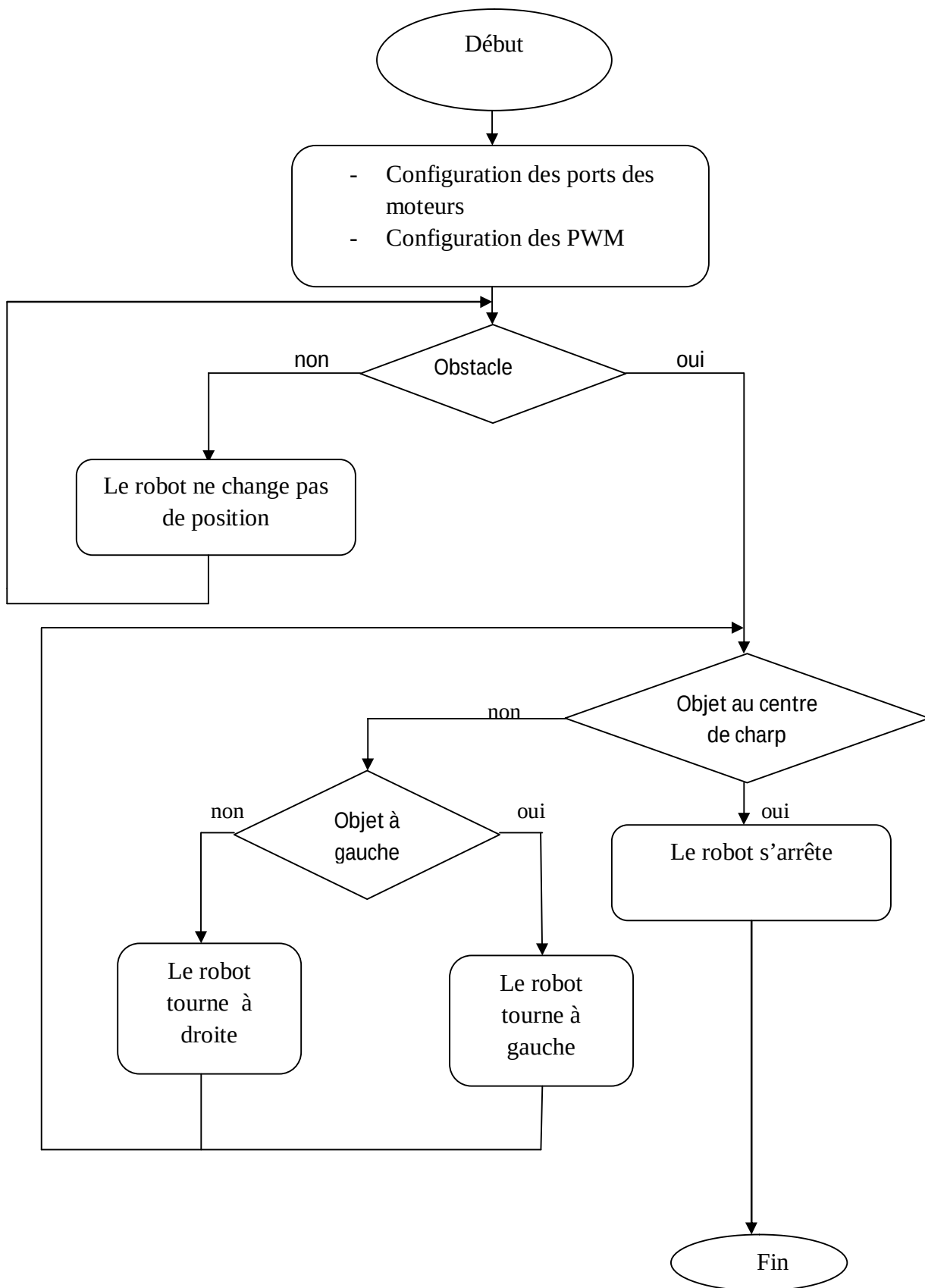
V.4. Conclusion:

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 .

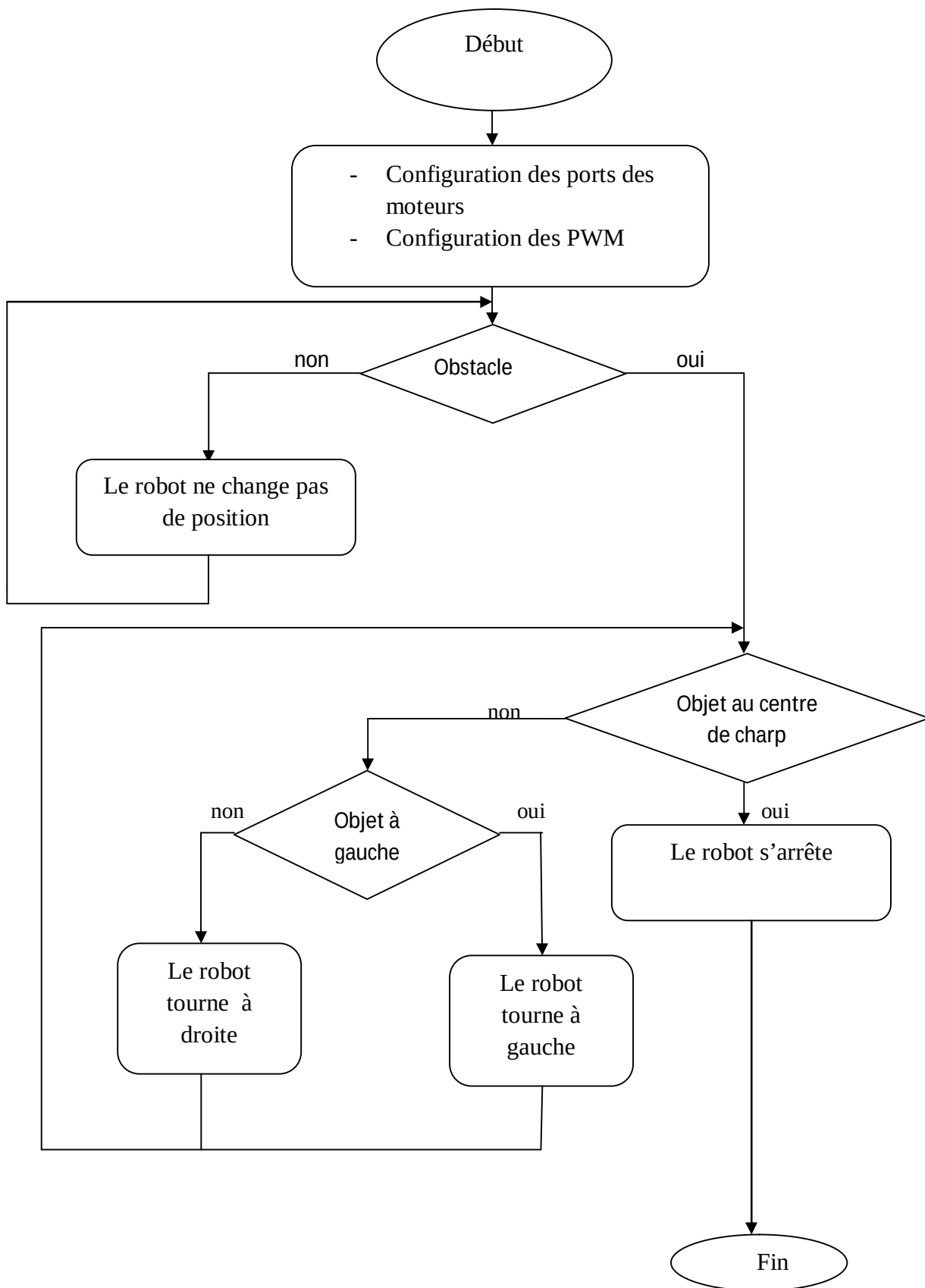
V.4. Conclusion:

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 .

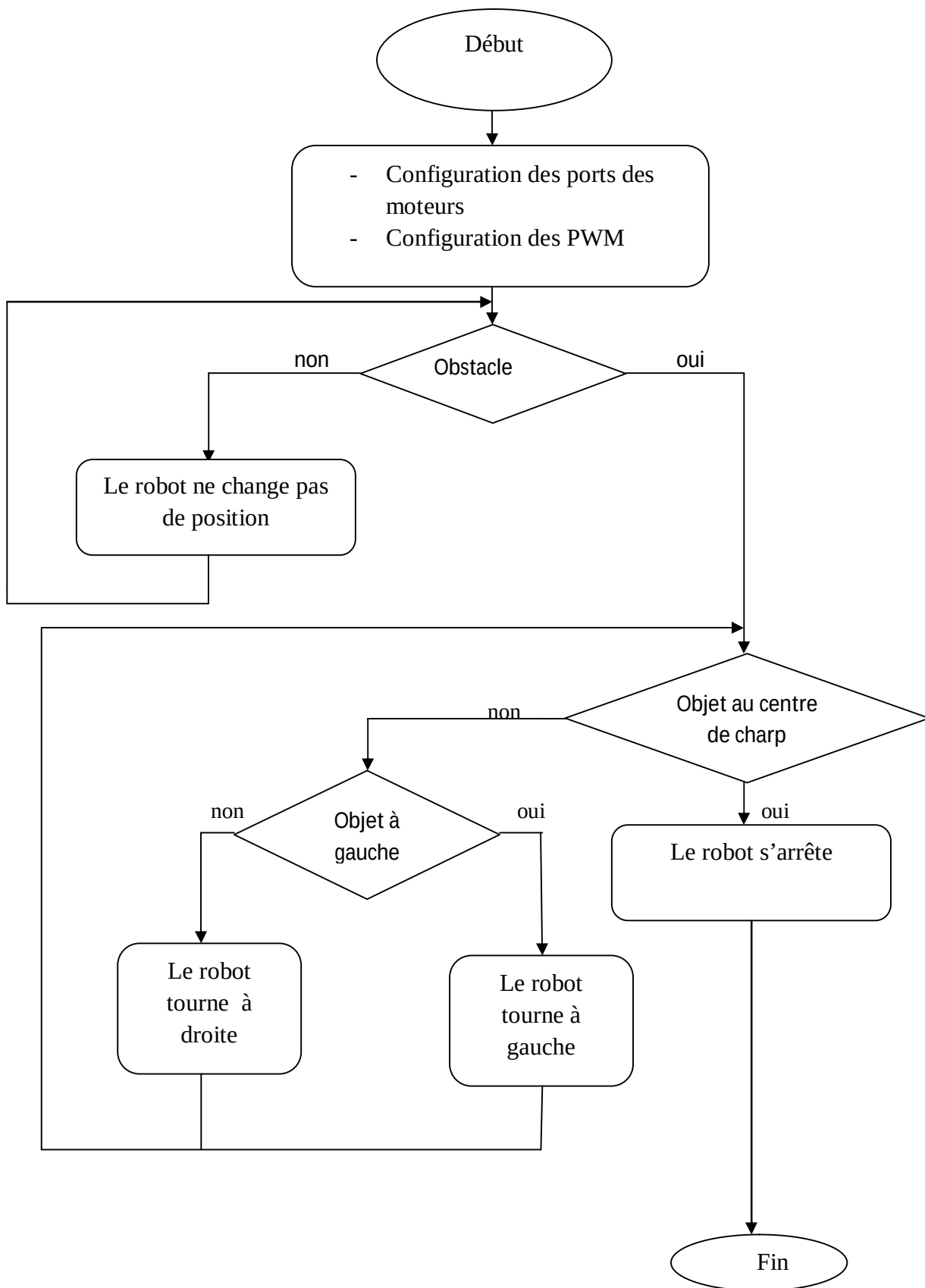
V.3. *algorithme droite gauche :*



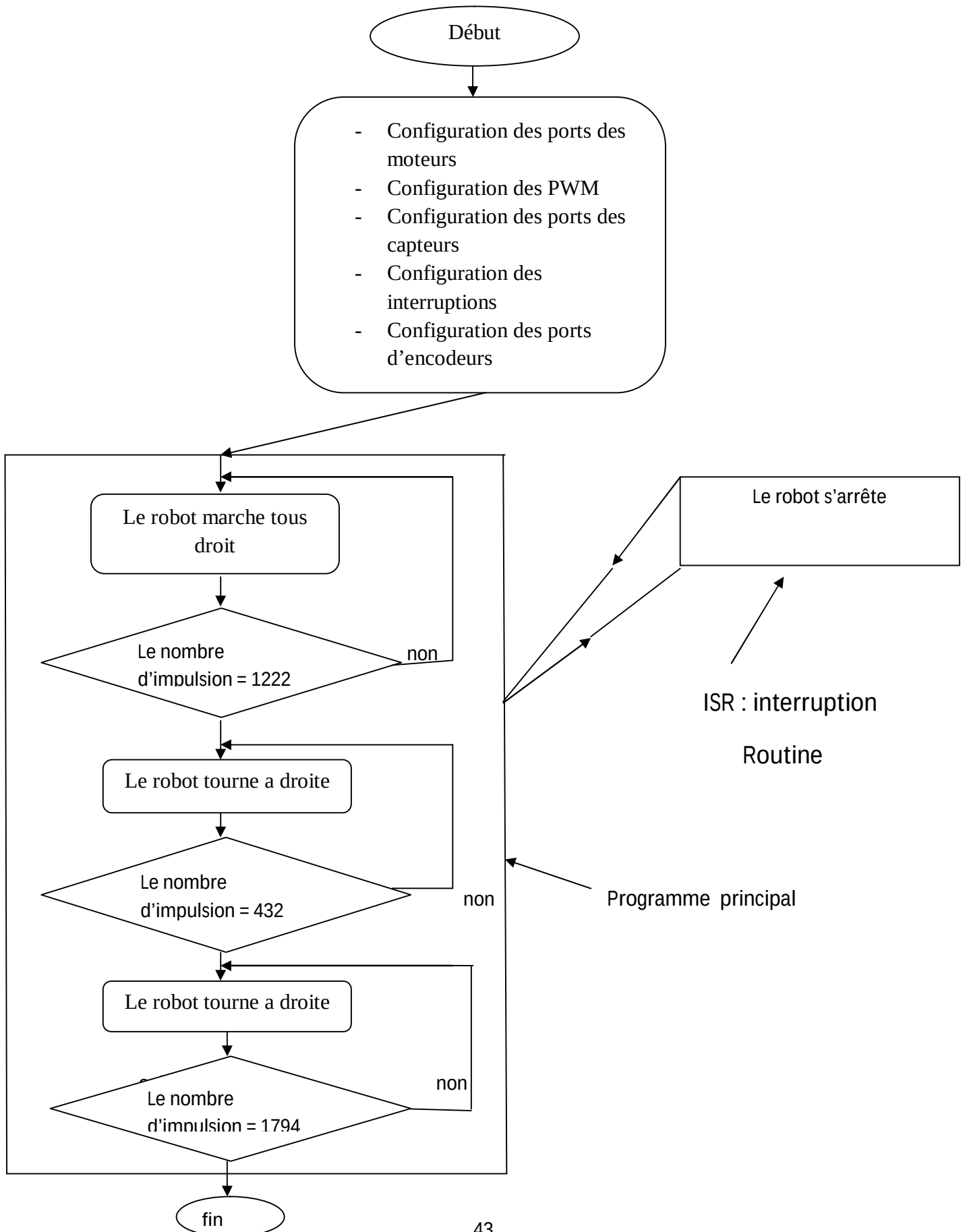
V.3. *algorithme droite gauche :*



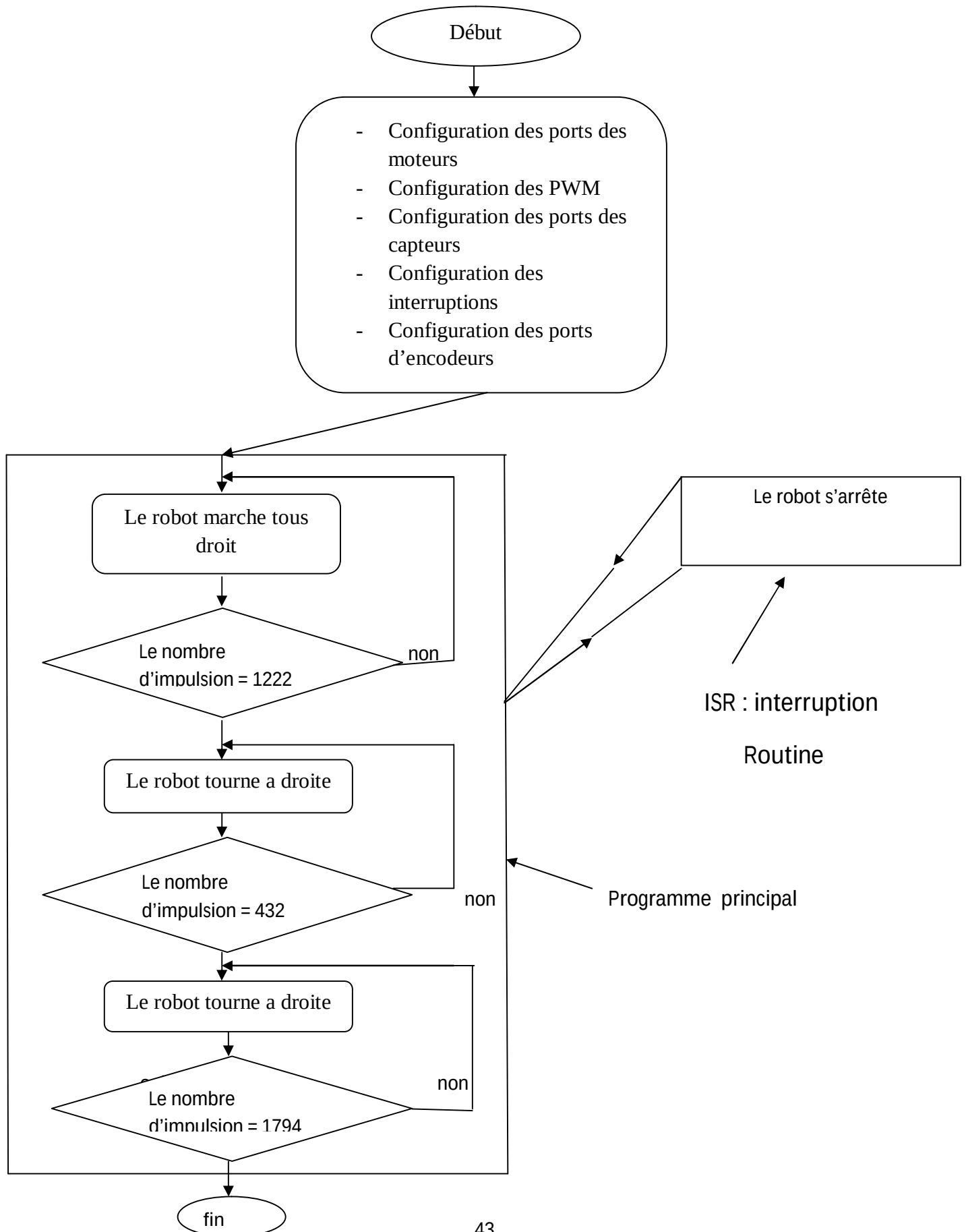
V.3. *algorithme droite gauche :*



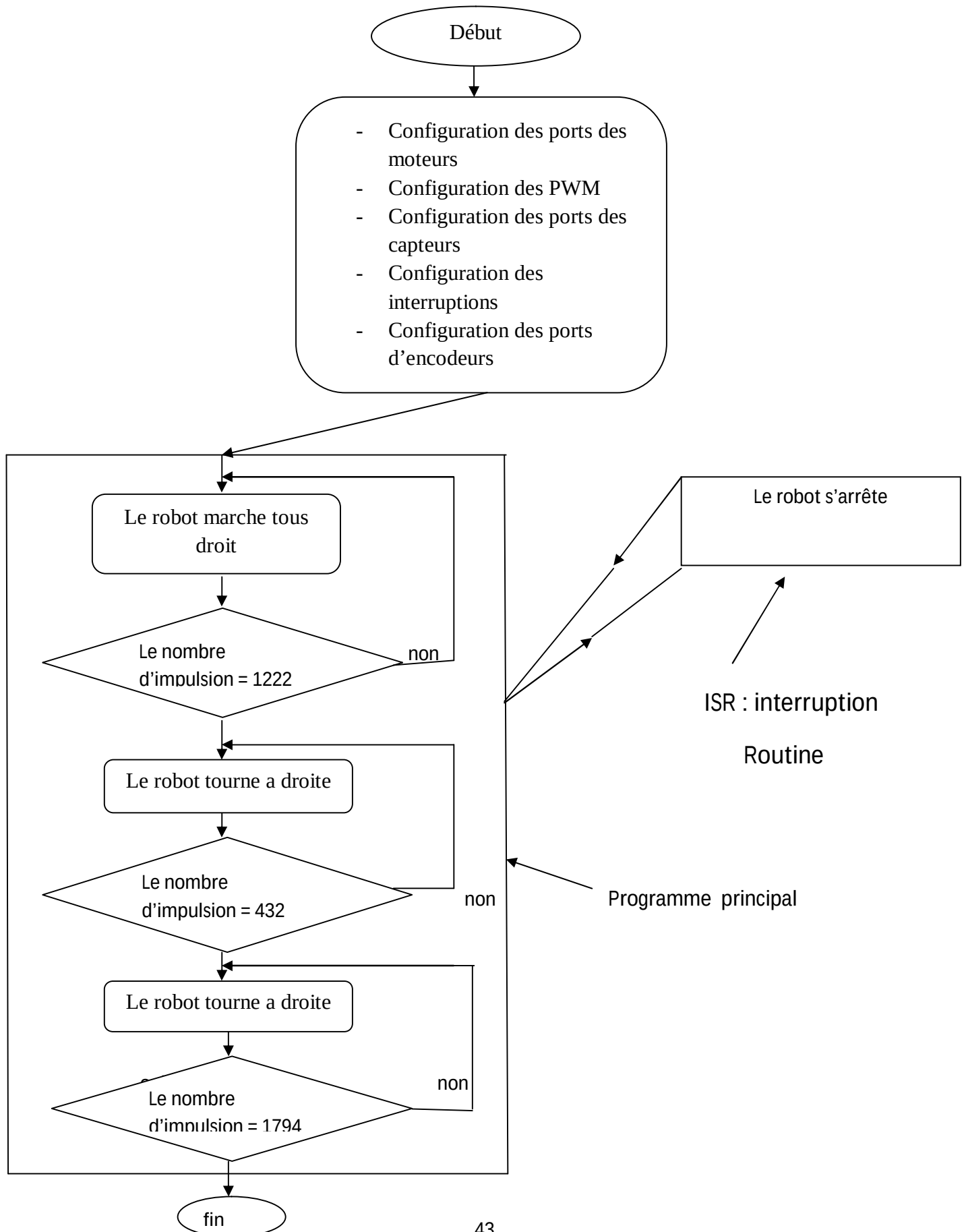
V.3. Les algorithmes : Algorithme de la première approche



V.3. Les algorithmes : Algorithme de la première approche



V.3. Les algorithmes : Algorithme de la première approche



CONCLUSION

GENERALE

CONCLUSION

GENERALE

CONCLUSION

GENERALE

Dédicaces :

Je dédie ce modeste travail à ma famille, particulièrement à **ma mère, ma mère, ma mère FERHOUL F/Z,**

mon père Rachid, mes frères hamoud et mohamed ,mes sœur , et mestapha et mohamed-riadh et imene et la petit maria et la plus belle maroua pour leur soutien et l'assistance permanente qu'ils m'ont

apporté avec patience et détermination tout au long de mon cursus universitaire .

Ces dédicaces vont aussi de manière chaleureuse et amicale à :

- *Tous mes amis : Berabeh elyamine ,Amin paul, Maamar, Aaradj ,Fouad, Rabie, Adel, Smail et Smail layri, Laichi, Walid,Redouane et Abd elkrim,Abd elrahman, Farid et Farid et Mohamed eldjouhar,Ahmed ,Stamboli,Brahim nafai,Mahfoudh, Hamza, Nassim, Sihem, Rym, Amel, Bia, surtout Ingrida n'oublie pas Nadine.....*
- *Tous mes amis de UNJA :Amin,Hamza,Nabil,Redouane,.....*
- **Tous mes amis de ROBOTIQUE.....**
- *Tous mes enseignants dont on a suivi les cours durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études.*

ZEMOUCHI ALI

Dédicaces :

Je dédie ce modeste travail à ma famille, particulièrement à **ma mère, ma mère, ma mère FERHOUL F/Z,**

mon père Rachid, mes frères hamoud et mohamed ,mes sœur , et mestapha et mohamed-riadh et imene et la petit maria et la plus belle maroua pour leur soutien et l'assistance permanente qu'ils m'ont

apporté avec patience et détermination tout au long de mon cursus universitaire .

Ces dédicaces vont aussi de manière chaleureuse et amicale à :

- *Tous mes amis : Berabeh elyamine ,Amin paul, Maamar, Aaradj ,Fouad, Rabie, Adel, Smail et Smail layri, Laichi, Walid,Redouane et Abd elkrim,Abd elrahman, Farid et Farid et Mohamed eldjouhar,Ahmed ,Stamboli,Brahim nafai,Mahfoudh, Hamza, Nassim, Sihem, Rym, Amel, Bia, surtout Ingrida n'oublie pas Nadine.....*
- *Tous mes amis de UNJA :Amin,Hamza,Nabil,Redouane,.....*
- *Tous mes amis **de ROBOTIQUE**.....*
- *Tous mes enseignants dont on a suivi les cours durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études.*

ZEMOUCHI ALI

Dédicaces :

Je dédie ce modeste travail à ma famille, particulièrement à **ma mère, ma mère, ma mère FERHOUL F/Z,**

mon père Rachid, mes frères hamoud et mohamed ,mes sœur , et mestapha et mohamed-riadh et imene et la petit maria et la plus belle maroua pour leur soutien et l'assistance permanente qu'ils m'ont

apporté avec patience et détermination tout au long de mon cursus universitaire .

Ces dédicaces vont aussi de manière chaleureuse et amicale à :

- *Tous mes amis : Berabeh elyamine ,Amin paul, Maamar, Aaradj ,Fouad, Rabie, Adel, Smail et Smail layri, Laichi, Walid,Redouane et Abd elkrim,Abd elrahman, Farid et Farid et Mohamed eldjouhar,Ahmed ,Stamboli,Brahim nafai,Mahfoudh, Hamza, Nassim, Sihem, Rym, Amel, Bia, surtout Ingrida n'oublie pas Nadine.....*
- *Tous mes amis de UNJA :Amin,Hamza,Nabil,Redouane,.....*
- **Tous mes amis de ROBOTIQUE.....**
- **Tous mes enseignants dont on a suivi les cours durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études.**

ZEMOUCHI ALI

ملخص:

الهدف من هذا المشروع هو تصميم و انجاز روبوت متحرك ذو تحكم ذاتي، يحتوي الروبوت على محركين ذو تيار مستمر، نتحكم فيها بواسطة بطاقة الكترونية تحتوي على ميكرومراقب اردوينو. يقوم الروبوت بالتنقل داخل محيط توجد فيه حواجز ثابتة و اخرى متحركة ومن أجل هذا زدنا الروبوت بلاقطات تمكنه من التعرف على الحواجز الموجودة داخل المحيط و تفاديها.

كلمات المفاتيح:

روبوت ،ميكرومراقب اردوينو ، الحواجز ، لاقطات

Résumé :

Le travail présenté dans ce mémoire est une contribution à la conception et à la réalisation d'un robot mobile autonome, il contient deux moteurs à courant continue contrôlés et commandés par une carte électronique à microcontrôleur Arduino».

Ce robot est équipé d'un capteur qui lui permet de reconnaitre et d'éviter les obstacles existants dans l'environnement où il se déplace.

Mots clés :

Robot, microcontrôleur Arduino, obstacles, capteur.

Abstract :

The aim of this project is to achieve and design a autonomous mobile robot. The robot contains two DC motors controlled by an electronic card that contains the microcontroller

This robot is moved in a surrounding where there are static obstacles and others in movement. For all this we create the robot with sensor that make it knows all obstacles in the surrounding and avoid them.

Keywords :

Robot, microcontrôleur Arduino , obstacles , sensor.

ملخص:

الهدف من هذا المشروع هو تصميم و انجاز روبوت متحرك ذو تحكم ذاتي، يحتوي الروبوت على محركين ذو تيار مستمر، نتحكم فيها بواسطة بطاقة الكترونية تحتوي على ميكرومراقب اردوينو. يقوم الروبوت بالتنقل داخل محيط توجد فيه حواجز ثابتة و اخرى متحركة ومن أجل هذا زدنا الروبوت بلاقطات تمكنه من التعرف على الحواجز الموجودة داخل المحيط و تفاديها.

كلمات المفاتيح:

روبوت ،ميكرومراقب اردوينو ، الحواجز ، لاقطات

Résumé :

Le travail présenté dans ce mémoire est une contribution à la conception et à la réalisation d'un robot mobile autonome, il contient deux moteurs à courant continue contrôlés et commandés par une carte électronique à microcontrôleur Arduino».

Ce robot est équipé d'un capteur qui lui permet de reconnaitre et d'éviter les obstacles existants dans l'environnement où il se déplace.

Mots clés :

Robot, microcontrôleur Arduino, obstacles, capteur.

Abstract :

The aim of this project is to achieve and design a autonomous mobile robot. The robot contains two DC motors controlled by an electronic card that contains the microcontroller

This robot is moved in a surrounding where there are static obstacles and others in movement. For all this we create the robot with sensor that make it knows all obstacles in the surrounding and avoid them.

Keywords :

Robot, microcontrôleur Arduino , obstacles , sensor.

ملخص:

الهدف من هذا المشروع هو تصميم و انجاز روبوت متحرك ذو تحكم ذاتي، يحتوي الروبوت على محركين ذو تيار مستمر، نتحكم فيها بواسطة بطاقة الكترونية تحتوي على ميكرومراقب اردوينو. يقوم الروبوت بالتنقل داخل محيط توجد فيه حواجز ثابتة و اخرى متحركة ومن أجل هذا زدنا الروبوت بلاقطات تمكنه من التعرف على الحواجز الموجودة داخل المحيط و تفاديها.

كلمات المفاتيح:

روبوت ،ميكرومراقب اردوينو ، الحواجز ، لاقطات

Résumé :

Le travail présenté dans ce mémoire est une contribution à la conception et à la réalisation d'un robot mobile autonome, il contient deux moteurs à courant continue contrôlés et commandés par une carte électronique à microcontrôleur Arduino».

Ce robot est équipé d'un capteur qui lui permet de reconnaitre et d'éviter les obstacles existants dans l'environnement où il se déplace.

Mots clés :

Robot, microcontrôleur Arduino, obstacles, capteur.

Abstract :

The aim of this project is to achieve and design a autonomous mobile robot. The robot contains two DC motors controlled by an electronic card that contains the microcontroller

This robot is moved in a surrounding where there are static obstacles and others in movement. For all this we create the robot with sensor that make it knows all obstacles in the surrounding and avoid them.

Keywords :

Robot, microcontrôleur Arduino , obstacles , sensor.

TABLE DES MATIERES

REMERCIEMENTS

DEDICACES

RESUME

ABSTRACT

ملخص

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

INTRODUCTION GENERALE

CHAPITRE I : Notions de robotique et de robots mobiles.

I-1- Introduction	
I-2- Historique	
I-3- La robotique mobile	
I-3-1- Autonomie en robotique mobile	
I-3-2- Composantes d'un robot mobile	
I-4- Conclusion.....	

CHAPITRE II : PARTIE MECANIQUE.

II.1. Introduction.....	
II.2. Structure générale du robot.....	
II.2.1. Le châssis.....	
II.2.2. La motorisation.....	
II.3. Conclusion	

CHAPITRE III : PARTIE ELECTRONIQUE.

III.1. Introduction.....	
III.2. Partie commande.....	
III.2.1. Eléments à contrôler.....	
III.2.2. Description fonctionnelle des cartes électroniques.....	

III.2.2.1. ARDUINO.....	
III.2.2.2. Le câble USB du ARDUINO.....	
III.3. Partie puissance.....	
III.3.1. La carte de puissance.....	
III.3.2. Le principe du pont en H.....	
III.3.3. La variation de vitesse (commande par PWM).....	
III.3.4. Intérêt du signal PWM.	
III.4. Perception-Capteurs.....	
III.4.1. Mesure de proximité et de distance.....	
III.4.1.1. Capteurs infrarouges.....	
III.4.1.1.1. Les capteurs utilisés « GP2D15 ».....	
III.4.1.1.2. Le « GP2D15 » en détecteur d'obstacle.....	
III.5 Conclusion.....	
CHAPITRE IV : ODOMETRIE.	
IV.1. Introduction.....	
IV.2. Principe de fonctionnement des encodeurs.....	
IV.3. Régulation en vitesse.....	
IV.4. Régulation en position.....	
IV.4.1. Calcul du déplacement en ligne droite.....	
IV.4.2. Calcule d'odométrie.....	
IV.5. Conclusion.....	
CHAPITRE V : IMPLEMENTATION SUR LE ROBOT ET RESULTATS.	
V.1. Introduction.....	
V.2. Communication avec les capteurs.....	
V.2.1. Evitement d'obstacle.....	
V.3. Résultats obtenu(organigramme des programme).....	
V.4. Conclusion.....	

CONCLUSION GENERALE	
BIBLIOGRAPHIE	
ANNEXES	

TABLE DES MATIERES

REMERCIEMENTS

DEDICACES

RESUME

ABSTRACT

ملخص

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

INTRODUCTION GENERALE

CHAPITRE I : Notions de robotique et de robots mobiles.

I-1- Introduction	
I-2- Historique	
I-3- La robotique mobile	
I-3-1- Autonomie en robotique mobile	
I-3-2- Composantes d'un robot mobile	
I-4- Conclusion.....	

CHAPITRE II : PARTIE MECANIQUE.

II.1. Introduction.....	
II.2. Structure générale du robot.....	
II.2.1. Le châssis.....	
II.2.2. La motorisation.....	
II.3. Conclusion	

CHAPITRE III : PARTIE ELECTRONIQUE.

III.1. Introduction.....	
III.2. Partie commande.....	
III.2.1. Eléments à contrôler.....	
III.2.2. Description fonctionnelle des cartes électroniques.....	

III.2.2.1. ARDUINO.....	
III.2.2.2. Le câble USB du ARDUINO.....	
III.3. Partie puissance.....	
III.3.1. La carte de puissance.....	
III.3.2. Le principe du pont en H.....	
III.3.3. La variation de vitesse (commande par PWM).....	
III.3.4. Intérêt du signal PWM.	
III.4. Perception-Capteurs.....	
III.4.1. Mesure de proximité et de distance.....	
III.4.1.1. Capteurs infrarouges.....	
III.4.1.1.1. Les capteurs utilisés « GP2D15 ».....	
III.4.1.1.2. Le « GP2D15 » en détecteur d'obstacle.....	
III.5 Conclusion.....	
CHAPITRE IV : ODOMETRIE.	
IV.1. Introduction.....	
IV.2. Principe de fonctionnement des encodeurs.....	
IV.3. Régulation en vitesse.....	
IV.4. Régulation en position.....	
IV.4.1. Calcul du déplacement en ligne droite.....	
IV.4.2. Calcule d'odométrie.....	
IV.5. Conclusion.....	
CHAPITRE V : IMPLEMENTATION SUR LE ROBOT ET RESULTATS.	
V.1. Introduction.....	
V.2. Communication avec les capteurs.....	
V.2.1. Evitement d'obstacle.....	
V.3. Résultats obtenu(organigramme des programme).....	
V.4. Conclusion.....	

CONCLUSION GENERALE	
BIBLIOGRAPHIE	
ANNEXES	

TABLE DES MATIERES

REMERCIEMENTS

DEDICACES

RESUME

ABSTRACT

ملخص

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

INTRODUCTION GENERALE

CHAPITRE I : Notions de robotique et de robots mobiles.

I-1- Introduction	
I-2- Historique	
I-3- La robotique mobile	
I-3-1- Autonomie en robotique mobile	
I-3-2- Composantes d'un robot mobile	
I-4- Conclusion.....	

CHAPITRE II : PARTIE MECANIQUE.

II.1. Introduction.....	
II.2. Structure générale du robot.....	
II.2.1. Le châssis.....	
II.2.2. La motorisation.....	
II.3. Conclusion	

CHAPITRE III : PARTIE ELECTRONIQUE.

III.1. Introduction.....	
III.2. Partie commande.....	
III.2.1. Eléments à contrôler.....	
III.2.2. Description fonctionnelle des cartes électroniques.....	

III.2.2.1. ARDUINO.....	
III.2.2.2. Le câble USB du ARDUINO.....	
III.3. Partie puissance.....	
III.3.1. La carte de puissance.....	
III.3.2. Le principe du pont en H.....	
III.3.3. La variation de vitesse (commande par PWM).....	
III.3.4. Intérêt du signal PWM.	
III.4. Perception-Capteurs.....	
III.4.1. Mesure de proximité et de distance.....	
III.4.1.1. Capteurs infrarouges.....	
III.4.1.1.1. Les capteurs utilisés « GP2D15 ».....	
III.4.1.1.2. Le « GP2D15 » en détecteur d'obstacle.....	
III.5 Conclusion.....	
CHAPITRE IV : ODOMETRIE.	
IV.1. Introduction.....	
IV.2. Principe de fonctionnement des encodeurs.....	
IV.3. Régulation en vitesse.....	
IV.4. Régulation en position.....	
IV.4.1. Calcul du déplacement en ligne droite.....	
IV.4.2. Calcule d'odométrie.....	
IV.5. Conclusion.....	
CHAPITRE V : IMPLEMENTATION SUR LE ROBOT ET RESULTATS.	
V.1. Introduction.....	
V.2. Communication avec les capteurs.....	
V.2.1. Evitement d'obstacle.....	
V.3. Résultats obtenu(organigramme des programme).....	
V.4. Conclusion.....	

CONCLUSION GENERALE
BIBLIOGRAPHIE
ANNEXES

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

FIGURE I.1. Robot marcheur.....	4
FIGURE I.2. Robot aérien.....	4
FIGURE I.3. Robot sous marin.....	4
FIGURE I.4. Exemples de robots mobiles rampant.....	5
FIGURE I.5. Schéma synoptique de principe d'un robot.....	6
FIGURE II.1. Le châssis.....	
FIGURE II.2. Roue du robot.....	
FIGURE II.3. Les roues-billes de robot	
FIGURE II.4. supporte de moteur	
FIGURE II .5.le moteur EMG30	
FIGURE III.1. Schéma synoptique des étages électroniques.....	
FIGURE III.2.La carte Arduino Duemilanove.....	
FIGURE III.3. Le câble USB du ARDUINO avec le module ARDUINO	
FIGURE III.4. Le câble USB du ARDUINO.	
FIGURE III.5. <i>Diagramme bloqué du L298</i>	
FIGURE III.6. <i>Tableau des entrées/sorties du pont en H</i>	
FIGURE III.7. <i>Brochage du circuit intégré L298</i>	
FIGURE III.8. Exemple de signaux PWM.....	
FIGURE III.9.Principe de mesure de distance.....	
FIGURE III.10 . Courbe d'hystérésis du GP2D15.....	
FIGURE III.11.Télémètre infrarouge GP2D15.....	
FIGURE IV.1. Les différents composants d'un encodeur rotatif.....	
FIGURE IV.2. L'encodeur en quadrature.....	
FIGURE IV.3. Couche de S neurones artificiels.....	36
FIGURE IV.4. Réseau neuronal multicouches.....	36
FIGURE IV.5. Apprentissage non supervisé.....	37
FIGURE IV.6. Apprentissage supervisé.....	38

FIGURE IV.7. Diagramme général des étapes de la modélisation	39
FIGURE IV.8. La méthode d'un seul RN	39
FIGURE IV.9. Organigramme de rétro-propagation du gradient	41
FIGURE IV.10. La représentation schématique du réseau	43
FIGURE IV.11. Les étapes principales de notre travail	44
FIGURE V.1. L'organigramme global	45
FIGURE V.2. L'organigramme de k-menas	46
FIGURE V.3. Image acquise par la webcam	47
FIGURE V.4. Image segmentée par la méthode de k-menas, avec $k=2$	47
FIGURE V.5. Le centre de gravité de l'objet à manipuler	48
FIGURE V.6. Localisation de l'objet dans le cas idéal	48
FIGURE V.7. Localisation de l'objet dans le cas réel	49
FIGURE V.8. L'organigramme de localisation et déplacement du système	49
TABLEAU II.1. Analogie: vision humaine-vision artificielle	10
TABLEAU III.1. Table de vérité des signaux de commande d'un moteur	29
TABLEAU IV.1. Analogie entre les modèles neuraux biologiques et artificiels.....	34
TABLEAU V.1. Table de vérité des commandes.....	50
TABLEAU V.2. La synchronisation des deux moteurs	51

Introduction générale

Le but de notre projet est la conception et la réalisation d'un système mobile, autonome, dont la fonction principale est d'accroître l'autonomie de déplacement d'un robot mobile. Cette autonomie a pour but de lui permettre de rejoindre un point quelconque de son environnement de travail tout en évitant les obstacles d'une part, et d'autre part de faire la modélisation dynamique de l'environnement, c'est-à-dire de pouvoir reconnaître les obstacles imprévus par le modèle et en même temps changer sa vitesse.

L'idée initiale de la réalisation des systèmes mobiles s'est développée dans le but de concevoir un système à caractéristiques humanoïdes, capable d'effectuer des corvées à la place de l'homme : un robot mobile. C'est ainsi qu'ont été développés tous les systèmes de localisation et de navigation dans un milieu donné.

Ces systèmes mobiles, qui permettent souvent de manipuler des objets, sont munis de différents types de capteurs tels que les capteurs infrarouges (IR), ultra-sons (us) et les capteurs tactiles et d'actionneurs, tous contrôlés par un processeur programmable.

Ils sont opérationnels dans un milieu structuré c'est-à-dire que tous les objets à manipuler (avec leurs dimensions, couleur, emplacement, etc..) sont préalablement prévus dans un programme, alors que si le système est soumis à un environnement partiellement variable ou totalement inconnu, il pourrait se trouver dans l'incapacité d'interpréter et donc de s'adapter de ces variations, et la détection des objets à manipuler devient difficile.

Pour minimiser les possibilités de déviation du système de sa trajectoire, nous avons établi un contrôle de vitesse des deux moteurs par un algorithme, qui va agir sur leur signal de commande : PWM.

Notre travail est détaillé dans ce mémoire qui est organisé comme suit :

Nous commençons par un rappel sur des notions de robotique et de robots mobiles. Dans le 2ème chapitre nous parlons sur la partie mécanique, Et dans le 3ème chapitre nous parlons sur la partie électronique. Dans le chapitre 4 nous parlons sur l'odométrie du robot, et le 5ème chapitre nous parlons sur l'implémentation sur le robot et les résultats et à la fin en terminant avec une conclusion générale.

Introduction générale

Le but de notre projet est la conception et la réalisation d'un système mobile, autonome, dont la fonction principale est d'accroître l'autonomie de déplacement d'un robot mobile. Cette autonomie a pour but de lui permettre de rejoindre un point quelconque de son environnement de travail tout en évitant les obstacles d'une part, et d'autre part de faire la modélisation dynamique de l'environnement, c'est-à-dire de pouvoir reconnaître les obstacles imprévus par le modèle et en même temps changer sa vitesse.

L'idée initiale de la réalisation des systèmes mobiles s'est développée dans le but de concevoir un système à caractéristiques humanoïdes, capable d'effectuer des corvées à la place de l'homme : un robot mobile. C'est ainsi qu'ont été développés tous les systèmes de localisation et de navigation dans un milieu donné.

Ces systèmes mobiles, qui permettent souvent de manipuler des objets, sont munis de différents types de capteurs tels que les capteurs infrarouges (IR), ultra-sons (us) et les capteurs tactiles et d'actionneurs, tous contrôlés par un processeur programmable.

Ils sont opérationnels dans un milieu structuré c'est-à-dire que tous les objets à manipuler (avec leurs dimensions, couleur, emplacement, etc..) sont préalablement prévus dans un programme, alors que si le système est soumis à un environnement partiellement variable ou totalement inconnu, il pourrait se trouver dans l'incapacité d'interpréter et donc de s'adapter de ces variations, et la détection des objets à manipuler devient difficile.

Pour minimiser les possibilités de déviation du système de sa trajectoire, nous avons établi un contrôle de vitesse des deux moteurs par un algorithme, qui va agir sur leur signal de commande : PWM.

Notre travail est détaillé dans ce mémoire qui est organisé comme suit :

Nous commençons par un rappel sur des notions de robotique et de robots mobiles. Dans le 2ème chapitre nous parlons sur la partie mécanique, Et dans le 3ème chapitre nous parlons sur la partie électronique. Dans le chapitre 4 nous parlons sur l'odométrie du robot, et le 5ème chapitre nous parlons sur l'implémentation sur le robot et les résultats et à la fin en terminant avec une conclusion générale.

Introduction générale

Le but de notre projet est la conception et la réalisation d'un système mobile, autonome, dont la fonction principale est d'accroître l'autonomie de déplacement d'un robot mobile. Cette autonomie a pour but de lui permettre de rejoindre un point quelconque de son environnement de travail tout en évitant les obstacles d'une part, et d'autre part de faire la modélisation dynamique de l'environnement, c'est-à-dire de pouvoir reconnaître les obstacles imprévus par le modèle et en même temps changer sa vitesse.

L'idée initiale de la réalisation des systèmes mobiles s'est développée dans le but de concevoir un système à caractéristiques humanoïdes, capable d'effectuer des corvées à la place de l'homme : un robot mobile. C'est ainsi qu'ont été développés tous les systèmes de localisation et de navigation dans un milieu donné.

Ces systèmes mobiles, qui permettent souvent de manipuler des objets, sont munis de différents types de capteurs tels que les capteurs infrarouges (IR), ultra-sons (us) et les capteurs tactiles et d'actionneurs, tous contrôlés par un processeur programmable.

Ils sont opérationnels dans un milieu structuré c'est-à-dire que tous les objets à manipuler (avec leurs dimensions, couleur, emplacement, etc..) sont préalablement prévus dans un programme, alors que si le système est soumis à un environnement partiellement variable ou totalement inconnu, il pourrait se trouver dans l'incapacité d'interpréter et donc de s'adapter de ces variations, et la détection des objets à manipuler devient difficile.

Pour minimiser les possibilités de déviation du système de sa trajectoire, nous avons établi un contrôle de vitesse des deux moteurs par un algorithme, qui va agir sur leur signal de commande : PWM.

Notre travail est détaillé dans ce mémoire qui est organisé comme suit :

Nous commençons par un rappel sur des notions de robotique et de robots mobiles. Dans le 2ème chapitre nous parlons sur la partie mécanique, Et dans le 3ème chapitre nous parlons sur la partie électronique. Dans le chapitre 4 nous parlons sur l'odométrie du robot, et le 5ème chapitre nous parlons sur l'implémentation sur le robot et les résultats et à la fin en terminant avec une conclusion générale.

I-1- Introduction :

Les robots sont capables de réaliser des tâches de plus en plus complexes, comme les robots géologiques envoyés sur Mars. Ces tâches de haut niveau reposent sur des capacités de base que doit avoir un robot : *se localiser dans un environnement et s'y déplacer correctement*; le travail du présent mémoire s'insère dans ce contexte.

Dans ce chapitre, nous présentons en premier lieu l'évolution de la robotique générale en introduisant la robotique mobile qui nous intéresse, puis nous détaillons un peu sa relation avec l'intelligence artificielle.

I-2- Historique :

La robotique fait aujourd'hui partie de notre quotidien. Nous la trouvons dans tous les domaines: la médecine, l'industrie, la recherche et dans différentes applications militaires.

La robotique industrielle est récente; elle date de la 2^{ème} moitié du XXe siècle. Pourtant, les dates importantes de son histoire se sont rapidement succédées 5 [1]:

1947: Première manipulation électrique télé-opérée, c'est-à-dire opérée à distance ;

1954 : Premier robot programmable ;

1961 : premier robot sur une chaîne de montage d'une usine GM (General Motors) ;

1961 à 1964 : Une confiserie de Kitchener, en Ontario, devient la première usine canadienne robotisée;

1963 : Première utilisation de la vision afin de contrôler un robot ;

Aujourd'hui, les concours de robots mobiles ont beaucoup de succès car ils permettent d'être créatif et d'apprendre beaucoup, tant du point de vue théorique que pratique.

I-3- La robotique mobile:

De manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition aux robots manipulateurs. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheur (figure I.1),

sous-marins (figure I.2) ou aériens (figure I.3).ou robots mobiles rampant (figure I.4)



Figure I.1. Robot marcheur



Figure I.2. Robot aérien



Figure I.3. Robot sous marin



Figure 1.4 Exemples de robots mobiles rampant

I-3-1- Autonomie en robotique mobile :

On cherche à réaliser des systèmes capables de réagir seuls à l'environnement, c'est-à-dire à un certain imprévu. C'est ce plus ou moins grand degré d'**autonomie** (permis par une **intelligence artificielle**) qui rapproche les robots des systèmes complètement autonomes.

Afin de l'être, un robot mobile doit posséder de nombreuses capacités :

- Premièrement, il doit être capable de percevoir son environnement et de se localiser dans celui-ci, sans l'intervention humaine, possédant des capteurs comme des sonars ou des dispositifs à balayage qui servent à mesurer des distances entre lui-même et les obstacles à proximité.
- Une fois localisé dans son environnement, le robot doit être capable de se déplacer d'un point à l'autre par des chemins efficaces et sécurisés, afin d'éviter des collisions avec des obstacles.

I-3-2- Composantes d'un robot mobile :

Le concept de ROBOT implique la faculté de **perception** de l'environnement, l'analyse de ces informations, la prise de **décision**, et enfin **l'action**: la mise en œuvre et la correction du comportement, en fonction des informations captées sur l'environnement. En cela, la perception de l'environnement constitue une fonction indispensable et caractéristique de tout système robotique.

On distingue dans un robot mobile: la partie commande qui traite toutes les informations, les actionneurs qui effectuent les actions et les capteurs qui informent le robot; comme l'indique le schéma synoptique suivant (figure I.4) :



Figure I.5. Schéma synoptique de principe d'un robot

- **Les capteurs:**

Ils ont pour fonction d'acquérir les données provenant de l'environnement. Ils peuvent à la fois informer le robot sur le milieu extérieur et le renseigner 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.

Les informations perçues ainsi que leur précision varient d'un capteur à un autre.

- **Les actionneurs:**

Pour bouger à l'intérieur de son environnement et y intégrer, un robot est muni d'actionneurs. A titre d'exemple, un robot mobile est muni d'un ou plusieurs moteurs pouvant faire tourner ses roues. La plupart des robots mobiles sont actionnés par des moteurs à courant continu.

- **Les modules (logiciels):**

Afin de faire fonctionner un robot mobile, plusieurs modules (logiciels) sont mis à contribution. Ces modules permettent d'interpréter les données perçues par les capteurs pour y extraire les informations ou à traiter des commandes et générer d'autres commandes. [3]

I-4- Conclusion :

La situation d'un robot mobile se déplaçant dans son environnement peut être comparée à un aveugle qui découvre un nouveau lieu. Le robot doit être capable de détecter et d'analyser un objet localement, puis de positionner globalement tous les objets détectés les uns par rapport aux autres. De cette manière, il construit un modèle de son environnement au fur et à mesure de son exploration.

I-1- Introduction :

Les robots sont capables de réaliser des tâches de plus en plus complexes, comme les robots géologiques envoyés sur Mars. Ces tâches de haut niveau reposent sur des capacités de base que doit avoir un robot : *se localiser dans un environnement et s'y déplacer correctement*; le travail du présent mémoire s'insère dans ce contexte.

Dans ce chapitre, nous présentons en premier lieu l'évolution de la robotique générale en introduisant la robotique mobile qui nous intéresse, puis nous détaillons un peu sa relation avec l'intelligence artificielle.

I-2- Historique :

La robotique fait aujourd'hui partie de notre quotidien. Nous la trouvons dans tous les domaines: la médecine, l'industrie, la recherche et dans différentes applications militaires.

La robotique industrielle est récente; elle date de la 2^{ème} moitié du XXe siècle. Pourtant, les dates importantes de son histoire se sont rapidement succédées 5 [1]:

1947: Première manipulation électrique télé-opérée, c'est-à-dire opérée à distance ;

1954 : Premier robot programmable ;

1961 : premier robot sur une chaîne de montage d'une usine GM (General Motors) ;

1961 à 1964 : Une confiserie de Kitchener, en Ontario, devient la première usine canadienne robotisée;

1963 : Première utilisation de la vision afin de contrôler un robot ;

Aujourd'hui, les concours de robots mobiles ont beaucoup de succès car ils permettent d'être créatif et d'apprendre beaucoup, tant du point de vue théorique que pratique.

I-3- La robotique mobile:

De manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition aux robots manipulateurs. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheur (figure I.1),

sous-marins (figure I.2) ou aériens (figure I.3).ou robots mobiles rampant (figure I.4)



Figure I.1. Robot marcheur



Figure I.2. Robot aérien



Figure I.3. Robot sous marin



Figure 1.4 Exemples de robots mobiles rampant

I-3-1- Autonomie en robotique mobile :

On cherche à réaliser des systèmes capables de réagir seuls à l'environnement, c'est-à-dire à un certain imprévu. C'est ce plus ou moins grand degré d'**autonomie** (permis par une **intelligence artificielle**) qui rapproche les robots des systèmes complètement autonomes.

Afin de l'être, un robot mobile doit posséder de nombreuses capacités :

- Premièrement, il doit être capable de percevoir son environnement et de se localiser dans celui-ci, sans l'intervention humaine, possédant des capteurs comme des sonars ou des dispositifs à balayage qui servent à mesurer des distances entre lui-même et les obstacles à proximité.
- Une fois localisé dans son environnement, le robot doit être capable de se déplacer d'un point à l'autre par des chemins efficaces et sécurisés, afin d'éviter des collisions avec des obstacles.

I-3-2- Composantes d'un robot mobile :

Le concept de ROBOT implique la faculté de **perception** de l'environnement, l'analyse de ces informations, la prise de **décision**, et enfin **l'action**: la mise en œuvre et la correction du comportement, en fonction des informations captées sur l'environnement. En cela, la perception de l'environnement constitue une fonction indispensable et caractéristique de tout système robotique.

On distingue dans un robot mobile: la partie commande qui traite toutes les informations, les actionneurs qui effectuent les actions et les capteurs qui informent le robot; comme l'indique le schéma synoptique suivant (figure I.4) :



Figure I.5. Schéma synoptique de principe d'un robot

- **Les capteurs:**

Ils ont pour fonction d'acquérir les données provenant de l'environnement. Ils peuvent à la fois informer le robot sur le milieu extérieur et le renseigner 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.

Les informations perçues ainsi que leur précision varient d'un capteur à un autre.

- **Les actionneurs:**

Pour bouger à l'intérieur de son environnement et y intégrer, un robot est muni d'actionneurs. A titre d'exemple, un robot mobile est muni d'un ou plusieurs moteurs pouvant faire tourner ses roues. La plupart des robots mobiles sont actionnés par des moteurs à courant continu.

- **Les modules (logiciels):**

Afin de faire fonctionner un robot mobile, plusieurs modules (logiciels) sont mis à contribution. Ces modules permettent d'interpréter les données perçues par les capteurs pour y extraire les informations ou à traiter des commandes et générer d'autres commandes. [3]

I-4- Conclusion :

La situation d'un robot mobile se déplaçant dans son environnement peut être comparée à un aveugle qui découvre un nouveau lieu. Le robot doit être capable de détecter et d'analyser un objet localement, puis de positionner globalement tous les objets détectés les uns par rapport aux autres. De cette manière, il construit un modèle de son environnement au fur et à mesure de son exploration.

I-1- Introduction :

Les robots sont capables de réaliser des tâches de plus en plus complexes, comme les robots géologiques envoyés sur Mars. Ces tâches de haut niveau reposent sur des capacités de base que doit avoir un robot : *se localiser dans un environnement et s'y déplacer correctement*; le travail du présent mémoire s'insère dans ce contexte.

Dans ce chapitre, nous présentons en premier lieu l'évolution de la robotique générale en introduisant la robotique mobile qui nous intéresse, puis nous détaillons un peu sa relation avec l'intelligence artificielle.

I-2- Historique :

La robotique fait aujourd'hui partie de notre quotidien. Nous la trouvons dans tous les domaines: la médecine, l'industrie, la recherche et dans différentes applications militaires.

La robotique industrielle est récente; elle date de la 2^{ème} moitié du XXe siècle. Pourtant, les dates importantes de son histoire se sont rapidement succédées 5 [1]:

1947: Première manipulation électrique télé-opérée, c'est-à-dire opérée à distance ;

1954 : Premier robot programmable ;

1961 : premier robot sur une chaîne de montage d'une usine GM (General Motors) ;

1961 à 1964 : Une confiserie de Kitchener, en Ontario, devient la première usine canadienne robotisée;

1963 : Première utilisation de la vision afin de contrôler un robot ;

Aujourd'hui, les concours de robots mobiles ont beaucoup de succès car ils permettent d'être créatif et d'apprendre beaucoup, tant du point de vue théorique que pratique.

I-3- La robotique mobile:

De manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition aux robots manipulateurs. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheur (figure I.1),

sous-marins (figure I.2) ou aériens (figure I.3).ou robots mobiles rampant (figure I.4)



Figure I.1. Robot marcheur



Figure I.2. Robot aérien



Figure I.3. Robot sous marin



Figure 1.4 Exemples de robots mobiles rampant

I-3-1- Autonomie en robotique mobile :

On cherche à réaliser des systèmes capables de réagir seuls à l'environnement, c'est-à-dire à un certain imprévu. C'est ce plus ou moins grand degré d'**autonomie** (permis par une **intelligence artificielle**) qui rapproche les robots des systèmes complètement autonomes.

Afin de l'être, un robot mobile doit posséder de nombreuses capacités :

- Premièrement, il doit être capable de percevoir son environnement et de se localiser dans celui-ci, sans l'intervention humaine, possédant des capteurs comme des sonars ou des dispositifs à balayage qui servent à mesurer des distances entre lui-même et les obstacles à proximité.
- Une fois localisé dans son environnement, le robot doit être capable de se déplacer d'un point à l'autre par des chemins efficaces et sécurisés, afin d'éviter des collisions avec des obstacles.

I-3-2- Composantes d'un robot mobile :

Le concept de ROBOT implique la faculté de **perception** de l'environnement, l'analyse de ces informations, la prise de **décision**, et enfin **l'action**: la mise en œuvre et la correction du comportement, en fonction des informations captées sur l'environnement. En cela, la perception de l'environnement constitue une fonction indispensable et caractéristique de tout système robotique.

On distingue dans un robot mobile: la partie commande qui traite toutes les informations, les actionneurs qui effectuent les actions et les capteurs qui informent le robot; comme l'indique le schéma synoptique suivant (figure I.4) :



Figure I.5. Schéma synoptique de principe d'un robot

- **Les capteurs:**

Ils ont pour fonction d'acquérir les données provenant de l'environnement. Ils peuvent à la fois informer le robot sur le milieu extérieur et le renseigner 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.

Les informations perçues ainsi que leur précision varient d'un capteur à un autre.

- **Les actionneurs:**

Pour bouger à l'intérieur de son environnement et y intégrer, un robot est muni d'actionneurs. A titre d'exemple, un robot mobile est muni d'un ou plusieurs moteurs pouvant faire tourner ses roues. La plupart des robots mobiles sont actionnés par des moteurs à courant continu.

- **Les modules (logiciels):**

Afin de faire fonctionner un robot mobile, plusieurs modules (logiciels) sont mis à contribution. Ces modules permettent d'interpréter les données perçues par les capteurs pour y extraire les informations ou à traiter des commandes et générer d'autres commandes. [3]

I-4- Conclusion :

La situation d'un robot mobile se déplaçant dans son environnement peut être comparée à un aveugle qui découvre un nouveau lieu. Le robot doit être capable de détecter et d'analyser un objet localement, puis de positionner globalement tous les objets détectés les uns par rapport aux autres. De cette manière, il construit un modèle de son environnement au fur et à mesure de son exploration.



FIGURE II.2. Roue du robot

➤ **La roue-bille :**

Le point d'appuis des roues n'étant pas suffisant pour assurer la stabilité et l'équilibre de notre robot.

Nous avons utilisé une roue-bille placée en avant.



FIGURE II.3.Les roues-billes de robot

Les supports des moteurs

Pour assuré une bon fixation des moteur on a utilise les supporte suivant :



FIGURE II .4.supporte de moteur

Motricité

Les moteurs utilisés sont des moteurs a courant continu (DC) avec réducteur de rapport 30 :1 équipé d'un encodeur de 360 impulsions/tour, de marque EMG30 d'une puissance de 6 w sous une tension de 12v. Il convient Parfaitement aux applications de robotique.



FIGURE II .5.le moteur EMG30

la structure finale du système :

Après avoir réalisé tous les systèmes mécaniques du robot, y compris les actionneurs nécessaires ainsi que leurs fonctions selon le cahier des charges vu précédemment.

II.3.Conclusion :

Dans ce chapitre nous avons vu les différentes étapes de la réalisation mécanique de notre robot et la définition des fonctions principales qui lui seront affectées et cela en utilisant les actionneurs adéquats.

Pour que tout ce mécanisme soit fonctionnel et contrôlable nous devons interfacer ces différents actionneurs avec un système électronique embarqué qui le commande, ce qu'on verra en détail dans le chapitre suivant.



FIGURE II.2. Roue du robot

➤ **La roue-bille :**

Le point d'appuis des roues n'étant pas suffisant pour assurer la stabilité et l'équilibre de notre robot.

Nous avons utilisé une roue-bille placée en avant.



FIGURE II.3.Les roues-billes de robot

Les supports des moteurs

Pour assuré une bon fixation des moteur on a utilise les supporte suivant :



FIGURE II .4.supporte de moteur

Motricité

Les moteurs utilisés sont des moteurs a courant continu (DC) avec réducteur de rapport 30 :1 équipé d'un encodeur de 360 impulsions/tour, de marque EMG30 d'une puissance de 6 w sous une tension de 12v. Il convient Parfaitement aux applications de robotique.



FIGURE II .5.le moteur EMG30

la structure finale du système :

Après avoir réalisé tous les systèmes mécaniques du robot, y compris les actionneurs nécessaires ainsi que leurs fonctions selon le cahier des charges vu précédemment.

II.3.Conclusion :

Dans ce chapitre nous avons vu les différentes étapes de la réalisation mécanique de notre robot et la définition des fonctions principales qui lui seront affectées et cela en utilisant les actionneurs adéquats.

Pour que tout ce mécanisme soit fonctionnel et contrôlable nous devons interfacer ces différents actionneurs avec un système électronique embarqué qui le commande, ce qu'on verra en détail dans le chapitre suivant.



FIGURE II.2. Roue du robot

➤ **La roue-bille :**

Le point d'appuis des roues n'étant pas suffisant pour assurer la stabilité et l'équilibre de notre robot.

Nous avons utilisé une roue-bille placée en avant.



FIGURE II.3.Les roues-billes de robot

Les supports des moteurs

Pour assuré une bon fixation des moteur on a utilise les supporte suivant :



FIGURE II .4.supporte de moteur

Motricité

Les moteurs utilisés sont des moteurs a courant continu (DC) avec réducteur de rapport 30 :1 équipé d'un encodeur de 360 impulsions/tour, de marque EMG30 d'une puissance de 6 w sous une tension de 12v. Il convient Parfaitement aux applications de robotique.



FIGURE II .5.le moteur EMG30

la structure finale du système :

Après avoir réalisé tous les systèmes mécaniques du robot, y compris les actionneurs nécessaires ainsi que leurs fonctions selon le cahier des charges vu précédemment.

II.3.Conclusion :

Dans ce chapitre nous avons vu les différentes étapes de la réalisation mécanique de notre robot et la définition des fonctions principales qui lui seront affectées et cela en utilisant les actionneurs adéquats.

Pour que tout ce mécanisme soit fonctionnel et contrôlable nous devons interfacer ces différents actionneurs avec un système électronique embarqué qui le commande, ce qu'on verra en détail dans le chapitre suivant.

III.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 d'accomplir ses tâches il est nécessaire de réaliser des cartes telles que la carte de commande, l'isolation galvanique et la carte de puissance, notre robot se compose de 2 cartes principales :

- Une carte mère qui est la carte de commande.

- Une carte de puissance qui comporte aussi un étage d'isolation galvanique permettant de commander 2 moteurs à courant continu.

Nous allons voir dans ce chapitre le fonctionnement de l'électronique ainsi que des capteurs et actionneurs utilisés.

III.2.Partie commande :

III.2.1.Eléments à contrôler :

- Moteurs de traction : servant au déplacement du robot.

- Microswitchs :

Pour se positionner droit en face des bordures .

- Télémètres infra rouges :

Pour la détection d'obstacle .

III.2.2.Description fonctionnelle des cartes électroniques :

La carte de commande contrôle directement les moteurs de traction et reçoit les signaux en provenance :

- * Des encodeurs des moteurs de locomotion.

- * Des capteurs infra rouges pour la détection d'obstacle.

- * Des microswitchs.

La carte de puissance (conçue autour du circuit L298) reçoit les signaux en provenance de la carte de commande pour commander les moteurs à courant continu.

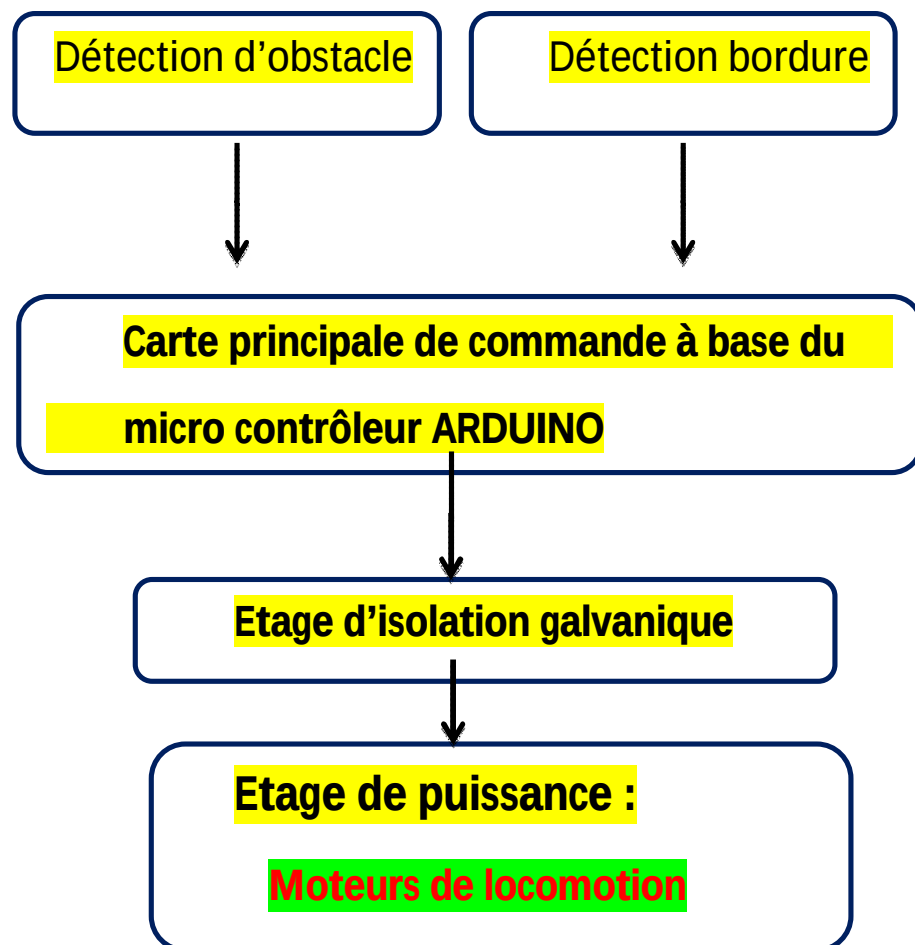


FIGURE III.1. Schéma synoptique des étages électronique

III.2.2.1.ARDUINO :



FIGURE III.2. La carte Arduino Duemilanove

Vue d'ensemble :

La carte Arduino Duemilanove("2009") est une carte à microcontrôleur basée sur l'ATmega168 pour les premières version ou sur l'ATmega328 pour les versions actuelles. Elle dispose :

de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),

de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),

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, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

"Duemilanove" signifie 2009 en Italien et ce nom est donné d'après l'année de sa sortie. La Duemilanove est la dernière née dans la série des cartes Arduino USB; pour une comparaison avec les cartes précédentes,

Alimentation :

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

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 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. [Aussi, la plage idéale recommandée pour alimenter la carte Duemilanove est entre 7V et 12V.](#)

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). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques

numériques nécessitent une tension d'alimentation parfaitement stable

dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

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) de la carte est disponible ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V).

L'intensité maximale disponible sur cette broche est de 50mA

GND. Broche de masse (ou 0V).

Mémoire :

L'ATmega 168 a 16 Ko de mémoire FLASH pour stocker le programme (dont 2ko sont utilisés par le bootloader); L'ATmega 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 2Ko

également utilisés par le bootloader). L'ATmega 168 a 1Ko de mémoire SRAM (volatile) et 512Ko d'EEPROM (non volatile) (mémoire qui peut être écrite et lue à l'aide de la librairie EEPROM?); L'ATmega 328 a 2ko de mémoire SRAM (volatile) et 1Ko d'EEPROM (non volatile).

Pour info : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port série, notamment lors de chaque programmation de la carte.

Entrées et sorties numériques :

Chacune des 14 broches numériques de la carte Duemilanove (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(), digitalWrite() et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite(broche, HIGH).

De plus, certaines broches ont des fonctions spécialisées :

Communication Serie: Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré FTDI de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Interruptions Externes: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction attachInterrupt() pour plus de détails.

Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite().

SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) qui, bien que disponible d'un point de vue matériel, n'est pas actuellement inclut dans le langage Arduino. Une librairie pour communication SPI est cependant disponible.

I2C: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils") .

LED: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte..

Broches analogiques :

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

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

Autres broches :

Il y a deux autres broches disponibles sur la carte

AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference().

Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

Communication :

La carte Arduino dispose de toute une série de facilité pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega 168 et l'ATmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour

communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré FT232RL (circuit FTDI) sur la carte assure la connexion

entre cette communication série vers le port USB de l'ordinateur et les drivers FTDI (qui sont inclus avec le logiciel Arduino) qui créent un port COM virtuel sur l'ordinateur utilisable par les logiciels.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré FTDI et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet la communication série sur n'importe quelle broche numérique de la carte Duemilanove.

L'ATmega 168 et l'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. Voir la documentation pour les détails Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

Programmation :

La carte Arduino Duemilanove peut être programmée avec le logiciel Arduino (à télécharger ici). Il suffit de sélectionner "Arduino Diecimila ou Duemilanove w/ ATmega168" ou "Arduino Duemilanove w/ ATmega328" dans le menu **Tools > Board** (en fonction du microcontrôleur présent sur votre carte). Pour plus de détails sur le langage Arduino, voir la référence du langage Arduino et pour apprendre à programmer en langage Arduino voir la page Apprendre.

Les microcontrôleurs ATmega168 ou ATmega328 présents sur la carte Arduino Duemilanove sont livrés avec un bootloader (petit programme de démarrage) préprogrammé qui vous permet de transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe. Ce bootloader communique avec le microcontrôleur en utilisant le protocole original STK500 (reference, fichiers C).

Vous pouvez bien sûr passer outre le bootloader et programmer le microcontrôleur via le connecteur ICSP (In-Circuit Serial Programming - "Programmation Série Dans le circuit" en français); voir ces instructions (en anglais) pour plus de détails.

Réinitialisation (logicielle) automatique :

Plutôt que de nécessiter un appui sur le bouton poussoir de réinitialisation avant un transfert de programme, la carte Arduino Duemilanove a été conçue de telle façon qu'elle puisse être réinitialisée par un logiciel tournant sur l'ordinateur. Une des broches matérielles de contrôle du flux (DTR) du

circuit intégré FT232RL est connecté à la ligne de réinitialisation de l'ATmega (168 ou 328) via un condensateur de 100 nanofarads. Lorsque cette broche est mise au niveau BAS, la broche de réinitialisation s'abaisse suffisamment longtemps pour réinitialiser le microcontrôleur. Le logiciel

Arduino utilise cette possibilité pour vous permettre de transférer votre programme dans la carte par un simple clic sur le bouton de transfert de la barre de boutons de l'environnement Arduino. Cela signifie que le bootloader peut avoir un temps mort plus court, la mise au niveau bas de la broche DTR étant bien coordonnée avec le début du transfert du programme.

Ce fonctionnement a d'autres implications. Quand la carte Duemilanove est connectée à un ordinateur fonctionnant soit sous Mac Os X ou Linux, la carte se réinitialise à chaque fois qu'une connexion se fait entre elle et le logiciel (via le port USB). Pendant la demi-seconde suivante, le bootloader s'exécute sur la carte Duemilanove. Puisqu'il est programmé pour ignorer les données mal formatées (càd toute donnée en plus du transfert d'un nouveau programme), il interceptera les quelques premiers octets de donnée envoyé à la carte après qu'une connexion soit établie. Si un programme s'exécutant sur la carte reçoit une configuration initiale

ou d'autres données quand il démarre, assurez-vous que le logiciel avec lequel il communique attend une seconde après l'ouverture de la connexion avant d'envoyer ces données.

La carte Duemilanove comporte une piste (du circuit imprimé) que vous pouvez couper pour désactiver la réinitialisation automatique. Les bords de chaque côté de la piste peuvent être solidarisés ensemble pour la réactiver. Vous pouvez également désactiver la réinitialisation automatique en connectant une résistance de 110 Ohms entre le +5V et la ligne de réinitialisation (reset); voir [ce fil du forum](#) (en anglais) pour plus de détails. **Protection du port USB contre la surcharge en intensité**

La carte Arduino Duemilanove a un polyfusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

Caractéristiques Mécaniques :

Les longueurs et largeurs maximales de la Duemilanove sont respectivement 6.86 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation Jack s'étendant au-delà des dimensions de la carte. Trois trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier. Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.

Ecouter le nom :

Voici comment vous devez prononcer le nom de la carte en italien correct : téléchargez le fichier sonore dans le format que vous préférez : [WAV](#), [OGG](#), [MP3](#), [FLAC](#), [WMA](#).

Commentaires utilisateurs :

Le bootloader est un programme préprogrammé une fois pour toute dans l'ATmega et qui permet la communication entre l'ATmega et le logiciel Arduino via le port série, notamment lors de chaque programmation de la carte.

La mémoire disponible avec l'ATMega328 est largement suffisante pour les programmes. A titre d'exemple, un programme simple utilisant clavier et afficheur LCD avec calculs mathématiques n'occupera que 5 ou 6K de mémoire, ce qui laisse de la marge...

Pour ceux qui auraient une carte Arduino avec un ATmega 168 et qui voudraient passer à l'ATmega 328, ce dernier est vendu pré-programmé avec le bootloader pour quelques euros.

En pratique, la programmation par un programmeur externe, bien que possible comme évoqué ci-dessus n'a pas grand intérêt puisque tout l'avantage du système Arduino est justement de pouvoir programmer la carte Arduino directement par le port USB, sans programmeur externe. Le bootloader ne faisant que 2Ko, c'est 30 Ko de programme qui sont disponibles avec l'ATmega 328, ce qui est très largement suffisant dans la plupart des cas.

III.2.2.2. Le câble USB du ARDUINO :

Ce câble nous permettra de télécharger nos programmes (tâches) au sein du module ARDUINO et de déboguer ces derniers.

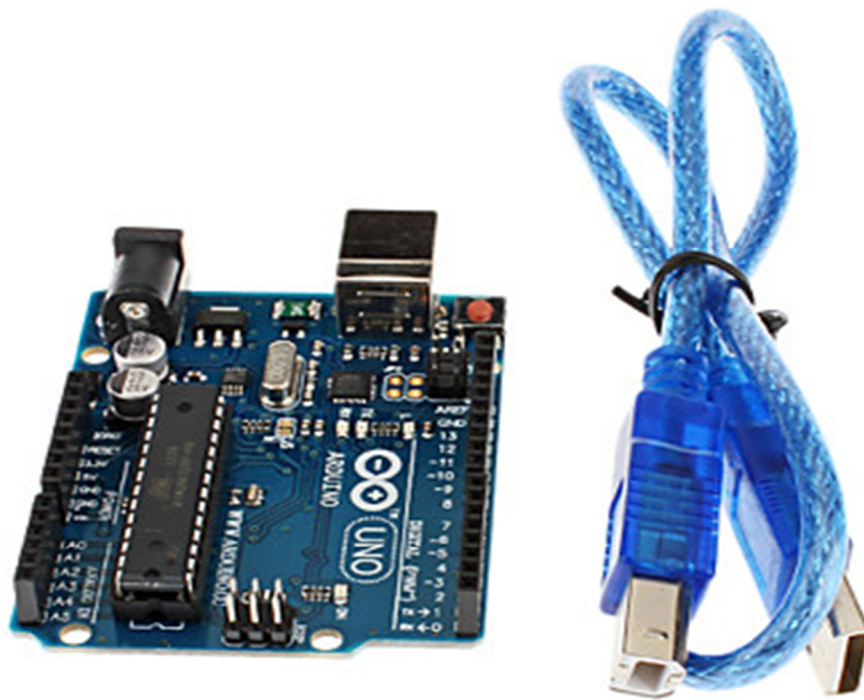


FIGURE III.3. Le câble USB du ARDUINO avec le module ARDUINO



FIGURE III.4. Le câble USB du ARDUINO.

III.3.Partie puissance :

III.3.1.La carte de puissance :

La carte de puissance joue le rôle d'une carte d'interface entre la carte de commande et la partie motrice comportant les composants mécaniques, elle nous permet de commander les moteurs à courant continu suivant les ordres fournis par la carte de commande.

Cette carte comporte également un étage d'isolation galvanique à base de circuit intégré TLP 521 afin de rendre les 2 parties totalement indépendantes et éviter tout phénomène de destruction par électricité statique.

Etant donné que notre robot contient 2 moteurs à courant continu, on a réalisé une carte qui comporte 2 sorties moteurs.

III.3.2. Le principe du pont en H :

Dans notre montage, nous avons besoin de quatre ponts en H, c'est pour cette raison que nous avons utilisées deux circuits intégrés L298, car chaque circuit offre un double pont en H.

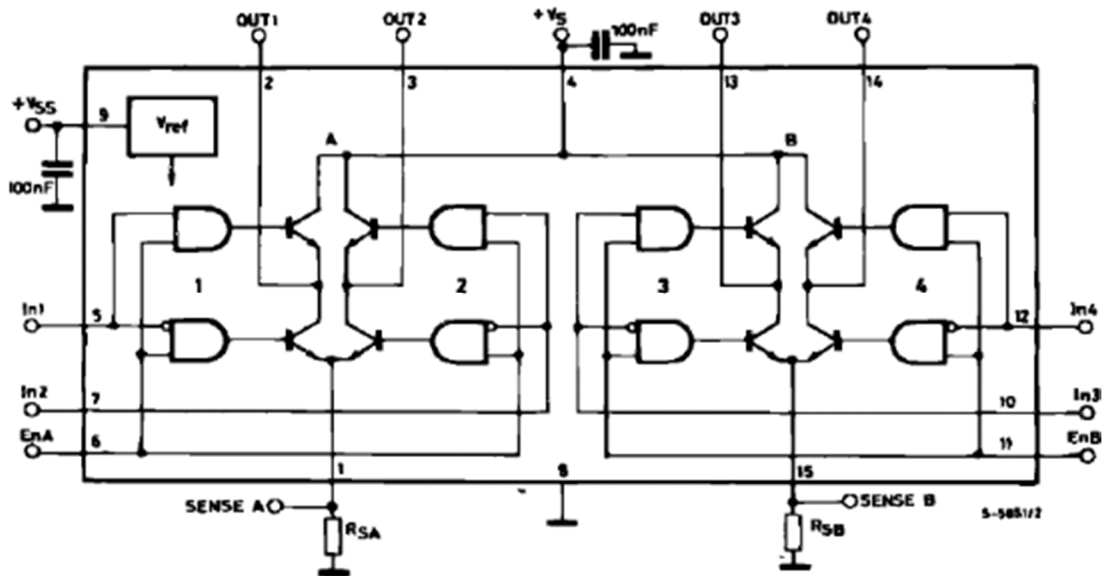


FIGURE III.5. Diagramme bloqué du L298.

Les caractéristiques du L298 sont les suivantes :

- Intensité maximale : 2A par pont ;
- Alimentation de puissance de 5.5V à 50V
- Type de boîtier : Multiwatt15 ;
- Dissipation puissance total : 25w
- Trois entrées par pont : In1, In2 et ENABLE.

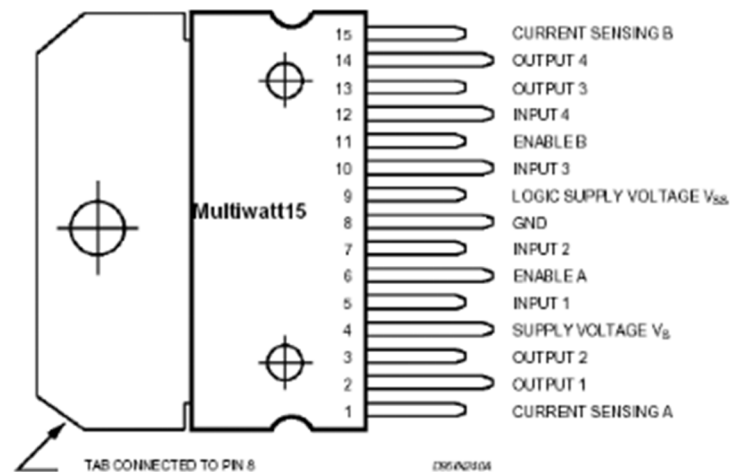


FIGURE III.7. Brochage du circuit intégré L298.

III.3.3. La variation de vitesse (commande par PWM) :

Quand on veut faire varier la vitesse d'un moteur, la première idée qui vient à l'esprit est de faire varier la tension aux bornes du moteur, mais on constate que pour des valeurs faibles de la tension,

le moteur ne tourne pas. Le moteur demande une tension minimale assez importante pour démarrer. Si cette dernière est trop basse, les forces électromagnétiques ne sont pas suffisantes pour combattre le frottement. Il devient donc difficile d'ajuster la vitesse de façon précise.

La solution à ce problème est astucieuse. Il suffit de fournir au moteur une tension qui est toujours la même soit la tension maximale ! Par contre, cette tension ne sera appliquée que par très courtes périodes de temps. En ajustant la longueur de ces périodes de temps, on arrive à faire avancer plus ou moins vite les moteurs. Mieux, on remarque que la vitesse des moteurs devient proportionnelle à la longueur des périodes de temps. Contrôler la longueur des périodes passées à la tension maximale par rapport au temps passé sans application de tension (tension nulle) est donc le cœur de la solution. En réalité, cette solution est fort connue en contrôle des systèmes et en électronique et elle porte le nom de PWM (Pulse Width Modulation) ou MLI pour Modulation en Largeur d'Impulsion.

Un signal PWM est un signal dont la période est fixe, mais le rapport cyclique varie. En d'autres termes, t_1 et t_2 varient tout en conservant $t_1 + t_2 = T = \text{constante}$.

III.3.4. Intérêt du signal PWM :

Le principal intérêt de la technique PWM est de limiter la chauffe des composants électroniques.

Par exemple : une lampe de 20 Watts allumée au maximum consomme 20 Watt.

Si par une commande de gradation elle est allumée au quart de sa puissance,

la puissance fournie est soit maximale, soit nulle.

Lorsqu'elle est maximale, pendant un quart du temps par exemple, il n'y a pas besoin de dissiper de puissance résiduelle. Lorsqu'elle est nulle, il n'y a pas besoin de dissiper non plus de puissance, car elle n'est pas fournie du tout...

Un autre intérêt de la PWM est que la tension appliquée au moteur pendant t_1 est V_{cc} .

Celle-ci est suffisante pour vaincre les frottements et faire tourner le moteur. Mais la tension moyenne appliquée au moteur est proportionnelle au rapport cyclique, ce qui permet d'avoir des consignes de vitesse faibles : $V_{MOY} = t_1 \cdot V_{cc} / T$.

La commande d'actionneurs de puissance par PWM est très liée à la notion de fréquence. Pour que l'impression d'une valeur moyenne constante d'allumage apparaisse, il faut que l'alternance d'allumage/extinction soit suffisamment rapide pour qu'elle ne se remarque pas.

Selon les utilisations la fréquence de PWM va de 100 Hz à 200 kHz.

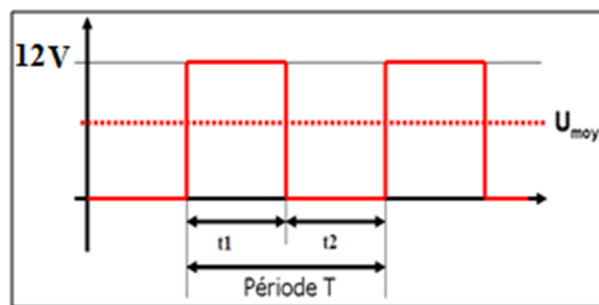


FIGURE III.8. Exemple de signaux PWM.

III.4. Perception-Capteurs :

III.4.1.Mesure de proximité et de distance :

On appelle télémétrie toute technique de mesure de distance par des procédés acoustiques, optiques ou radioélectriques.

L'appareil permettant de mesurer les distances est appelé télémètre.

De même qu'il existe différentes techniques de mesure de distance (mesure du temps de vol d'une onde, triangulation), il existe différentes technologies pour réaliser des télémètres.

Nous présentons ici les plus répandues en robotique mobile, en donnant une idée de leur gamme de mesure et d'application. Tous les capteurs télémétriques, basés sur des mesures de l'environnement, sont bien évidemment actifs.

III.4.1.1.Capteurs infrarouges :

Les détecteurs photoélectriques autrement appelés capteurs opto-électriques, se composent essentiellement d'un émetteur de lumière associé à un récepteur photosensible.

Ils sont schématisés de la manière suivante :

Emetteur et récepteur sont regroupés dans un même boîtier. La portée dépend du pouvoir réfléchissant et de la dimension de l'objet. En l'absence de l'objet, le faisceau n'est pas renvoyé.

En présence de l'objet, le faisceau est renvoyé : il y a alors détection.

Les capteurs de la famille GP2DXX, que nous avons utilisés pour le projet envoient une impulsion lumineuse dans l'air. Si elle est renvoyée par un obstacle dans un certain laps de temps donné, le faisceau parvient au récepteur formant un triangle. Le récepteur, donne l'information que l'objet est proche et renvoie un niveau logique correspondant. Dans la famille, d'autres capteurs mesurent l'angle formé par le capteur avec le faisceau, déterminant précisément la distance à l'objet.

En sortie, le signal est donc analogique et doit être mis en forme, pour être traité.

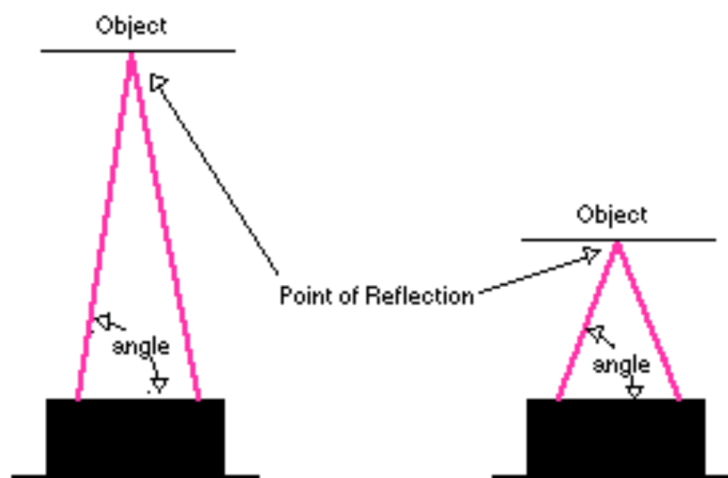


FIGURE III.9.Principe de mesure de distance.

III.4.1.1.1.Les capteurs utilisés « GP2D15 et GP2D150» :

Notre robot est équipé de capteurs photoélectriques « numériques » GP2D15, qui ont la particularité de posséder un seuil de détection situé aux alentours de 24 cm. En effet un état haut apparaît lorsque la distance est inférieure ou égale à 24 cm. Le GP2D15 a un seuil à hystérésis.

Ce capteur -contrairement aux capteurs analogiques- a la particularité d'être insensible à la lumière, c'est-à-dire, fonctionnant dans la lumière ou dans l'obscurité, il renvoie le même signal de sortie V_{out} .

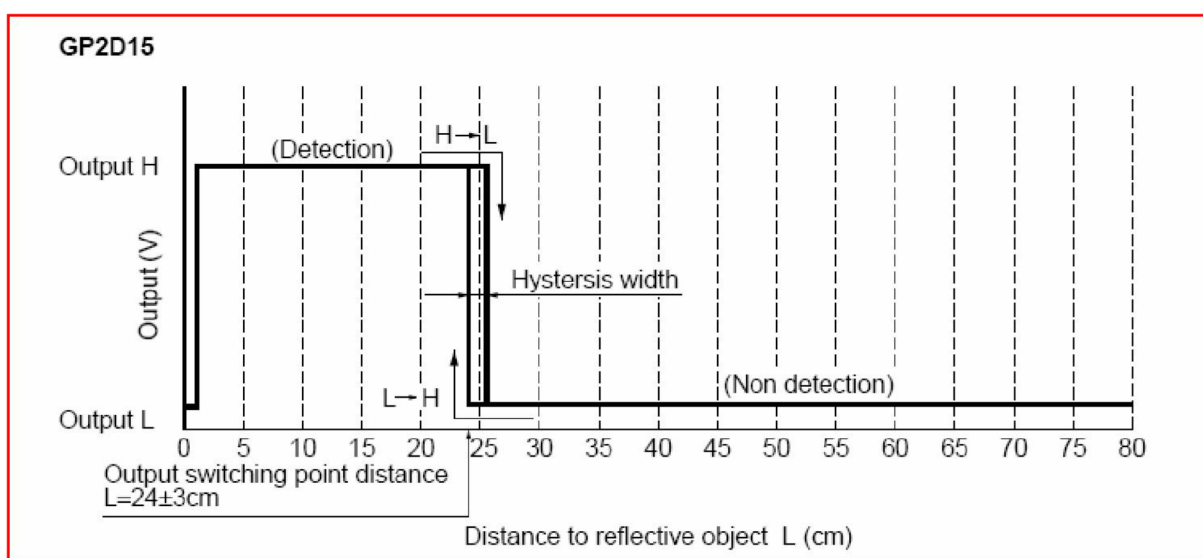


FIGURE III.10 . Courbe d'hystérésis du GP2D15.

III.4.1.1.2. Le « GP2D15 » en détecteur d'obstacle:

Le robot doit être équipé d'un système d'évitement d'obstacle pour éviter le robot adverse et assurer sa trajectoire vers la zone de construction.

Ceci peut se faire en fixant un capteur GP2D15 à droite et un autre à gauche, une routine d'évitement d'obstacle s'active dès que le robot franchit le seuil de déclenchement du capteur (24 cm).

Ce capteur a l'avantage d'avoir une distance minimale de 0 cm, c'est-à-dire qu'il se déclenchera à n'importe quelle distance inférieure à 24 cm s'il rencontre un obstacle, contrairement à d'autres capteurs qui ont une distance minimale de 4 jusqu'à 20 cm.

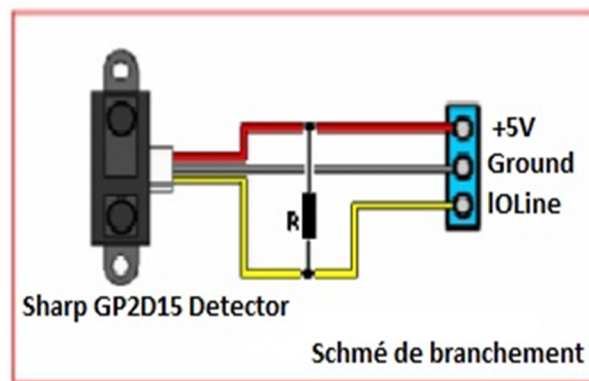


FIGURE III.11. Télémètre infrarouge GP2D15.

III.5. Conclusion :

Ce chapitre nous a permis de voir le cœur de notre robot, les actionneurs chargés des mouvements de construction ainsi que les différents composants qui communiquent avec l'extérieur (les capteurs).

Notre but dans ce travail étant de bien contrôler le déplacement du robot, nous aurons

donc besoin d'exploiter et gérer les signaux reçus des encodeurs, c'est ce que nous verrons dans le chapitre suivant concernant l'odométrie.

III.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 d'accomplir ses tâches il est nécessaire de réaliser des cartes telles que la carte de commande, l'isolation galvanique et la carte de puissance, notre robot se compose de 2 cartes principales :

- Une carte mère qui est la carte de commande.

- Une carte de puissance qui comporte aussi un étage d'isolation galvanique permettant de commander 2 moteurs à courant continu.

Nous allons voir dans ce chapitre le fonctionnement de l'électronique ainsi que des capteurs et actionneurs utilisés.

III.2.Partie commande :

III.2.1.Eléments à contrôler :

- Moteurs de traction : servant au déplacement du robot.

- Microswitchs :

Pour se positionner droit en face des bordures .

- Télémètres infra rouges :

Pour la détection d'obstacle .

III.2.2.Description fonctionnelle des cartes électroniques :

La carte de commande contrôle directement les moteurs de traction et reçoit les signaux en provenance :

- * Des encodeurs des moteurs de locomotion.

- * Des capteurs infra rouges pour la détection d'obstacle.

- * Des microswitchs.

La carte de puissance (conçue autour du circuit L298) reçoit les signaux en provenance de la carte de commande pour commander les moteurs à courant continu.

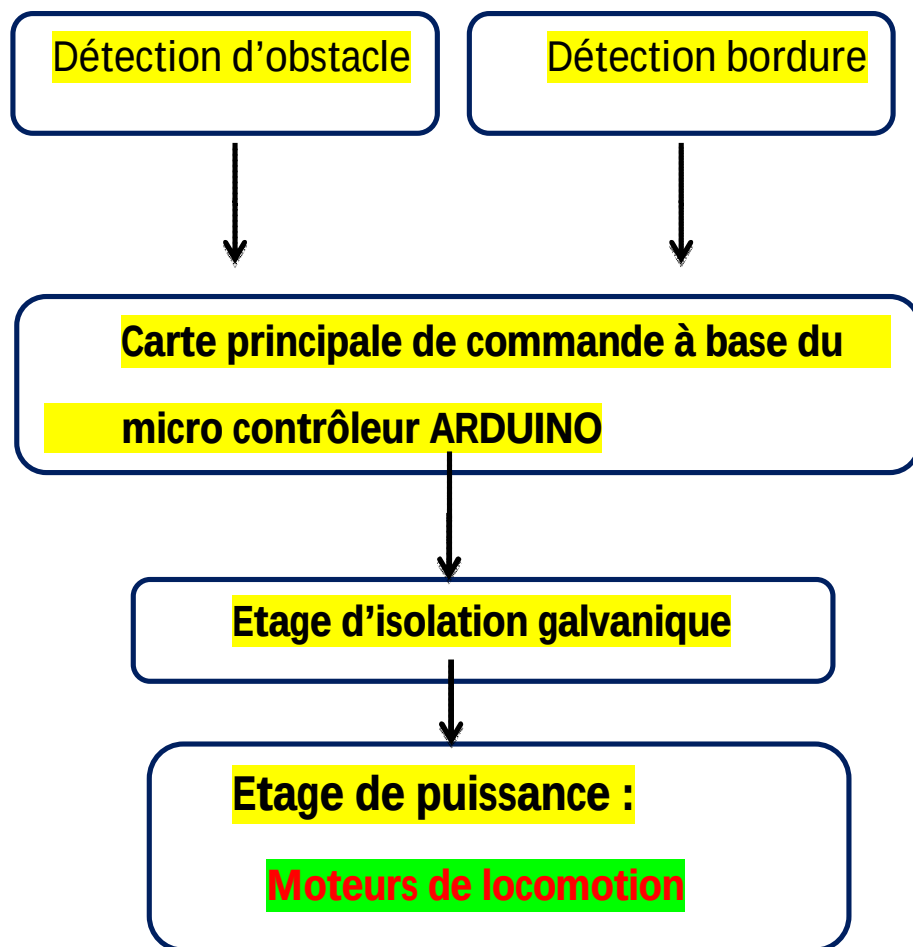


FIGURE III.1. Schéma synoptique des étages électronique

III.2.2.1.ARDUINO :



FIGURE III.2. La carte Arduino Duemilanove

Vue d'ensemble :

La carte Arduino Duemilanove("2009") est une carte à microcontrôleur basée sur l'ATmega168 pour les premières version ou sur l'ATmega328 pour les versions actuelles. Elle dispose :

de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),

de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),

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, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

"Duemilanove" signifie 2009 en Italien et ce nom est donné d'après l'année de sa sortie. La Duemilanove est la dernière née dans la série des cartes Arduino USB; pour une comparaison avec les cartes précédentes,

Alimentation :

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

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 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. [Aussi, la plage idéale recommandée pour alimenter la carte Duemilanove est entre 7V et 12V.](#)

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). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques

numériques nécessitent une tension d'alimentation parfaitement stable

dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

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) de la carte est disponible ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V).

L'intensité maximale disponible sur cette broche est de 50mA

GND. Broche de masse (ou 0V).

Mémoire :

L'ATmega 168 a 16 Ko de mémoire FLASH pour stocker le programme (dont 2ko sont utilisés par le bootloader); L'ATmega 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 2Ko

également utilisés par le bootloader). L'ATmega 168 a 1Ko de mémoire SRAM (volatile) et 512Ko d'EEPROM (non volatile) (mémoire qui peut être écrite et lue à l'aide de la librairie EEPROM?); L'ATmega 328 a 2ko de mémoire SRAM (volatile) et 1Ko d'EEPROM (non volatile).

Pour info : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port série, notamment lors de chaque programmation de la carte.

Entrées et sorties numériques :

Chacune des 14 broches numériques de la carte Duemilanove (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(), digitalWrite() et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite(broche, HIGH).

De plus, certaines broches ont des fonctions spécialisées :

Communication Serie: Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré FTDI de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Interruptions Externes: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction [attachInterrupt\(\)](#) pour plus de détails.

Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction [analogWrite\(\)](#).

SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) qui, bien que disponible d'un point de vue matériel, n'est pas actuellement inclut dans le langage Arduino. Une [bibliothèque pour communication SPI](#) est cependant disponible.

I2C: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant [la bibliothèque Wire/I2C \(ou TWI - Two-Wire interface - interface "2 fils"\)](#) .

LED: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte..

Broches analogiques :

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

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

Autres broches :

Il y a deux autres broches disponibles sur la carte

AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction [analogReference\(\)](#).

Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

Communication :

La carte Arduino dispose de toute une série de facilité pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega 168 et l'ATmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour

communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré FT232RL (circuit FTDI) sur la carte assure la connexion

entre cette communication série vers le port USB de l'ordinateur et les drivers FTDI (qui sont inclus avec le logiciel Arduino) qui créent un port COM virtuel sur l'ordinateur utilisable par les logiciels.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré FTDI et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet la communication série sur n'importe quelle broche numérique de la carte Duemilanove.

L'ATmega 168 et l'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. Voir la documentation pour les détails Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

Programmation :

La carte Arduino Duemilanove peut être programmée avec le logiciel Arduino (à télécharger ici). Il suffit de sélectionner "Arduino Diecimila ou Duemilanove w/ ATmega168" ou "Arduino Duemilanove w/ ATmega328" dans le menu **Tools > Board** (en fonction du microcontrôleur présent sur votre carte). Pour plus de détails sur le langage Arduino, voir la référence du langage Arduino et pour apprendre à programmer en langage Arduino voir la page Apprendre.

Les microcontrôleurs ATmega168 ou ATmega328 présents sur la carte Arduino Duemilanove sont livrés avec un bootloader (petit programme de démarrage) préprogrammé qui vous permet de transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe. Ce bootloader communique avec le microcontrôleur en utilisant le protocole original STK500 (reference, fichiers C).

Vous pouvez bien sûr passer outre le bootloader et programmer le microcontrôleur via le connecteur ICSP (In-Circuit Serial Programming - "Programmation Série Dans le circuit" en français); voir ces instructions (en anglais) pour plus de détails.

Réinitialisation (logicielle) automatique :

Plutôt que de nécessiter un appui sur le bouton poussoir de réinitialisation avant un transfert de programme, la carte Arduino Duemilanove a été conçue de telle façon qu'elle puisse être réinitialisée par un logiciel tournant sur l'ordinateur. Une des broches matérielles de contrôle du flux (DTR) du

circuit intégré FT232RL est connecté à la ligne de réinitialisation de l'ATmega (168 ou 328) via un condensateur de 100 nanofarads. Lorsque cette broche est mise au niveau BAS, la broche de réinitialisation s'abaisse suffisamment longtemps pour réinitialiser le microcontrôleur. Le logiciel

Arduino utilise cette possibilité pour vous permettre de transférer votre programme dans la carte par un simple clic sur le bouton de transfert de la barre de boutons de l'environnement Arduino. Cela signifie que le bootloader peut avoir un temps mort plus court, la mise au niveau bas de la broche DTR étant bien coordonnée avec le début du transfert du programme.

Ce fonctionnement a d'autres implications. Quand la carte Duemilanove est connectée à un ordinateur fonctionnant soit sous Mac Os X ou Linux, la carte se réinitialise à chaque fois qu'une connexion se fait entre elle et le logiciel (via le port USB). Pendant la demi-seconde suivante, le bootloader s'exécute sur la carte Duemilanove. Puisqu'il est programmé pour ignorer les données mal formatées (càd toute donnée en plus du transfert d'un nouveau programme), il interceptera les quelques premiers octets de donnée envoyé à la carte après qu'une connexion soit établie. Si un programme s'exécutant sur la carte reçoit une configuration initiale

ou d'autres données quand il démarre, assurez-vous que le logiciel avec lequel il communique attend une seconde après l'ouverture de la connexion avant d'envoyer ces données.

La carte Duemilanove comporte une piste (du circuit imprimé) que vous pouvez couper pour désactiver la réinitialisation automatique. Les bords de chaque côté de la piste peuvent être solidarisés ensemble pour la réactiver. Vous pouvez également désactiver la réinitialisation automatique en connectant une résistance de 110 Ohms entre le +5V et la ligne de réinitialisation (reset); voir [ce fil du forum](#) (en anglais) pour plus de détails. **Protection du port USB contre la surcharge en intensité**

La carte Arduino Duemilanove a un polyfusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

Caractéristiques Mécaniques :

Les longueurs et largeurs maximales de la Duemilanove sont respectivement 6.86 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation Jack s'étendant au-delà des dimensions de la carte. Trois trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier. Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.

Ecouter le nom :

Voici comment vous devez prononcer le nom de la carte en italien correct : téléchargez le fichier sonore dans le format que vous préférez : [WAV](#), [OGG](#), [MP3](#), [FLAC](#), [WMA](#).

Commentaires utilisateurs :

Le bootloader est un programme préprogrammé une fois pour toute dans l'ATmega et qui permet la communication entre l'ATmega et le logiciel Arduino via le port série, notamment lors de chaque programmation de la carte.

La mémoire disponible avec l'ATMega328 est largement suffisante pour les programmes. A titre d'exemple, un programme simple utilisant clavier et afficheur LCD avec calculs mathématiques n'occupera que 5 ou 6K de mémoire, ce qui laisse de la marge...

Pour ceux qui auraient une carte Arduino avec un ATmega 168 et qui voudraient passer à l'ATmega 328, ce dernier est vendu pré-programmé avec le bootloader pour quelques euros.

En pratique, la programmation par un programmeur externe, bien que possible comme évoqué ci-dessus n'a pas grand intérêt puisque tout l'avantage du système Arduino est justement de pouvoir programmer la carte Arduino directement par le port USB, sans programmeur externe. Le bootloader ne faisant que 2Ko, c'est 30 Ko de programme qui sont disponibles avec l'ATmega 328, ce qui est très largement suffisant dans la plupart des cas.

III.2.2.2. Le câble USB du ARDUINO :

Ce câble nous permettra de télécharger nos programmes (tâches) au sein du module ARDUINO et de déboguer ces derniers.

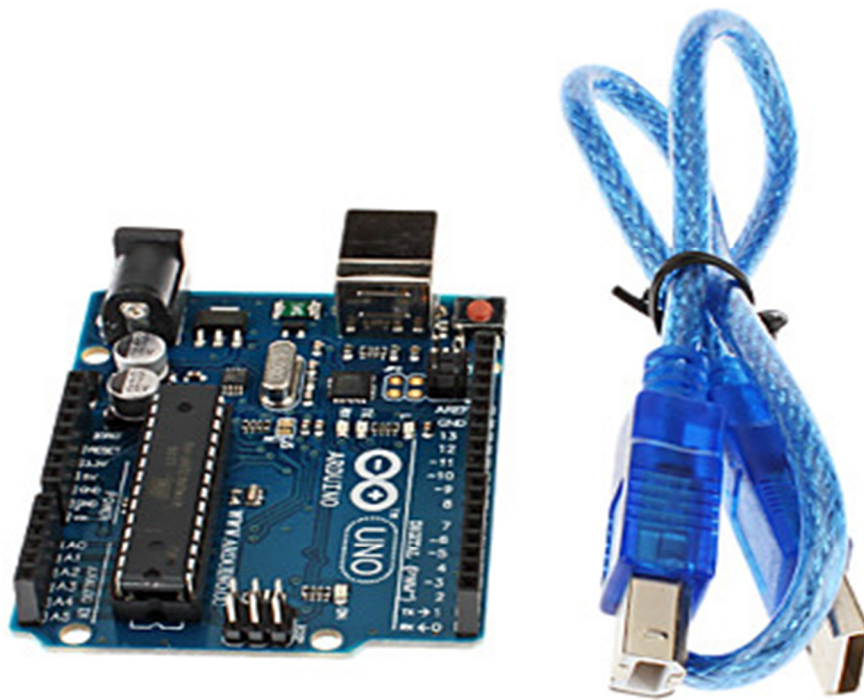


FIGURE III.3. Le câble USB du ARDUINO avec le module ARDUINO



FIGURE III.4. Le câble USB du ARDUINO.

III.3.Partie puissance :

III.3.1.La carte de puissance :

La carte de puissance joue le rôle d'une carte d'interface entre la carte de commande et la partie motrice comportant les composants mécaniques, elle nous permet de commander les moteurs à courant continu suivant les ordres fournis par la carte de commande.

Cette carte comporte également un étage d'isolation galvanique à base de circuit intégré TLP 521 afin de rendre les 2 parties totalement indépendantes et éviter tout phénomène de destruction par électricité statique.

Etant donné que notre robot contient 2 moteurs à courant continu, on a réalisé une carte qui comporte 2 sorties moteurs.

III.3.2. Le principe du pont en H :

Dans notre montage, nous avons besoin de quatre ponts en H, c'est pour cette raison que nous avons utilisées deux circuits intégrés L298, car chaque circuit offre un double pont en H.

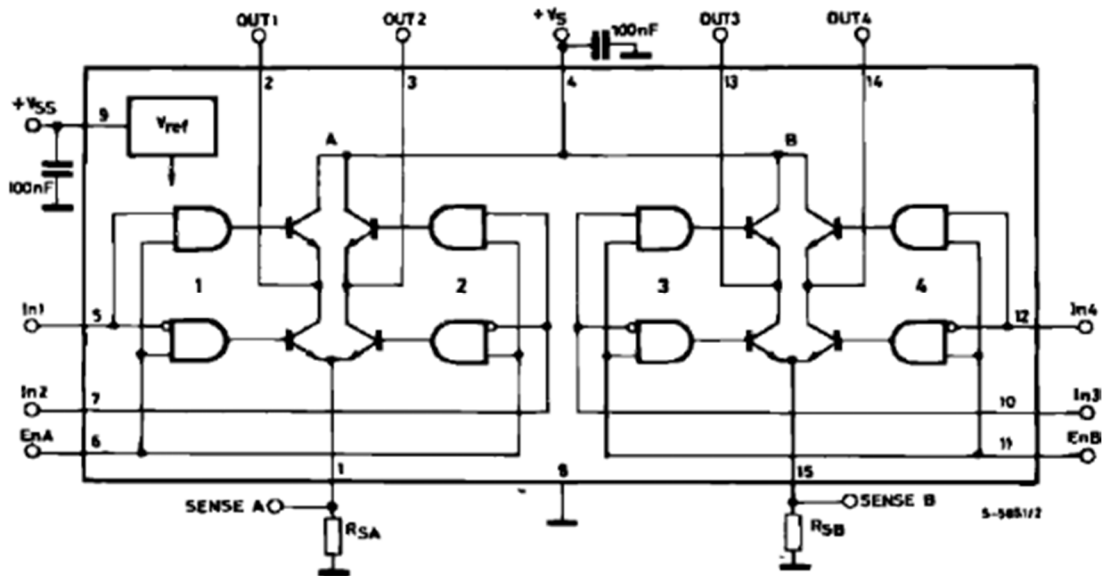


FIGURE III.5. Diagramme bloqué du L298.

Les caractéristiques du L298 sont les suivantes :

- Intensité maximale : 2A par pont ;
- Alimentation de puissance de 5.5V à 50V
- Type de boîtier : Multiwatt15 ;
- Dissipation puissance total : 25w
- Trois entrées par pont : In1, In2 et ENABLE.

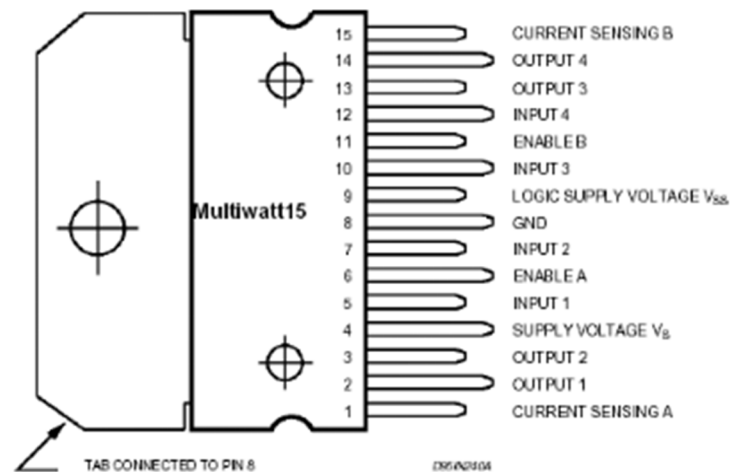


FIGURE III.7. Brochage du circuit intégré L298.

III.3.3. La variation de vitesse (commande par PWM) :

Quand on veut faire varier la vitesse d'un moteur, la première idée qui vient à l'esprit est de faire varier la tension aux bornes du moteur, mais on constate que pour des valeurs faibles de la tension,

le moteur ne tourne pas. Le moteur demande une tension minimale assez importante pour démarrer. Si cette dernière est trop basse, les forces électromagnétiques ne sont pas suffisantes pour combattre le frottement. Il devient donc difficile d'ajuster la vitesse de façon précise.

La solution à ce problème est astucieuse. Il suffit de fournir au moteur une tension qui est toujours la même soit la tension maximale ! Par contre, cette tension ne sera appliquée que par très courtes périodes de temps. En ajustant la longueur de ces périodes de temps, on arrive à faire avancer plus ou moins vite les moteurs. Mieux, on remarque que la vitesse des moteurs devient proportionnelle à la longueur des périodes de temps. Contrôler la longueur des périodes passées à la tension maximale par rapport au temps passé sans application de tension (tension nulle) est donc le cœur de la solution. En réalité, cette solution est fort connue en contrôle des systèmes et en électronique et elle porte le nom de PWM (Pulse Width Modulation) ou MLI pour Modulation en Largeur d'Impulsion.

Un signal PWM est un signal dont la période est fixe, mais le rapport cyclique varie. En d'autres termes, t_1 et t_2 varient tout en conservant $t_1 + t_2 = T = \text{constante}$.

III.3.4. Intérêt du signal PWM :

Le principal intérêt de la technique PWM est de limiter la chauffe des composants électroniques.

Par exemple : une lampe de 20 Watts allumée au maximum consomme 20 Watt.

Si par une commande de gradation elle est allumée au quart de sa puissance,

la puissance fournie est soit maximale, soit nulle.

Lorsqu'elle est maximale, pendant un quart du temps par exemple, il n'y a pas besoin de dissiper de puissance résiduelle. Lorsqu'elle est nulle, il n'y a pas besoin de dissiper non plus de puissance, car elle n'est pas fournie du tout...

Un autre intérêt de la PWM est que la tension appliquée au moteur pendant t_1 est V_{cc} .

Celle-ci est suffisante pour vaincre les frottements et faire tourner le moteur. Mais la tension moyenne appliquée au moteur est proportionnelle au rapport cyclique, ce qui permet d'avoir des consignes de vitesse faibles : $V_{MOY} = t_1 \cdot V_{cc} / T$.

La commande d'actionneurs de puissance par PWM est très liée à la notion de fréquence. Pour que l'impression d'une valeur moyenne constante d'allumage apparaisse, il faut que l'alternance d'allumage/extinction soit suffisamment rapide pour qu'elle ne se remarque pas.

Selon les utilisations la fréquence de PWM va de 100 Hz à 200 kHz.

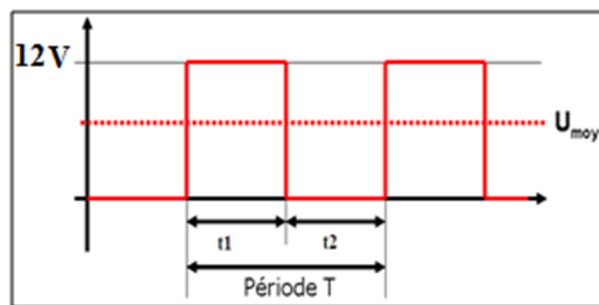


FIGURE III.8. Exemple de signaux PWM.

III.4. Perception-Capteurs :

III.4.1.Mesure de proximité et de distance :

On appelle télémétrie toute technique de mesure de distance par des procédés acoustiques, optiques ou radioélectriques.

L'appareil permettant de mesurer les distances est appelé télémètre.

De même qu'il existe différentes techniques de mesure de distance (mesure du temps de vol d'une onde, triangulation), il existe différentes technologies pour réaliser des télémètres.

Nous présentons ici les plus répandues en robotique mobile, en donnant une idée de leur gamme de mesure et d'application. Tous les capteurs télémétriques, basés sur des mesures de l'environnement, sont bien évidemment actifs.

III.4.1.1.Capteurs infrarouges :

Les détecteurs photoélectriques autrement appelés capteurs opto-électriques, se composent essentiellement d'un émetteur de lumière associé à un récepteur photosensible.

Ils sont schématisés de la manière suivante :

Emetteur et récepteur sont regroupés dans un même boîtier. La portée dépend du pouvoir réfléchissant et de la dimension de l'objet. En l'absence de l'objet, le faisceau n'est pas renvoyé.

En présence de l'objet, le faisceau est renvoyé : il y a alors détection.

Les capteurs de la famille GP2DXX, que nous avons utilisés pour le projet envoient une impulsion lumineuse dans l'air. Si elle est renvoyée par un obstacle dans un certain laps de temps donné, le faisceau parvient au récepteur formant un triangle. Le récepteur, donne l'information que l'objet est proche et renvoie un niveau logique correspondant. Dans la famille, d'autres capteurs mesurent l'angle formé par le capteur avec le faisceau, déterminant précisément la distance à l'objet.

En sortie, le signal est donc analogique et doit être mis en forme, pour être traité.

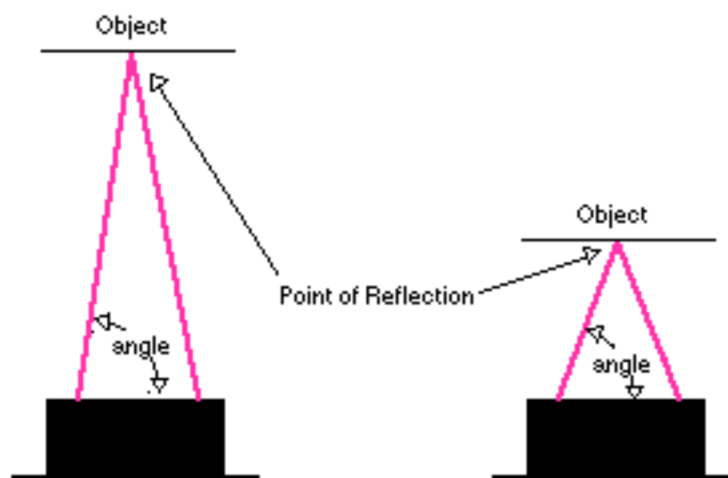


FIGURE III.9.Principe de mesure de distance.

III.4.1.1.1.Les capteurs utilisés « GP2D15 et GP2D150» :

Notre robot est équipé de capteurs photoélectriques « numériques » GP2D15, qui ont la particularité de posséder un seuil de détection situé aux alentours de 24 cm. En effet un état haut apparaît lorsque la distance est inférieure ou égale à 24 cm. Le GP2D15 a un seuil à hystérésis.

Ce capteur -contrairement aux capteurs analogiques- a la particularité d'être insensible à la lumière, c'est-à-dire, fonctionnant dans la lumière ou dans l'obscurité, il renvoie le même signal de sortie V_{out} .

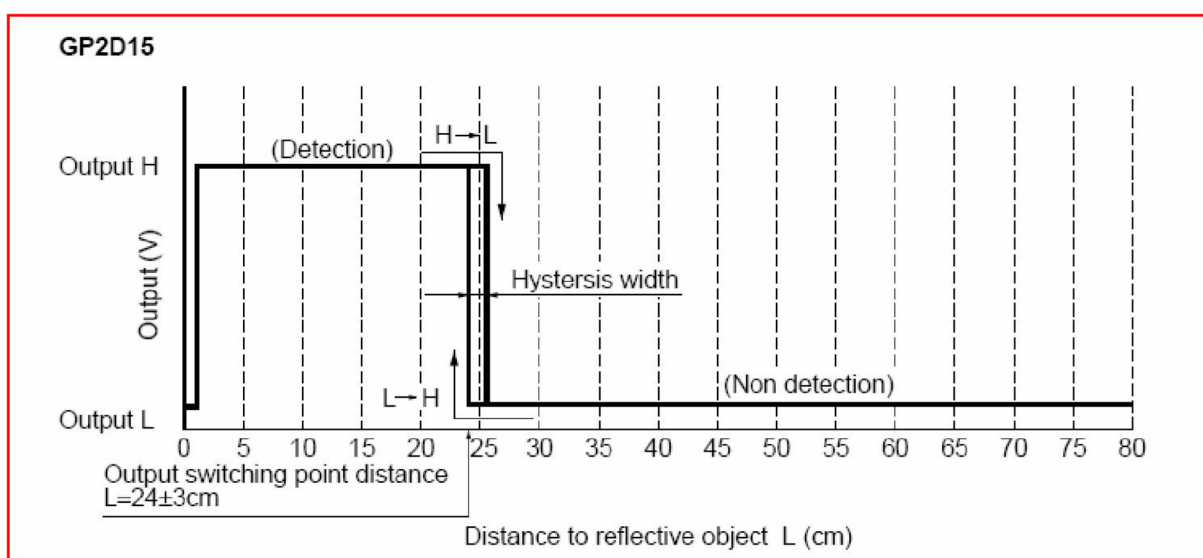


FIGURE III.10 . Courbe d'hystérésis du GP2D15.

III.4.1.1.2. Le « GP2D15 » en détecteur d'obstacle:

Le robot doit être équipé d'un système d'évitement d'obstacle pour éviter le robot adverse et assurer sa trajectoire vers la zone de construction.

Ceci peut se faire en fixant un capteur GP2D15 à droite et un autre à gauche, une routine d'évitement d'obstacle s'active dès que le robot franchit le seuil de déclenchement du capteur (24 cm).

Ce capteur a l'avantage d'avoir une distance minimale de 0 cm, c'est-à-dire qu'il se déclenchera à n'importe quelle distance inférieure à 24 cm s'il rencontre un obstacle, contrairement à d'autres capteurs qui ont une distance minimale de 4 jusqu'à 20 cm.

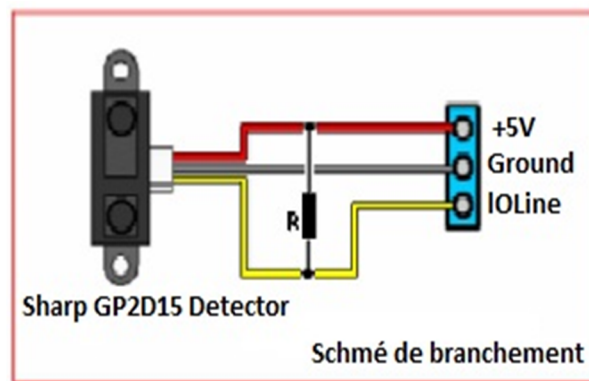


FIGURE III.11. Télémètre infrarouge GP2D15.

III.5. Conclusion :

Ce chapitre nous a permis de voir le cœur de notre robot, les actionneurs chargés des mouvements de construction ainsi que les différents composants qui communiquent avec l'extérieur (les capteurs).

Notre but dans ce travail étant de bien contrôler le déplacement du robot, nous aurons

donc besoin d'exploiter et gérer les signaux reçus des encodeurs, c'est ce que nous verrons dans le chapitre suivant concernant l'odométrie.

III.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 d'accomplir ses tâches il est nécessaire de réaliser des cartes telles que la carte de commande, l'isolation galvanique et la carte de puissance, notre robot se compose de 2 cartes principales :

- Une carte mère qui est la carte de commande.

- Une carte de puissance qui comporte aussi un étage d'isolation galvanique permettant de commander 2 moteurs à courant continu.

Nous allons voir dans ce chapitre le fonctionnement de l'électronique ainsi que des capteurs et actionneurs utilisés.

III.2.Partie commande :

III.2.1.Eléments à contrôler :

- Moteurs de traction : servant au déplacement du robot.

- Microswitchs :

Pour se positionner droit en face des bordures .

- Télémètres infra rouges :

Pour la détection d'obstacle .

III.2.2.Description fonctionnelle des cartes électroniques :

La carte de commande contrôle directement les moteurs de traction et reçoit les signaux en provenance :

- * Des encodeurs des moteurs de locomotion.

- * Des capteurs infra rouges pour la détection d'obstacle.

- * Des microswitchs.

La carte de puissance (conçue autour du circuit L298) reçoit les signaux en provenance de la carte de commande pour commander les moteurs à courant continu.

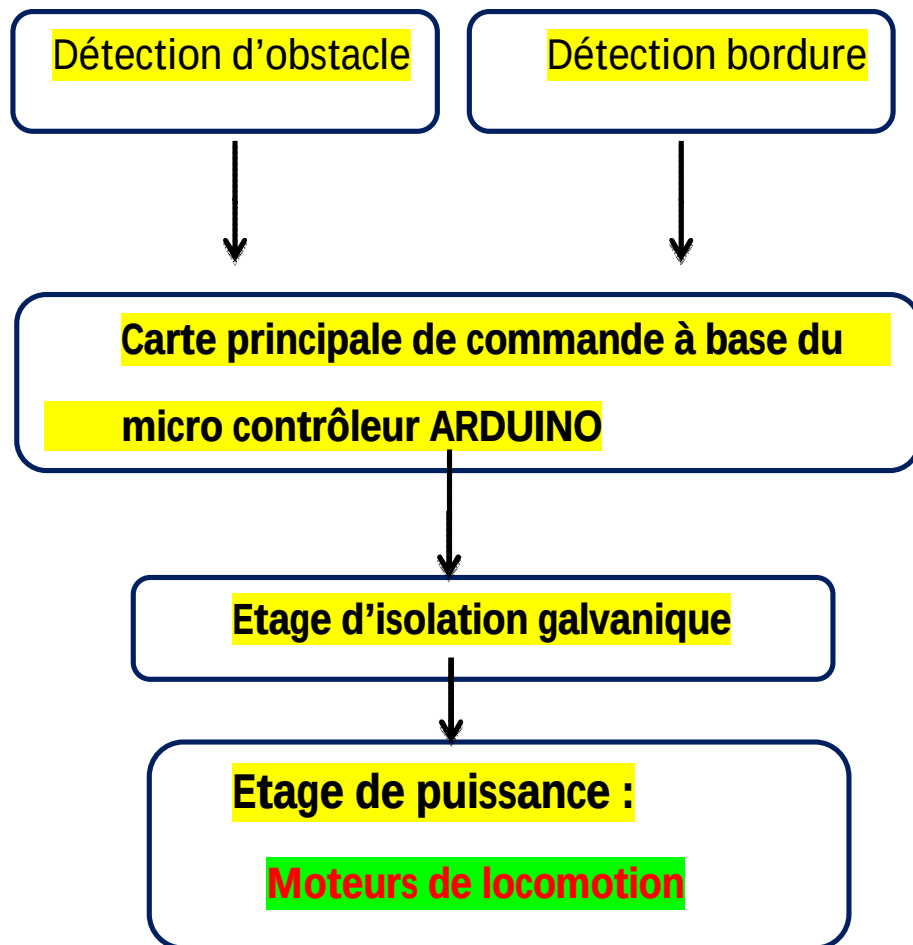


FIGURE III.1. Schéma synoptique des étages électronique

III.2.2.1.ARDUINO :



FIGURE III.2. La carte Arduino Duemilanove

Vue d'ensemble :

La carte Arduino Duemilanove("2009") est une carte à microcontrôleur basée sur l'ATmega168 pour les premières version ou sur l'ATmega328 pour les versions actuelles. Elle dispose :

de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),

de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),

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, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

"Duemilanove" signifie 2009 en Italien et ce nom est donné d'après l'année de sa sortie. La Duemilanove est la dernière née dans la série des cartes Arduino USB; pour une comparaison avec les cartes précédentes,

Alimentation :

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

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 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. [Aussi, la plage idéale recommandée pour alimenter la carte Duemilanove est entre 7V et 12V.](#)

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). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques

numériques nécessitent une tension d'alimentation parfaitement stable

dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

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) de la carte est disponible ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V).

L'intensité maximale disponible sur cette broche est de 50mA

GND. Broche de masse (ou 0V).

Mémoire :

L'ATmega 168 a 16 Ko de mémoire FLASH pour stocker le programme (dont 2ko sont utilisés par le bootloader); L'ATmega 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 2Ko

également utilisés par le bootloader). L'ATmega 168 a 1Ko de mémoire SRAM (volatile) et 512Ko d'EEPROM (non volatile) (mémoire qui peut être écrite et lue à l'aide de la librairie EEPROM?); L'ATmega 328 a 2ko de mémoire SRAM (volatile) et 1Ko d'EEPROM (non volatile).

Pour info : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port série, notamment lors de chaque programmation de la carte.

Entrées et sorties numériques :

Chacune des 14 broches numériques de la carte Duemilanove (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(), digitalWrite() et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite(broche, HIGH).

De plus, certaines broches ont des fonctions spécialisées :

Communication Serie: Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré FTDI de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Interruptions Externes: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction attachInterrupt() pour plus de détails.

Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite().

SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) qui, bien que disponible d'un point de vue matériel, n'est pas actuellement inclut dans le langage Arduino. Une librairie pour communication SPI est cependant disponible.

I2C: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils") .

LED: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte..

Broches analogiques :

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

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

Autres broches :

Il y a deux autres broches disponibles sur la carte

AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference().

Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

Communication :

La carte Arduino dispose de toute une série de facilité pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega 168 et l'ATmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour

communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré FT232RL (circuit FTDI) sur la carte assure la connexion

entre cette communication série vers le port USB de l'ordinateur et les drivers FTDI (qui sont inclus avec le logiciel Arduino) qui créent un port COM virtuel sur l'ordinateur utilisable par les logiciels.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré FTDI et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet la communication série sur n'importe quelle broche numérique de la carte Duemilanove.

L'ATmega 168 et l'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. Voir la documentation pour les détails Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

Programmation :

La carte Arduino Duemilanove peut être programmée avec le logiciel Arduino (à télécharger ici). Il suffit de sélectionner "Arduino Diecimila ou Duemilanove w/ ATmega168" ou "Arduino Duemilanove w/ ATmega328" dans le menu **Tools > Board** (en fonction du microcontrôleur présent sur votre carte). Pour plus de détails sur le langage Arduino, voir la référence du langage Arduino et pour apprendre à programmer en langage Arduino voir la page Apprendre.

Les microcontrôleurs ATmega168 ou ATmega328 présents sur la carte Arduino Duemilanove sont livrés avec un bootloader (petit programme de démarrage) préprogrammé qui vous permet de transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe. Ce bootloader communique avec le microcontrôleur en utilisant le protocole original STK500 (reference, fichiers C).

Vous pouvez bien sûr passer outre le bootloader et programmer le microcontrôleur via le connecteur ICSP (In-Circuit Serial Programming - "Programmation Série Dans le circuit" en français); voir ces instructions (en anglais) pour plus de détails.

Réinitialisation (logicielle) automatique :

Plutôt que de nécessiter un appui sur le bouton poussoir de réinitialisation avant un transfert de programme, la carte Arduino Duemilanove a été conçue de telle façon qu'elle puisse être réinitialisée par un logiciel tournant sur l'ordinateur. Une des broches matérielles de contrôle du flux (DTR) du

circuit intégré FT232RL est connecté à la ligne de réinitialisation de l'ATmega (168 ou 328) via un condensateur de 100 nanofarads. Lorsque cette broche est mise au niveau BAS, la broche de réinitialisation s'abaisse suffisamment longtemps pour réinitialiser le microcontrôleur. Le logiciel

Arduino utilise cette possibilité pour vous permettre de transférer votre programme dans la carte par un simple clic sur le bouton de transfert de la barre de boutons de l'environnement Arduino. Cela signifie que le bootloader peut avoir un temps mort plus court, la mise au niveau bas de la broche DTR étant bien coordonnée avec le début du transfert du programme.

Ce fonctionnement a d'autres implications. Quand la carte Duemilanove est connectée à un ordinateur fonctionnant soit sous Mac Os X ou Linux, la carte se réinitialise à chaque fois qu'une connexion se fait entre elle et le logiciel (via le port USB). Pendant la demi-seconde suivante, le bootloader s'exécute sur la carte Duemilanove. Puisqu'il est programmé pour ignorer les données mal formatées (càd toute donnée en plus du transfert d'un nouveau programme), il interceptera les quelques premiers octets de donnée envoyé à la carte après qu'une connexion soit établie. Si un programme s'exécutant sur la carte reçoit une configuration initiale

ou d'autres données quand il démarre, assurez-vous que le logiciel avec lequel il communique attend une seconde après l'ouverture de la connexion avant d'envoyer ces données.

La carte Duemilanove comporte une piste (du circuit imprimé) que vous pouvez couper pour désactiver la réinitialisation automatique. Les bords de chaque côté de la piste peuvent être solidarisés ensemble pour la réactiver. Vous pouvez également désactiver la réinitialisation automatique en connectant une résistance de 110 Ohms entre le +5V et la ligne de réinitialisation (reset); voir [ce fil du forum](#) (en anglais) pour plus de détails. **Protection du port USB contre la surcharge en intensité**

La carte Arduino Duemilanove a un polyfusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

Caractéristiques Mécaniques :

Les longueurs et largeurs maximales de la Duemilanove sont respectivement 6.86 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation Jack s'étendant au-delà des dimensions de la carte. Trois trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier. Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.

Ecouter le nom :

Voici comment vous devez prononcer le nom de la carte en italien correct : téléchargez le fichier sonore dans le format que vous préférez : [WAV](#), [OGG](#), [MP3](#), [FLAC](#), [WMA](#).

Commentaires utilisateurs :

Le bootloader est un programme préprogrammé une fois pour toute dans l'ATmega et qui permet la communication entre l'ATmega et le logiciel Arduino via le port série, notamment lors de chaque programmation de la carte.

La mémoire disponible avec l'ATMega328 est largement suffisante pour les programmes. A titre d'exemple, un programme simple utilisant clavier et afficheur LCD avec calculs mathématiques n'occupera que 5 ou 6K de mémoire, ce qui laisse de la marge...

Pour ceux qui auraient une carte Arduino avec un ATmega 168 et qui voudraient passer à l'ATmega 328, ce dernier est vendu pré-programmé avec le bootloader pour quelques euros.

En pratique, la programmation par un programmeur externe, bien que possible comme évoqué ci-dessus n'a pas grand intérêt puisque tout l'avantage du système Arduino est justement de pouvoir programmer la carte Arduino directement par le port USB, sans programmeur externe. Le bootloader ne faisant que 2Ko, c'est 30 Ko de programme qui sont disponibles avec l'ATmega 328, ce qui est très largement suffisant dans la plupart des cas.

III.2.2.2. Le câble USB du ARDUINO :

Ce câble nous permettra de télécharger nos programmes (tâches) au sein du module ARDUINO et de déboguer ces derniers.

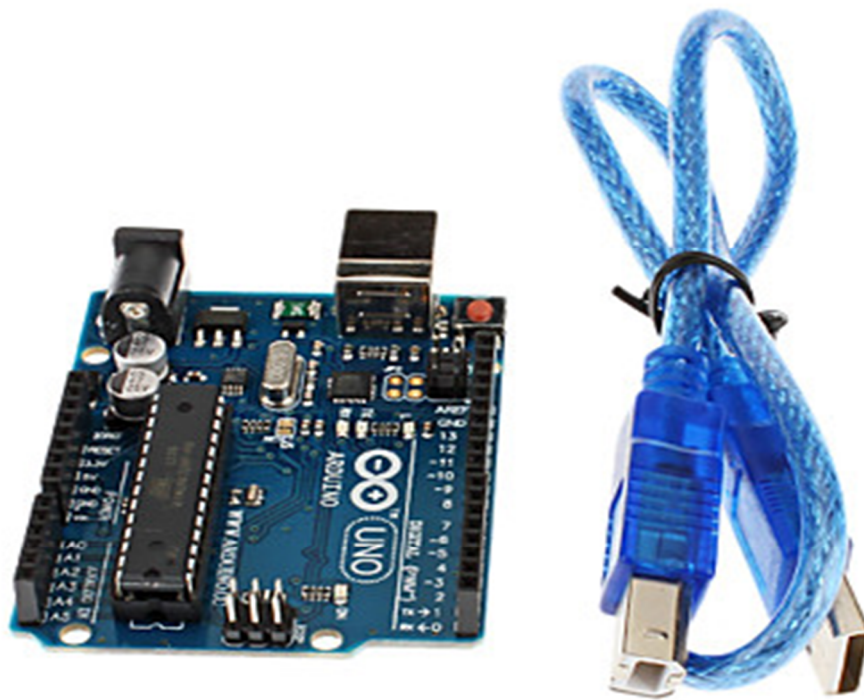


FIGURE III.3. Le câble USB du ARDUINO avec le module ARDUINO



FIGURE III.4. Le câble USB du ARDUINO.

III.3.Partie puissance :

III.3.1.La carte de puissance :

La carte de puissance joue le rôle d'une carte d'interface entre la carte de commande et la partie motrice comportant les composants mécaniques, elle nous permet de commander les moteurs à courant continu suivant les ordres fournis par la carte de commande.

Cette carte comporte également un étage d'isolation galvanique à base de circuit intégré TLP 521 afin de rendre les 2 parties totalement indépendantes et éviter tout phénomène de destruction par électricité statique.

Etant donné que notre robot contient 2 moteurs à courant continu, on a réalisé une carte qui comporte 2 sorties moteurs.

III.3.2. Le principe du pont en H :

Dans notre montage, nous avons besoin de quatre ponts en H, c'est pour cette raison que nous avons utilisées deux circuits intégrés L298, car chaque circuit offre un double pont en H.

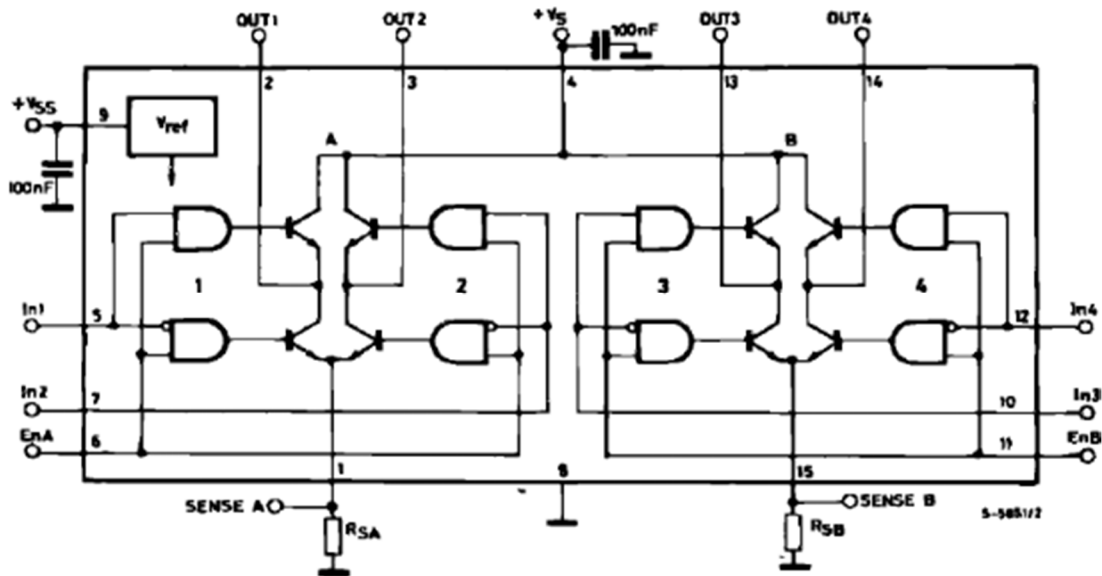


FIGURE III.5. Diagramme bloqué du L298.

Les caractéristiques du L298 sont les suivantes :

- Intensité maximale : 2A par pont ;
- Alimentation de puissance de 5.5V à 50V
- Type de boîtier : Multiwatt15 ;
- Dissipation puissance total : 25w
- Trois entrées par pont : In1, In2 et ENABLE.

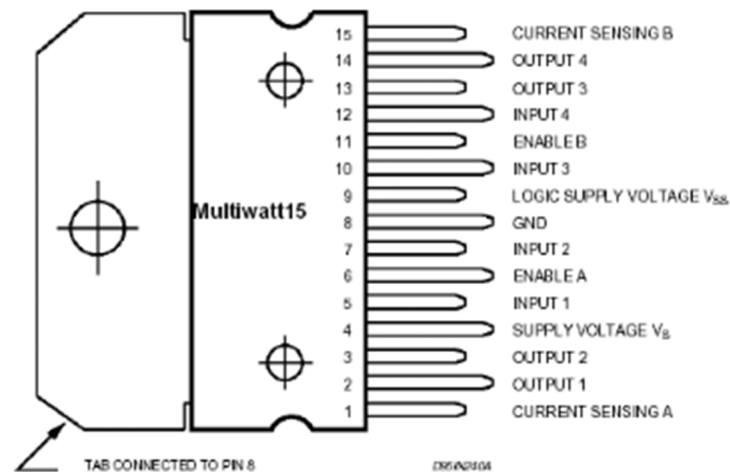


FIGURE III.7. Brochage du circuit intégré L298.

III.3.3. La variation de vitesse (commande par PWM) :

Quand on veut faire varier la vitesse d'un moteur, la première idée qui vient à l'esprit est de faire varier la tension aux bornes du moteur, mais on constate que pour des valeurs faibles de la tension,

le moteur ne tourne pas. Le moteur demande une tension minimale assez importante pour démarrer. Si cette dernière est trop basse, les forces électromagnétiques ne sont pas suffisantes pour combattre le frottement. Il devient donc difficile d'ajuster la vitesse de façon précise.

La solution à ce problème est astucieuse. Il suffit de fournir au moteur une tension qui est toujours la même soit la tension maximale ! Par contre, cette tension ne sera appliquée que par très courtes périodes de temps. En ajustant la longueur de ces périodes de temps, on arrive à faire avancer plus ou moins vite les moteurs. Mieux, on remarque que la vitesse des moteurs devient proportionnelle à la longueur des périodes de temps. Contrôler la longueur des périodes passées à la tension maximale par rapport au temps passé sans application de tension (tension nulle) est donc le cœur de la solution. En réalité, cette solution est fort connue en contrôle des systèmes et en électronique et elle porte le nom de PWM (Pulse Width Modulation) ou MLI pour Modulation en Largeur d'Impulsion.

Un signal PWM est un signal dont la période est fixe, mais le rapport cyclique varie. En d'autres termes, t_1 et t_2 varient tout en conservant $t_1 + t_2 = T = \text{constante}$.

III.3.4. Intérêt du signal PWM :

Le principal intérêt de la technique PWM est de limiter la chauffe des composants électroniques.

Par exemple : une lampe de 20 Watts allumée au maximum consomme 20 Watt.

Si par une commande de gradation elle est allumée au quart de sa puissance,

la puissance fournie est soit maximale, soit nulle.

Lorsqu'elle est maximale, pendant un quart du temps par exemple, il n'y a pas besoin de dissiper de puissance résiduelle. Lorsqu'elle est nulle, il n'y a pas besoin de dissiper non plus de puissance, car elle n'est pas fournie du tout...

Un autre intérêt de la PWM est que la tension appliquée au moteur pendant t_1 est V_{cc} .

Celle-ci est suffisante pour vaincre les frottements et faire tourner le moteur. Mais la tension moyenne appliquée au moteur est proportionnelle au rapport cyclique, ce qui permet d'avoir des consignes de vitesse faibles : $V_{MOY} = t_1 \cdot V_{cc} / T$.

La commande d'actionneurs de puissance par PWM est très liée à la notion de fréquence. Pour que l'impression d'une valeur moyenne constante d'allumage apparaisse, il faut que l'alternance d'allumage/extinction soit suffisamment rapide pour qu'elle ne se remarque pas.

Selon les utilisations la fréquence de PWM va de 100 Hz à 200 kHz.

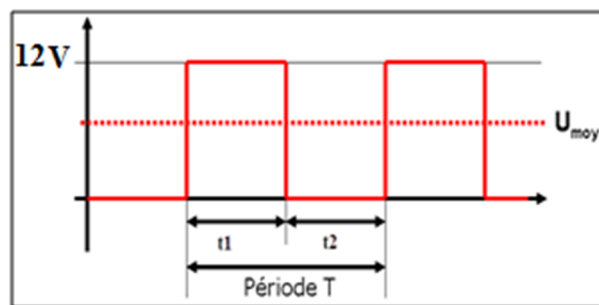


FIGURE III.8. Exemple de signaux PWM.

III.4. Perception-Capteurs :

III.4.1.Mesure de proximité et de distance :

On appelle télémétrie toute technique de mesure de distance par des procédés acoustiques, optiques ou radioélectriques.

L'appareil permettant de mesurer les distances est appelé télémètre.

De même qu'il existe différentes techniques de mesure de distance (mesure du temps de vol d'une onde, triangulation), il existe différentes technologies pour réaliser des télémètres.

Nous présentons ici les plus répandues en robotique mobile, en donnant une idée de leur gamme de mesure et d'application. Tous les capteurs télémétriques, basés sur des mesures de l'environnement, sont bien évidemment actifs.

III.4.1.1.Capteurs infrarouges :

Les détecteurs photoélectriques autrement appelés capteurs opto-électriques, se composent essentiellement d'un émetteur de lumière associé à un récepteur photosensible.

Ils sont schématisés de la manière suivante :

Emetteur et récepteur sont regroupés dans un même boîtier. La portée dépend du pouvoir réfléchissant et de la dimension de l'objet. En l'absence de l'objet, le faisceau n'est pas renvoyé.

En présence de l'objet, le faisceau est renvoyé : il y a alors détection.

Les capteurs de la famille GP2DXX, que nous avons utilisés pour le projet envoient une impulsion lumineuse dans l'air. Si elle est renvoyée par un obstacle dans un certain laps de temps donné, le faisceau parvient au récepteur formant un triangle. Le récepteur, donne l'information que l'objet est proche et renvoie un niveau logique correspondant. Dans la famille, d'autres capteurs mesurent l'angle formé par le capteur avec le faisceau, déterminant précisément la distance à l'objet.

En sortie, le signal est donc analogique et doit être mis en forme, pour être traité.

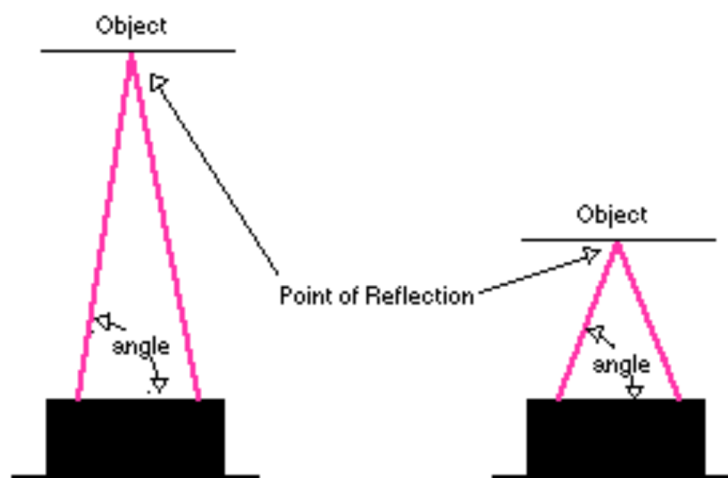


FIGURE III.9.Principe de mesure de distance.

III.4.1.1.1.Les capteurs utilisés « GP2D15 et GP2D150» :

Notre robot est équipé de capteurs photoélectriques « numériques » GP2D15, qui ont la particularité de posséder un seuil de détection situé aux alentours de 24 cm. En effet un état haut apparaît lorsque la distance est inférieure ou égale à 24 cm. Le GP2D15 a un seuil à hystérésis.

Ce capteur -contrairement aux capteurs analogiques- a la particularité d'être insensible à la lumière, c'est-à-dire, fonctionnant dans la lumière ou dans l'obscurité, il renvoie le même signal de sortie V_{out} .

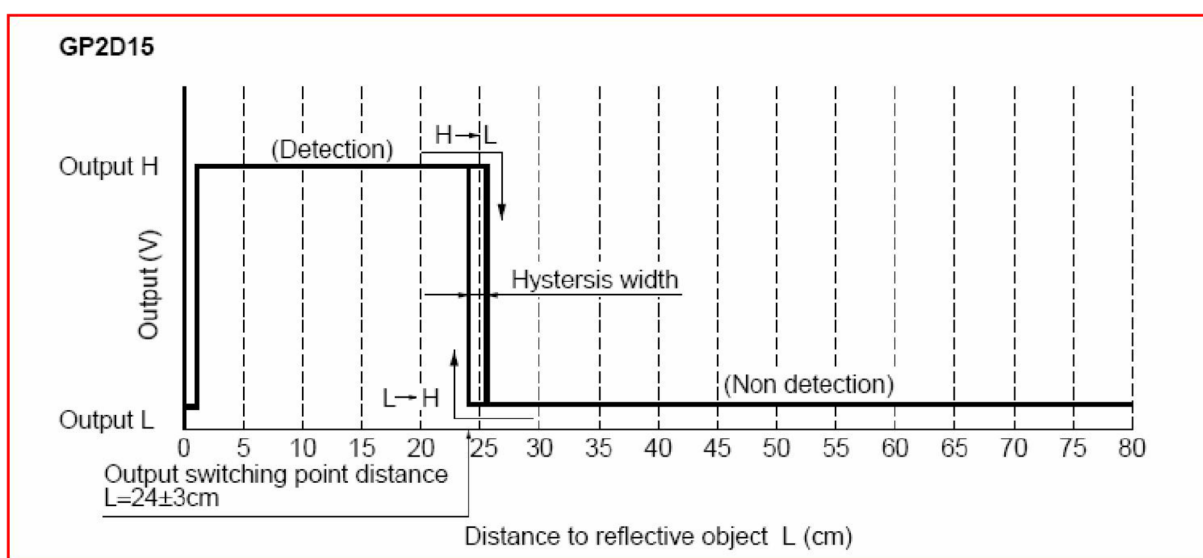


FIGURE III.10 . Courbe d'hystérésis du GP2D15.

III.4.1.1.2. Le « GP2D15 » en détecteur d'obstacle:

Le robot doit être équipé d'un système d'évitement d'obstacle pour éviter le robot adverse et assurer sa trajectoire vers la zone de construction.

Ceci peut se faire en fixant un capteur GP2D15 à droite et un autre à gauche, une routine d'évitement d'obstacle s'active dès que le robot franchit le seuil de déclenchement du capteur (24 cm).

Ce capteur a l'avantage d'avoir une distance minimale de 0 cm, c'est-à-dire qu'il se déclenchera à n'importe quelle distance inférieure à 24 cm s'il rencontre un obstacle, contrairement à d'autres capteurs qui ont une distance minimale de 4 jusqu'à 20 cm.

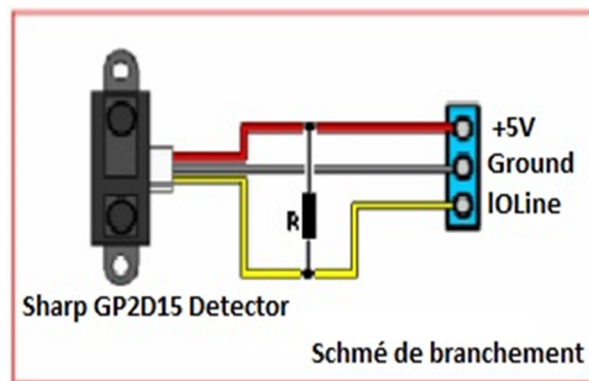


FIGURE III.11. Télémètre infrarouge GP2D15.

III.5. Conclusion :

Ce chapitre nous a permis de voir le cœur de notre robot, les actionneurs chargés des mouvements de construction ainsi que les différents composants qui communiquent avec l'extérieur (les capteurs).

Notre but dans ce travail étant de bien contrôler le déplacement du robot, nous aurons

donc besoin d'exploiter et gérer les signaux reçus des encodeurs, c'est ce que nous verrons dans le chapitre suivant concernant l'odométrie.

IV.1.Introduction :

L'odométrie est une technique permettant d'estimer la position d'un robot mobile en mouvement. Littéralement, il s'agit de la science de la mesure et du contrôle du mouvement d'un robot. La mesure se base pour cela sur la mesure individuelle des déplacements des roues du robot.

En effet, l'idée est assez simple : si l'on connaît le sens de rotation d'une roue, sa vitesse ainsi que sa durée de rotation et que cette mesure intervient sur toutes les roues du robot, alors on peut estimer l'emplacement du robot par rapport à son point de départ.

L'odométrie fonctionne relativement bien avec des robots munis de roues, elle est bruitée avec les robots utilisant des chenilles à cause du contact de ces dernières avec le sol qui est mal défini, et enfin elle est très mauvaise avec les robots utilisant des pattes pour le déplacement.

Le problème avec l'odométrie, c'est que les erreurs s'accumulent au fil du temps. Il est souvent intéressant de coupler ces mesures avec des mesures ponctuelles basées sur d'autres capteurs (ultrasons, infrarouge, lasers...).

IV.2 Principe de fonctionnement des encodeurs:

Un encodeur est un dispositif électromécanique capable de mesurer le mouvement ou la position. La plupart des encodeurs utilisent des capteurs optiques pour fournir des signaux électriques sous forme de trains d'impulsions, lesquels peuvent, à leur tour, être interprétés comme mouvement, direction ou position.

Les encodeurs rotatifs servent à mesurer le déplacement rotatif d'un axe. La figure n°1 montre les composants de base d'un encodeur rotatif :

une diode électroluminescente (LED), un disque et un capteur de lumière de l'autre côté du disque. Le disque, monté sur l'axe rotatif, comporte des motifs encodés de secteurs opaques et transparents. Lorsque le disque tourne, les segments opaques bloquent la lumière alors que ceux où le verre est clair la laissent passer.

Ceci génère des impulsions d'onde carrée qui peuvent ensuite être interprétées comme position ou mouvement. En général, les encodeurs ont entre 100 et 6 000 segments par révolution.

Ceci correspond à une résolution de 3,6 degrés pour un encodeur à 100 segments et à une résolution de 0,06 degrés pour un encodeur à 6 000 segments.

Le même principe de fonctionnement est valable pour les encodeurs linéaires, si ce n'est que le disque rotatif est remplacé par une bande opaque stationnaire dont la surface comporte des fentes et que l'assemblage capteur/LED est fixé sur la partie mobile.

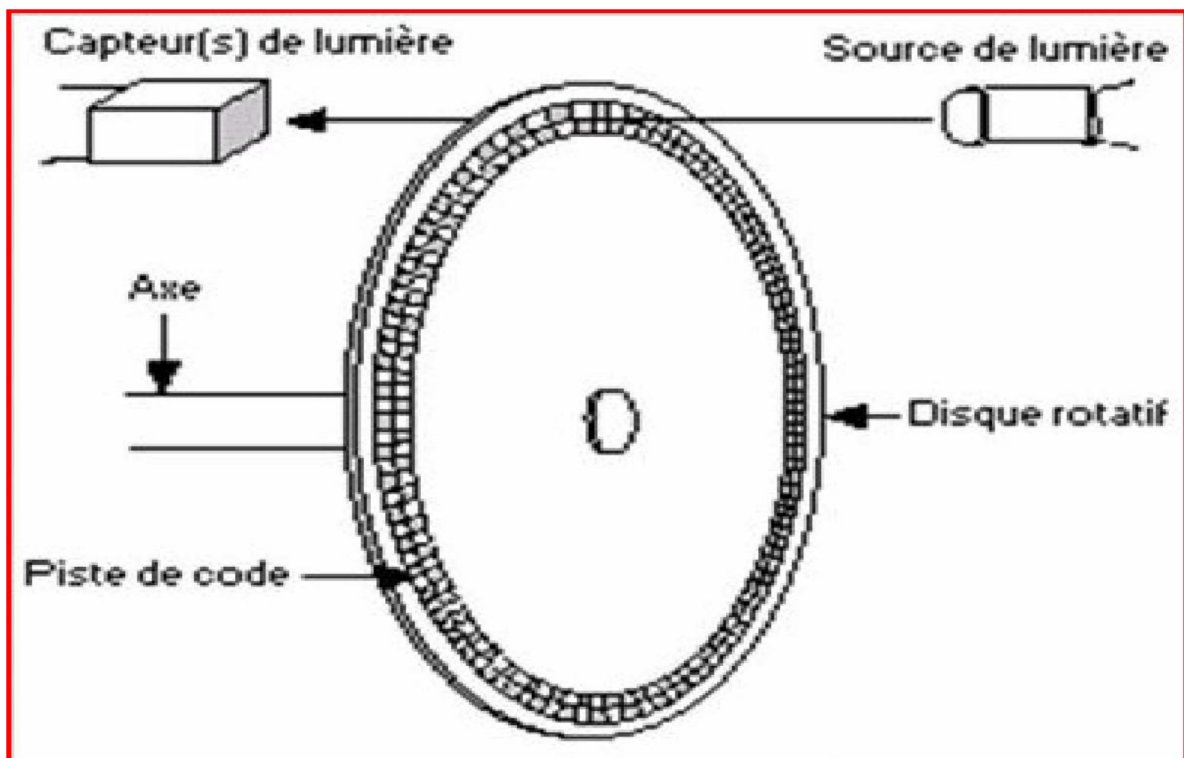


FIGURE IV.1. Les différents composants d'un encodeur rotatif.

Un encodeur ne générant qu'une série d'impulsions ne serait pas très utile puisqu'il ne pourrait pas indiquer la direction de la rotation. L'encodeur en quadrature comporte deux pistes de code dont les secteurs sont décalés de 90 degrés d'une piste à l'autre ; ceci permet d'avoir deux sorties de voie qui indiquent la position et la direction de la rotation. Si A devance B, par exemple, le disque tourne dans le sens des aiguilles d'une montre. Si B devance A, le disque

tourne dans le sens contraire. Par conséquent, en surveillant à la fois le nombre d'impulsions et les phases relatives des signaux A et B, vous pouvez suivre la position et la direction de la rotation.

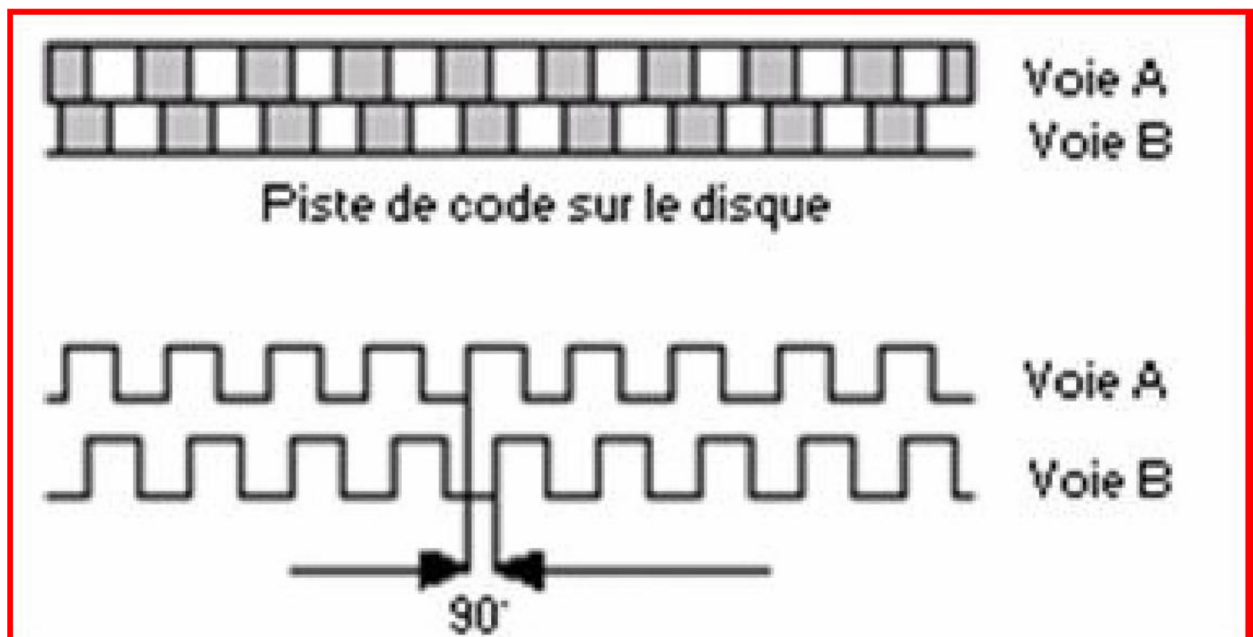


FIGURE IV.2. L'encodeur en quadrature.

Certains encodeurs en quadrature comportent aussi une troisième sortie de voie, appelée signal de référence ou zéro, qui renvoie une impulsion par révolution. On peut utiliser cette impulsion pour déterminer précisément une position de référence. Dans la plupart des encodeurs, ce signal correspond au terminal ou index.

IV.3.Régulation en vitesse :

Le choix du profil de vitesse est un problème similaire dans le cas du déplacement rectiligne et de la rotation pure. Il s'agit, à partir d'une vitesse et d'une accélération nulles, d'atteindre une nouvelle position avec également une vitesse et accélération nulles à l'arrivée.

Pour modifier cette vitesse il suffit juste de modifier le signal PWM.

Sachant que la roue tourne à 121 tours/min à 12V ça sera donc sa vitesse maximal

On aura alors :

$$V_{\max} = (121 \times 2\pi \times r) / 60.$$

Avec r : rayon de la roue qui est 100 mm.

On aura donc

$$V_{\max} = 1.12 \text{ m/s.}$$

Afin d'éviter le glissement des roues il est conseillé de diminuer cette vitesse pendant le déplacement.

IV.4.Régulation en position :

IV.4.1.Calcul du déplacement en ligne droite:

Etant donné que notre robot effectuera des déplacements en marche avant et en rotation, nous allons donc utiliser un seul canal de l'encodeur qui est le canal A, pour le comptage des impulsions les 2 pins counter_0 et counter_1 du module ARDUINO seront exploitées En

connaissant aussi le diamètre des roues, nous pourrons définir exactement la distance parcourue par notre robot ainsi que les angles de rotations.

En partant de la formule suivante, on peut calculer la distance parcourue, il suffit de multiplier l'angle obtenu par le diamètre de la roue $\times \pi$.

$$(\text{Angle de rotation}) = (\text{Nbre d'impulsions} / N) \times 360.$$

Où N est: le

nombre d'impulsions générées par l'encodeur à chaque révolution de l'axe.

Donc la distance D parcourue par la roue sera donnée par :

$$D = \text{angle de rotation} \times d \times \pi.$$

Où d est le diamètre de la roue.

IV.4.2.Calcul d'odométrie :

La position du robot est calculée au moyen d'une formule basée sur la vitesse angulaire et la vitesse d'avance du robot. Cette méthode a été choisie pour les raisons suivantes :

- vitesses de rotation des roues mesurables, et par conséquent vitesses angulaire et d'avance faciles à calculer .
- calculs réalisables dans le temps à disposition .
- facile à implémenter.

Les vitesses angulaires et d'avance ne sont pas directement mesurables. C'est la vitesse de chaque roue qui est mesurée à chaque instant.

On en déduit facilement les vitesses angulaires et d'avance.

La formule suivante permet d'obtenir la vitesse de rotation du robot :

$$\omega = (V_{\text{droite}} - V_{\text{gauche}}) / E$$

Où E est la distance entre les deux roues du robot. Une vitesse positive correspond à une rotation dans le sens trigonométrique positif. Nous estimons ensuite la vitesse d'avance du robot en faisant la moyenne entre les deux vitesses des roues :

$$V = (V_{\text{droite}} + V_{\text{gauche}}) / 2$$

Connaissant le temps de cycle et une estimation de la vitesse angulaire, on trouve

facilement l'angle effectué, qu'il suffit d'ajouter à l'ancien angle α_{old} pour connaître la nouvelle orientation α_{new} du robot. Concernant le changement de position, le principe est similaire. On détermine une vitesse d'avance du robot, qu'on considère comme constante durant le cycle précédent.

Là aussi on prend la moyenne entre la vitesse actuelle et la vitesse précédente. Une fois la distance parcourue calculée, il est nécessaire de connaître la direction du déplacement. On prend pour cela la moyenne entre l'angle actuel et l'angle précédent.

Les calculs permettant de mettre à jour la position sont présentés ci-après.

La figure suivant permet de comprendre intuitivement la méthode utilisée.

Localisation relative

Le principe de la localisation relative consiste à placer le robot dans un environnement et de lui faire effectuer une trajectoire. Connaissant la position initiale et avec les informations données grâce au encodeurs des roues, nous pouvons déterminer la position du robot à un instant quelconque.

Les schémas ci-dessous expliquent ce principe. La position du robot est ensuite déterminée à l'aide des équations mathématiques ci-dessous.

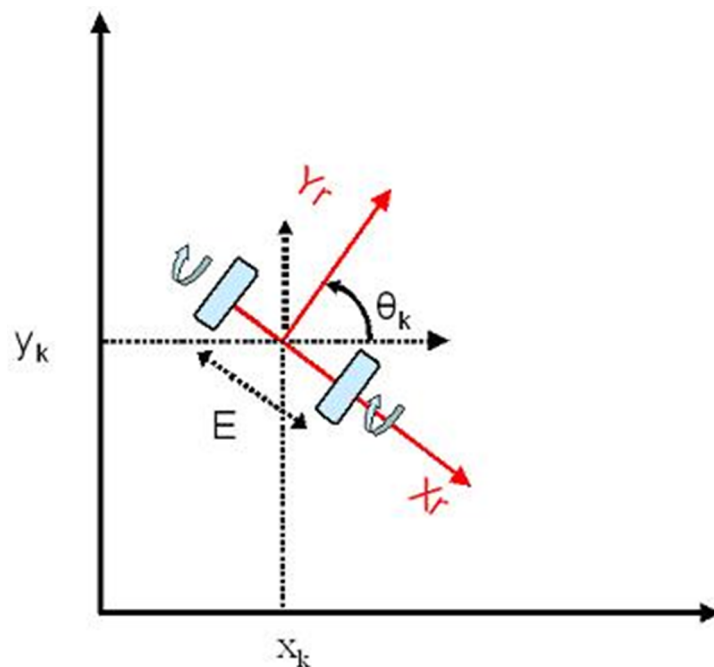


FIGURE IV.3 position du robot .

(X_r, Y_r) : repère attaché au robot

E : largeur du robot

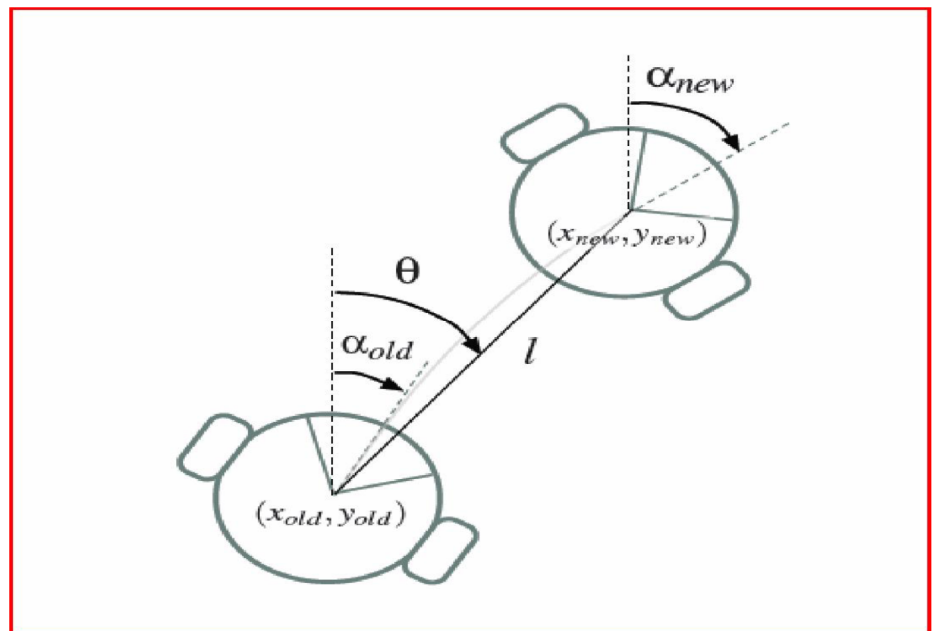
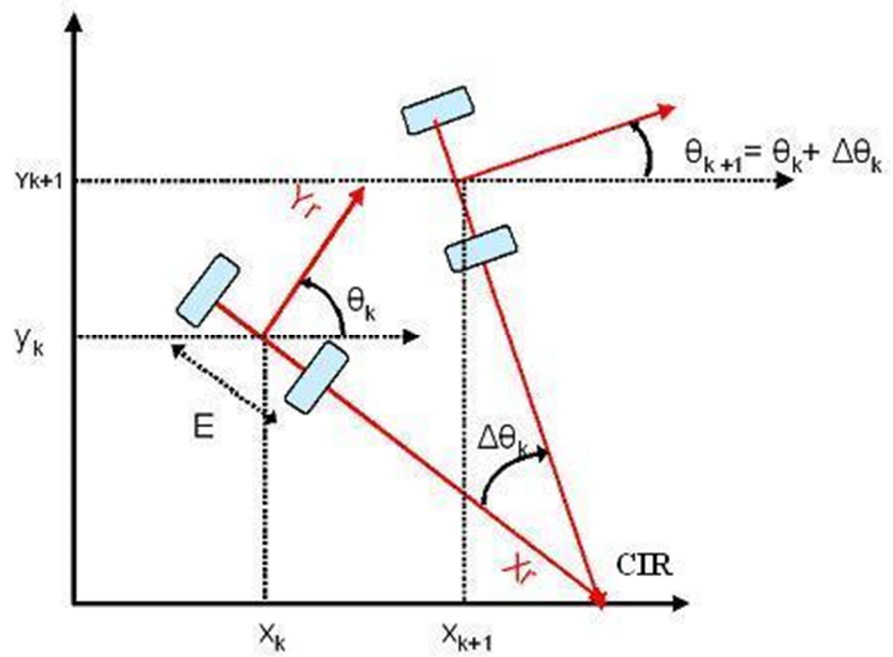


FIGURE IV.4. Estimation de la position absolue du robot.

$$x_{k+1} = x_k + \Delta D_k * \cos(\theta_k + \Delta\theta_k/2)$$

$$y_{k+1} = y_k + \Delta D_k * \sin(\theta_k + \Delta\theta_k/2)$$

Avec

$$- \Delta D_k = (\Delta d_d + \Delta d_g)/2$$

$$- \Delta\theta_k = (\Delta d_d - \Delta d_g)/E$$

Δd_d et Δd_g : déplacements élémentaires des roues droite et gauche

IV.5. Conclusion :

Nous avons vu dans ce chapitre des méthodes de calculs des différentes positions du robot.

En sachant pertinemment que ce qu'on aura en pratique est loin de ressembler à ce que

nous avons obtenu théoriquement, nous avons donc centré la majorité de notre travail sur le plan pratique, et en termine avec une conclusion general.

IV.1.Introduction :

L'odométrie est une technique permettant d'estimer la position d'un robot mobile en mouvement. Littéralement, il s'agit de la science de la mesure et du contrôle du mouvement d'un robot. La mesure se base pour cela sur la mesure individuelle des déplacements des roues du robot.

En effet, l'idée est assez simple : si l'on connaît le sens de rotation d'une roue, sa vitesse ainsi que sa durée de rotation et que cette mesure intervient sur toutes les roues du robot, alors on peut estimer l'emplacement du robot par rapport à son point de départ.

L'odométrie fonctionne relativement bien avec des robots munis de roues, elle est bruitée avec les robots utilisant des chenilles à cause du contact de ces dernières avec le sol qui est mal défini, et enfin elle est très mauvaise avec les robots utilisant des pattes pour le déplacement.

Le problème avec l'odométrie, c'est que les erreurs s'accumulent au fil du temps. Il est souvent intéressant de coupler ces mesures avec des mesures ponctuelles basées sur d'autres capteurs (ultrasons, infrarouge, lasers...).

IV.2 Principe de fonctionnement des encodeurs:

Un encodeur est un dispositif électromécanique capable de mesurer le mouvement ou la position. La plupart des encodeurs utilisent des capteurs optiques pour fournir des signaux électriques sous forme de trains d'impulsions, lesquels peuvent, à leur tour, être interprétés comme mouvement, direction ou position.

Les encodeurs rotatifs servent à mesurer le déplacement rotatif d'un axe. La figure n°1 montre les composants de base d'un encodeur rotatif :

une diode électroluminescente (LED), un disque et un capteur de lumière de l'autre côté du disque. Le disque, monté sur l'axe rotatif, comporte des motifs encodés de secteurs opaques et transparents. Lorsque le disque tourne, les segments opaques bloquent la lumière alors que ceux où le verre est clair la laissent passer.

Ceci génère des impulsions d'onde carrée qui peuvent ensuite être interprétées comme position ou mouvement. En général, les encodeurs ont entre 100 et 6 000 segments par révolution.

Ceci correspond à une résolution de 3,6 degrés pour un encodeur à 100 segments et à une résolution de 0,06 degrés pour un encodeur à 6 000 segments.

Le même principe de fonctionnement est valable pour les encodeurs linéaires, si ce n'est que le disque rotatif est remplacé par une bande opaque stationnaire dont la surface comporte des fentes et que l'assemblage capteur/LED est fixé sur la partie mobile.

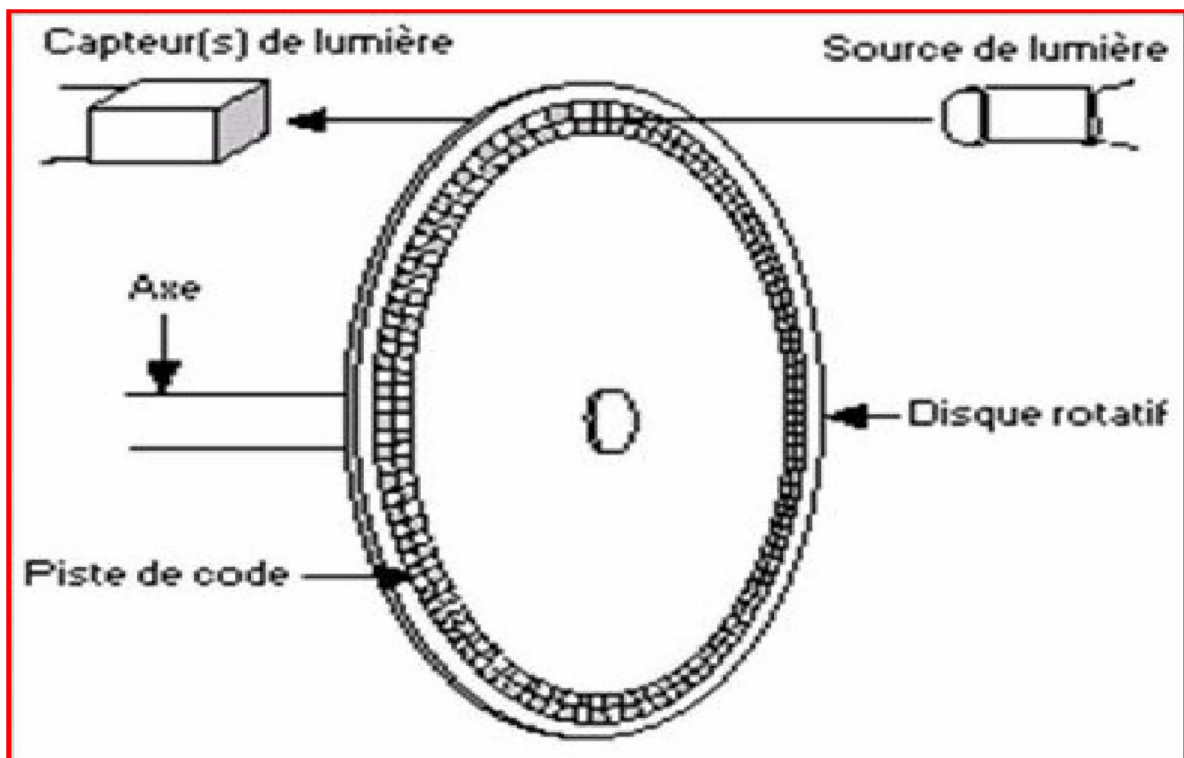


FIGURE IV.1. Les différents composants d'un encodeur rotatif.

Un encodeur ne générant qu'une série d'impulsions ne serait pas très utile puisqu'il ne pourrait pas indiquer la direction de la rotation. L'encodeur en quadrature comporte deux pistes de code dont les secteurs sont décalés de 90 degrés d'une piste à l'autre ; ceci permet d'avoir deux sorties de voie qui indiquent la position et la direction de la rotation. Si A devance B, par exemple, le disque tourne dans le sens des aiguilles d'une montre. Si B devance A, le disque

tourne dans le sens contraire. Par conséquent, en surveillant à la fois le nombre d'impulsions et les phases relatives des signaux A et B, vous pouvez suivre la position et la direction de la rotation.

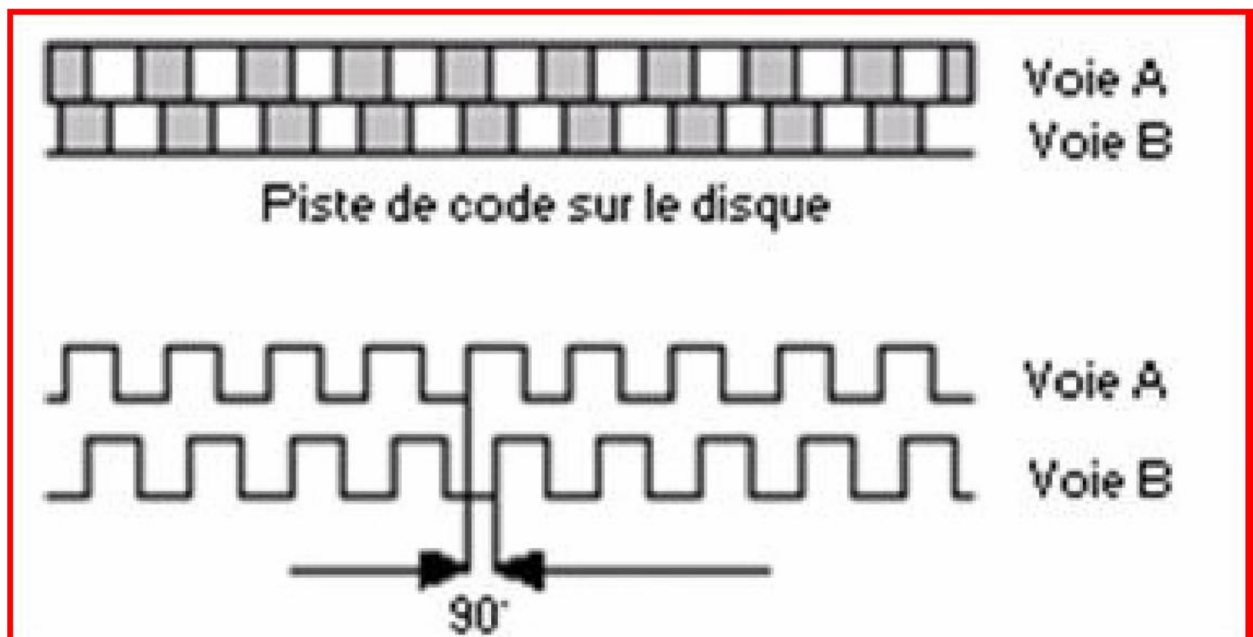


FIGURE IV.2. L'encodeur en quadrature.

Certains encodeurs en quadrature comportent aussi une troisième sortie de voie, appelée signal de référence ou zéro, qui renvoie une impulsion par révolution. On peut utiliser cette impulsion pour déterminer précisément une position de référence. Dans la plupart des encodeurs, ce signal correspond au terminal ou index.

IV.3.Régulation en vitesse :

Le choix du profil de vitesse est un problème similaire dans le cas du déplacement rectiligne et de la rotation pure. Il s'agit, à partir d'une vitesse et d'une accélération nulles, d'atteindre une nouvelle position avec également une vitesse et accélération nulles à l'arrivée.

Pour modifier cette vitesse il suffit juste de modifier le signal PWM.

Sachant que la roue tourne à 121 tours/min à 12V ça sera donc sa vitesse maximal

On aura alors :

$$V_{\max} = (121 \times 2\pi \times r) / 60.$$

Avec r : rayon de la roue qui est 100 mm.

On aura donc

$$V_{\max} = 1.12 \text{ m/s.}$$

Afin d'éviter le glissement des roues il est conseillé de diminuer cette vitesse pendant le déplacement.

IV.4.Régulation en position :

IV.4.1.Calcul du déplacement en ligne droite:

Etant donné que notre robot effectuera des déplacements en marche avant et en rotation, nous allons donc utiliser un seul canal de l'encodeur qui est le canal A, pour le comptage des impulsions les 2 pins counter_0 et counter_1 du module ARDUINO seront exploitées En

connaissant aussi le diamètre des roues, nous pourrions définir exactement la distance parcourue par notre robot ainsi que les angles de rotations.

En partant de la formule suivante, on peut calculer la distance parcourue, il suffit de multiplier l'angle obtenu par le diamètre de la roue $\times \pi$.

$$(\text{Angle de rotation}) = (\text{Nbre d'impulsions} / N) \times 360.$$

Où N est: le

nombre d'impulsions générées par l'encodeur à chaque révolution de l'axe.

Donc la distance D parcourue par la roue sera donnée par :

$$D = \text{angle de rotation} \times d \times \pi.$$

Où d est le diamètre de la roue.

IV.4.2.Calcul d'odométrie :

La position du robot est calculée au moyen d'une formule basée sur la vitesse angulaire et la vitesse d'avance du robot. Cette méthode a été choisie pour les raisons suivantes :

- vitesses de rotation des roues mesurables, et par conséquent vitesses angulaire et d'avance faciles à calculer .
- calculs réalisables dans le temps à disposition .
- facile à implémenter.

Les vitesses angulaires et d'avance ne sont pas directement mesurables. C'est la vitesse de chaque roue qui est mesurée à chaque instant.

On en déduit facilement les vitesses angulaires et d'avance.

La formule suivante permet d'obtenir la vitesse de rotation du robot :

$$\omega = (V_{\text{droite}} - V_{\text{gauche}}) / E$$

Où E est la distance entre les deux roues du robot. Une vitesse positive correspond à une rotation dans le sens trigonométrique positif. Nous estimons ensuite la vitesse d'avance du robot en faisant la moyenne entre les deux vitesses des roues :

$$V = (V_{\text{droite}} + V_{\text{gauche}}) / 2$$

Connaissant le temps de cycle et une estimation de la vitesse angulaire, on trouve

facilement l'angle effectué, qu'il suffit d'ajouter à l'ancien angle α_{old} pour connaître la nouvelle orientation α_{new} du robot. Concernant le changement de position, le principe est similaire. On détermine une vitesse d'avance du robot, qu'on considère comme constante durant le cycle précédent.

Là aussi on prend la moyenne entre la vitesse actuelle et la vitesse précédente. Une fois la distance parcourue calculée, il est nécessaire de connaître la direction du déplacement. On prend pour cela la moyenne entre l'angle actuel et l'angle précédent.

Les calculs permettant de mettre à jour la position sont présentés ci-après.

La figure suivant permet de comprendre intuitivement la méthode utilisée.

Localisation relative

Le principe de la localisation relative consiste à placer le robot dans un environnement et de lui faire effectuer une trajectoire. Connaissant la position initiale et avec les informations données grâce au encodeurs des roues, nous pouvons déterminer la position du robot à un instant quelconque.

Les schémas ci-dessous expliquent ce principe. La position du robot est ensuite déterminée à l'aide des équations mathématiques ci-dessous.

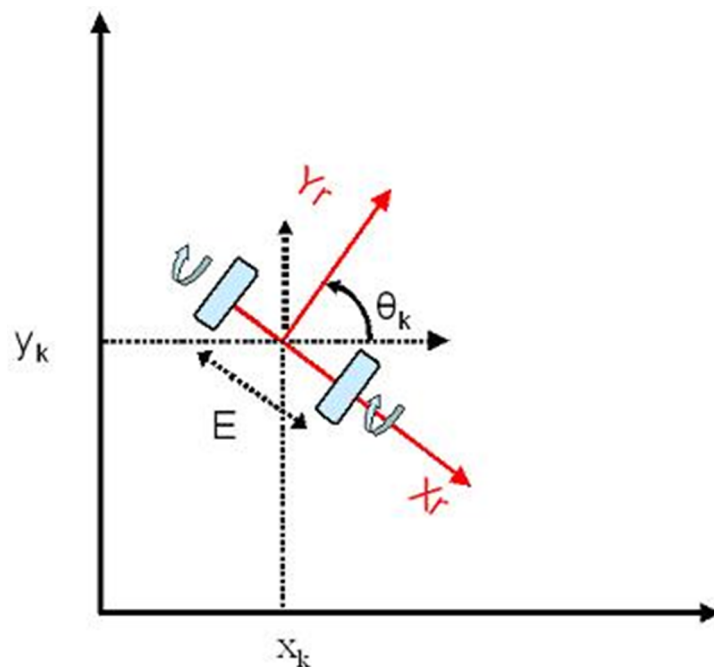


FIGURE IV.3 position du robot .

(X_r, Y_r) : repère attaché au robot

E : largeur du robot

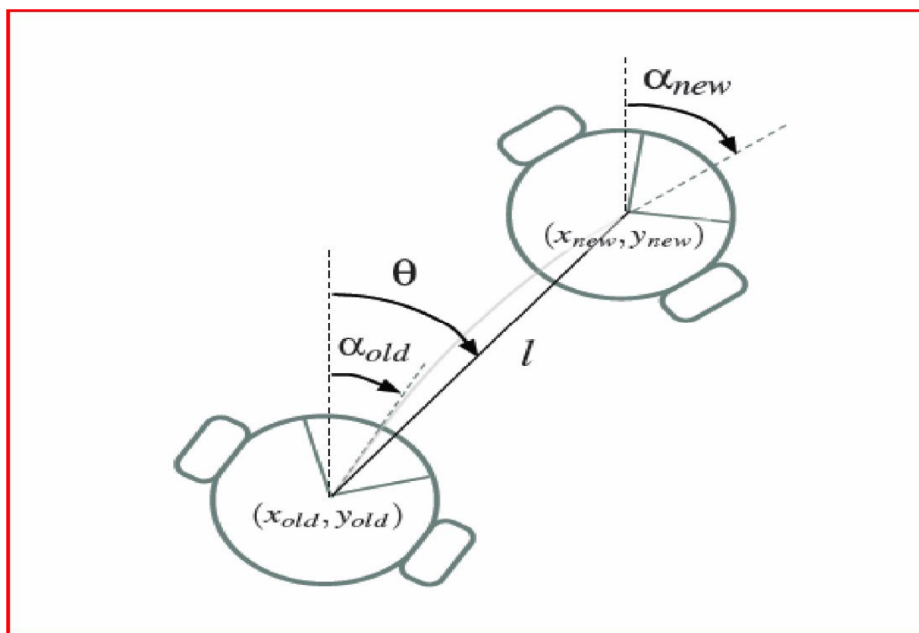
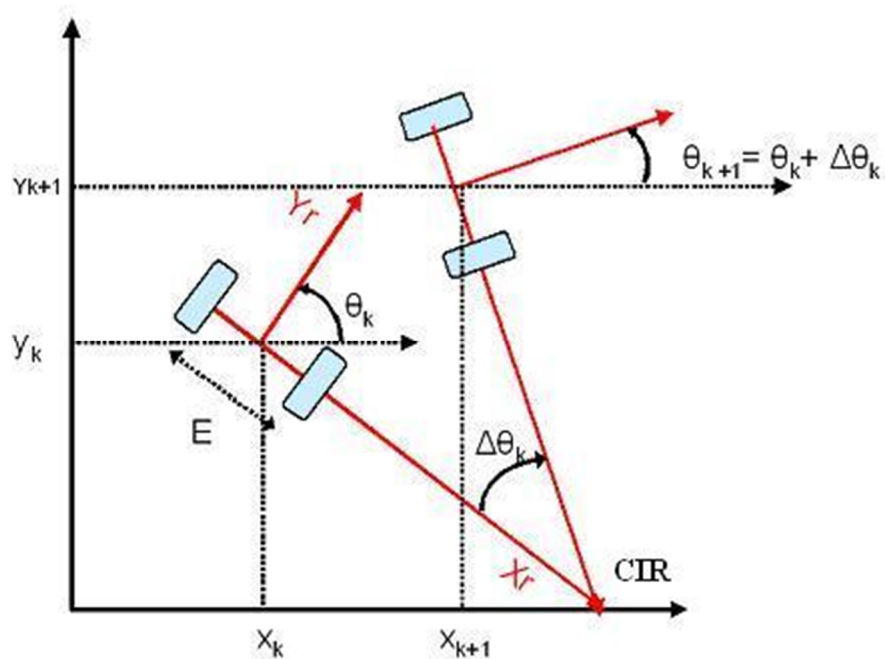


FIGURE IV.4. Estimation de la position absolue du robot.

$$x_{k+1} = x_k + \Delta D_k * \cos(\theta_k + \Delta\theta_k/2)$$

$$y_{k+1} = y_k + \Delta D_k * \sin(\theta_k + \Delta\theta_k/2)$$

Avec

$$- \Delta D_k = (\Delta d_d + \Delta d_g)/2$$

$$- \Delta\theta_k = (\Delta d_d - \Delta d_g)/E$$

Δd_d et Δd_g : déplacements élémentaires des roues droite et gauche

IV.5. Conclusion :

Nous avons vu dans ce chapitre des méthodes de calculs des différentes positions du robot.

En sachant pertinemment que ce qu'on aura en pratique est loin de ressembler à ce que

nous avons obtenu théoriquement, nous avons donc centré la majorité de notre travail sur le plan pratique, et en termine avec une conclusion general.

IV.1.Introduction :

L'odométrie est une technique permettant d'estimer la position d'un robot mobile en mouvement. Littéralement, il s'agit de la science de la mesure et du contrôle du mouvement d'un robot. La mesure se base pour cela sur la mesure individuelle des déplacements des roues du robot.

En effet, l'idée est assez simple : si l'on connaît le sens de rotation d'une roue, sa vitesse ainsi que sa durée de rotation et que cette mesure intervient sur toutes les roues du robot, alors on peut estimer l'emplacement du robot par rapport à son point de départ.

L'odométrie fonctionne relativement bien avec des robots munis de roues, elle est bruitée avec les robots utilisant des chenilles à cause du contact de ces dernières avec le sol qui est mal défini, et enfin elle est très mauvaise avec les robots utilisant des pattes pour le déplacement.

Le problème avec l'odométrie, c'est que les erreurs s'accumulent au fil du temps. Il est souvent intéressant de coupler ces mesures avec des mesures ponctuelles basées sur d'autres capteurs (ultrasons, infrarouge, lasers...).

IV.2 Principe de fonctionnement des encodeurs:

Un encodeur est un dispositif électromécanique capable de mesurer le mouvement ou la position. La plupart des encodeurs utilisent des capteurs optiques pour fournir des signaux électriques sous forme de trains d'impulsions, lesquels peuvent, à leur tour, être interprétés comme mouvement, direction ou position.

Les encodeurs rotatifs servent à mesurer le déplacement rotatif d'un axe. La figure n°1 montre les composants de base d'un encodeur rotatif :

une diode électroluminescente (LED), un disque et un capteur de lumière de l'autre côté du disque. Le disque, monté sur l'axe rotatif, comporte des motifs encodés de secteurs opaques et transparents. Lorsque le disque tourne, les segments opaques bloquent la lumière alors que ceux où le verre est clair la laissent passer.

Ceci génère des impulsions d'onde carrée qui peuvent ensuite être interprétées comme position ou mouvement. En général, les encodeurs ont entre 100 et 6 000 segments par révolution.

Ceci correspond à une résolution de 3,6 degrés pour un encodeur à 100 segments et à une résolution de 0,06 degrés pour un encodeur à 6 000 segments.

Le même principe de fonctionnement est valable pour les encodeurs linéaires, si ce n'est que le disque rotatif est remplacé par une bande opaque stationnaire dont la surface comporte des fentes et que l'assemblage capteur/LED est fixé sur la partie mobile.

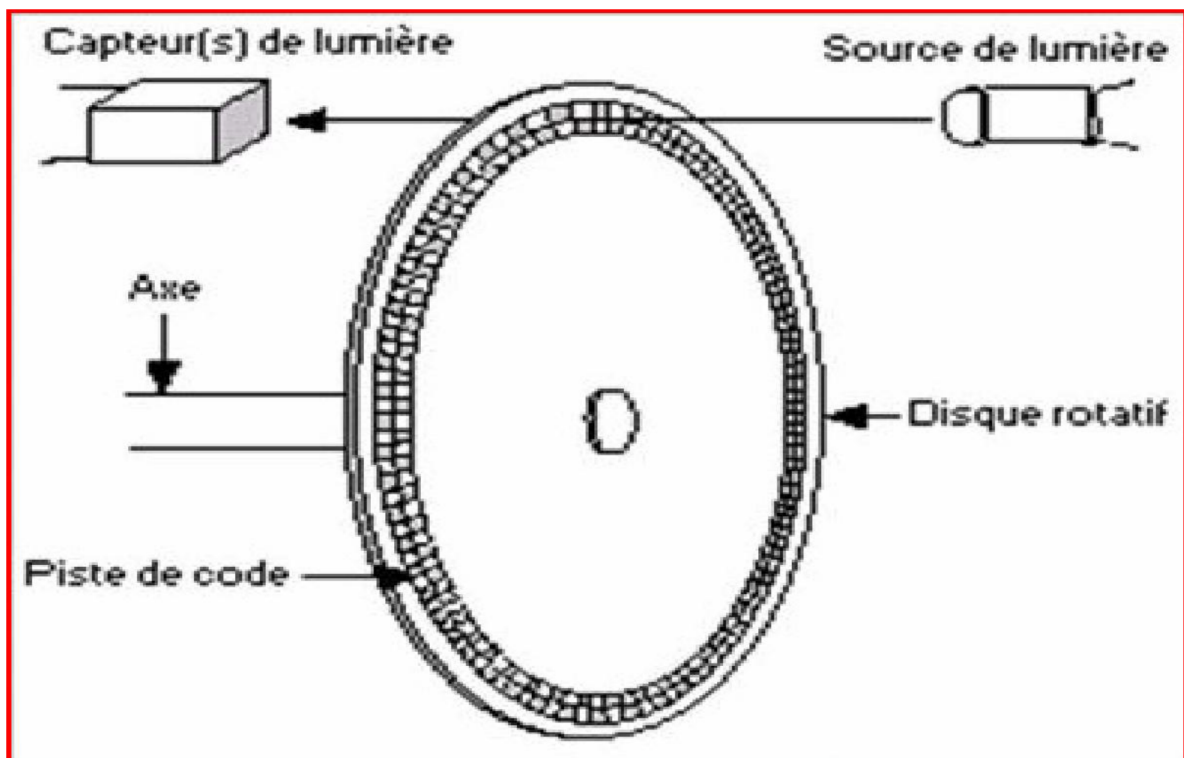


FIGURE IV.1. Les différents composants d'un encodeur rotatif.

Un encodeur ne générant qu'une série d'impulsions ne serait pas très utile puisqu'il ne pourrait pas indiquer la direction de la rotation. L'encodeur en quadrature comporte deux pistes de code dont les secteurs sont décalés de 90 degrés d'une piste à l'autre ; ceci permet d'avoir deux sorties de voie qui indiquent la position et la direction de la rotation. Si A devance B, par exemple, le disque tourne dans le sens des aiguilles d'une montre. Si B devance A, le disque

tourne dans le sens contraire. Par conséquent, en surveillant à la fois le nombre d'impulsions et les phases relatives des signaux A et B, vous pouvez suivre la position et la direction de la rotation.

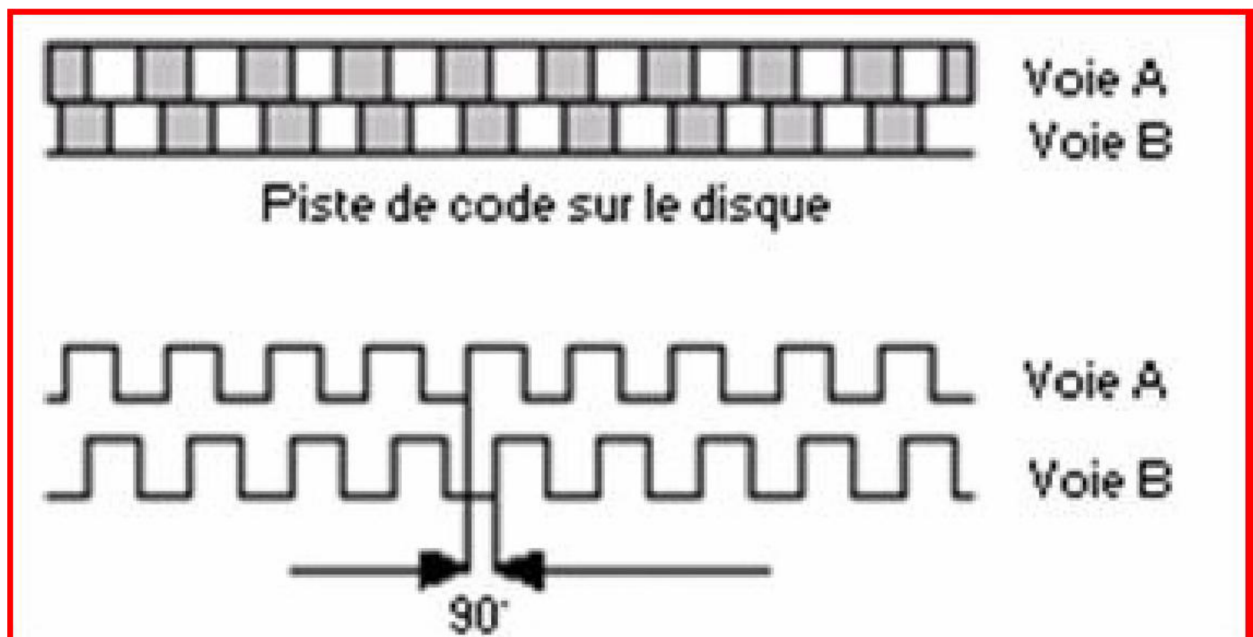


FIGURE IV.2. L'encodeur en quadrature.

Certains encodeurs en quadrature comportent aussi une troisième sortie de voie, appelée signal de référence ou zéro, qui renvoie une impulsion par révolution. On peut utiliser cette impulsion pour déterminer précisément une position de référence. Dans la plupart des encodeurs, ce signal correspond au terminal ou index.

IV.3.Régulation en vitesse :

Le choix du profil de vitesse est un problème similaire dans le cas du déplacement rectiligne et de la rotation pure. Il s'agit, à partir d'une vitesse et d'une accélération nulles, d'atteindre une nouvelle position avec également une vitesse et accélération nulles à l'arrivée.

Pour modifier cette vitesse il suffit juste de modifier le signal PWM.

Sachant que la roue tourne à 121 tours/min à 12V ça sera donc sa vitesse maximal

On aura alors :

$$V_{\max} = (121 \times 2\pi \times r) / 60.$$

Avec r : rayon de la roue qui est 100 mm.

On aura donc

$$V_{\max} = 1.12 \text{ m/s.}$$

Afin d'éviter le glissement des roues il est conseillé de diminuer cette vitesse pendant le déplacement.

IV.4.Régulation en position :

IV.4.1.Calcul du déplacement en ligne droite:

Etant donné que notre robot effectuera des déplacements en marche avant et en rotation, nous allons donc utiliser un seul canal de l'encodeur qui est le canal A, pour le comptage des impulsions les 2 pins counter_0 et counter_1 du module ARDUINO seront exploitées En

connaissant aussi le diamètre des roues, nous pourrons définir exactement la distance parcourue par notre robot ainsi que les angles de rotations.

En partant de la formule suivante, on peut calculer la distance parcourue, il suffit de multiplier l'angle obtenu par le diamètre de la roue $\times \pi$.

$$(\text{Angle de rotation}) = (\text{Nbre d'impulsions} / N) \times 360.$$

Où N est: le

nombre d'impulsions générées par l'encodeur à chaque révolution de l'axe.

Donc la distance D parcourue par la roue sera donnée par :

$$D = \text{angle de rotation} \times d \times \pi.$$

Où d est le diamètre de la roue.

IV.4.2.Calcul d'odométrie :

La position du robot est calculée au moyen d'une formule basée sur la vitesse angulaire et la vitesse d'avance du robot. Cette méthode a été choisie pour les raisons suivantes :

- vitesses de rotation des roues mesurables, et par conséquent vitesses angulaire et d'avance faciles à calculer .
- calculs réalisables dans le temps à disposition .
- facile à implémenter.

Les vitesses angulaires et d'avance ne sont pas directement mesurables. C'est la vitesse de chaque roue qui est mesurée à chaque instant.

On en déduit facilement les vitesses angulaires et d'avance.

La formule suivante permet d'obtenir la vitesse de rotation du robot :

$$\omega = (V_{\text{droite}} - V_{\text{gauche}}) / E$$

Où E est la distance entre les deux roues du robot. Une vitesse positive correspond à une rotation dans le sens trigonométrique positif. Nous estimons ensuite la vitesse d'avance du robot en faisant la moyenne entre les deux vitesses des roues :

$$V = (V_{\text{droite}} + V_{\text{gauche}}) / 2$$

Connaissant le temps de cycle et une estimation de la vitesse angulaire, on trouve

facilement l'angle effectué, qu'il suffit d'ajouter à l'ancien angle α_{old} pour connaître la nouvelle orientation α_{new} du robot. Concernant le changement de position, le principe est similaire. On détermine une vitesse d'avance du robot, qu'on considère comme constante durant le cycle précédent.

Là aussi on prend la moyenne entre la vitesse actuelle et la vitesse précédente. Une fois la distance parcourue calculée, il est nécessaire de connaître la direction du déplacement. On prend pour cela la moyenne entre l'angle actuel et l'angle précédent.

Les calculs permettant de mettre à jour la position sont présentés ci-après.

La figure suivant permet de comprendre intuitivement la méthode utilisée.

Localisation relative

Le principe de la localisation relative consiste à placer le robot dans un environnement et de lui faire effectuer une trajectoire. Connaissant la position initiale et avec les informations données grâce au encodeurs des roues, nous pouvons déterminer la position du robot à un instant quelconque.

Les schémas ci-dessous expliquent ce principe. La position du robot est ensuite déterminée à l'aide des équations mathématiques ci-dessous.

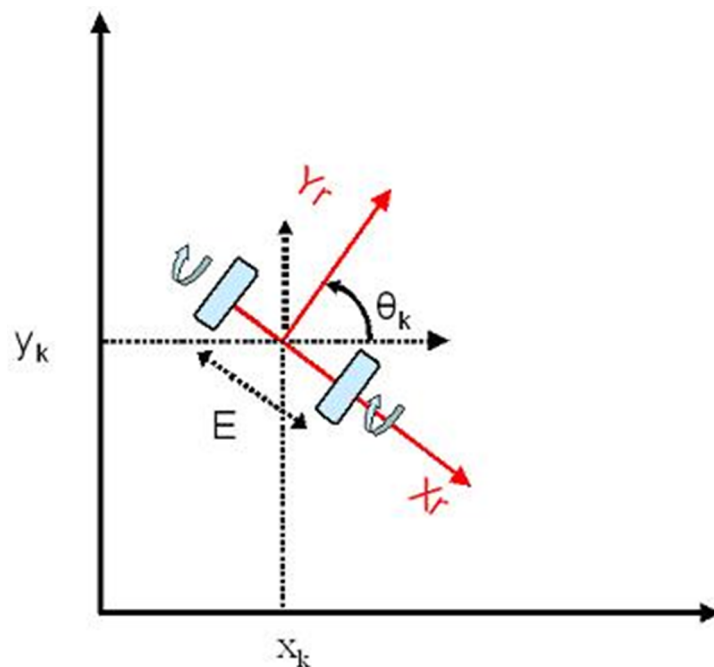


FIGURE IV.3 position du robot .

(X_r, Y_r) : repère attaché au robot

E : largeur du robot

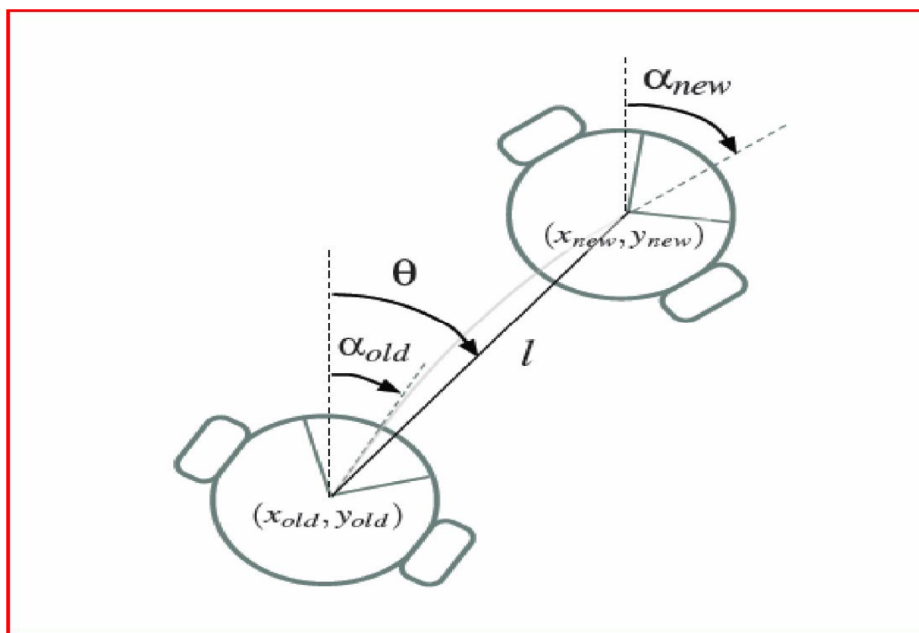
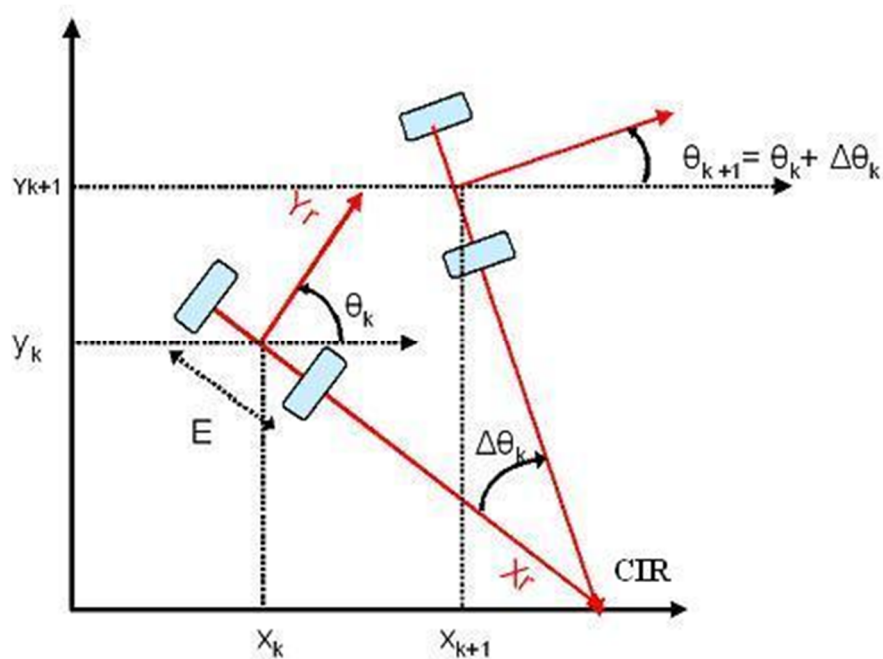


FIGURE IV.4. Estimation de la position absolue du robot.

$$x_{k+1} = x_k + \Delta D_k * \cos(\theta_k + \Delta\theta_k/2)$$

$$y_{k+1} = y_k + \Delta D_k * \sin(\theta_k + \Delta\theta_k/2)$$

Avec

$$- \Delta D_k = (\Delta d_d + \Delta d_g)/2$$

$$- \Delta\theta_k = (\Delta d_d - \Delta d_g)/E$$

Δd_d et Δd_g : déplacements élémentaires des roues droite et gauche

IV.5. Conclusion :

Nous avons vu dans ce chapitre des méthodes de calculs des différentes positions du robot.

En sachant pertinemment que ce qu'on aura en pratique est loin de ressembler à ce que

nous avons obtenu théoriquement, nous avons donc centré la majorité de notre travail sur le plan pratique, et en termine avec une conclusion general.

V.I.Introduction :

Pour mettre en oeuvre tout ce travail, le module ARDUINO doit être programmé selon le cahier de charge que ona fait.

. Le langage utilisé pour programmé c'est le langage (C).

V.2.communication avec les capteurs :

V.2.1.Evitement d'obstacle :

Le capture infrarouges GP2D15 numériques qui génèrent une interruption dès que le robot se rapproche de 24 cm d'un obstacle. Quand cela arrive il attend un certain temps (5 secondes),puis s'il y a toujours un obstacle en face de lui il change complètement de direction grâce à la distance récupérée par les encodeurs.

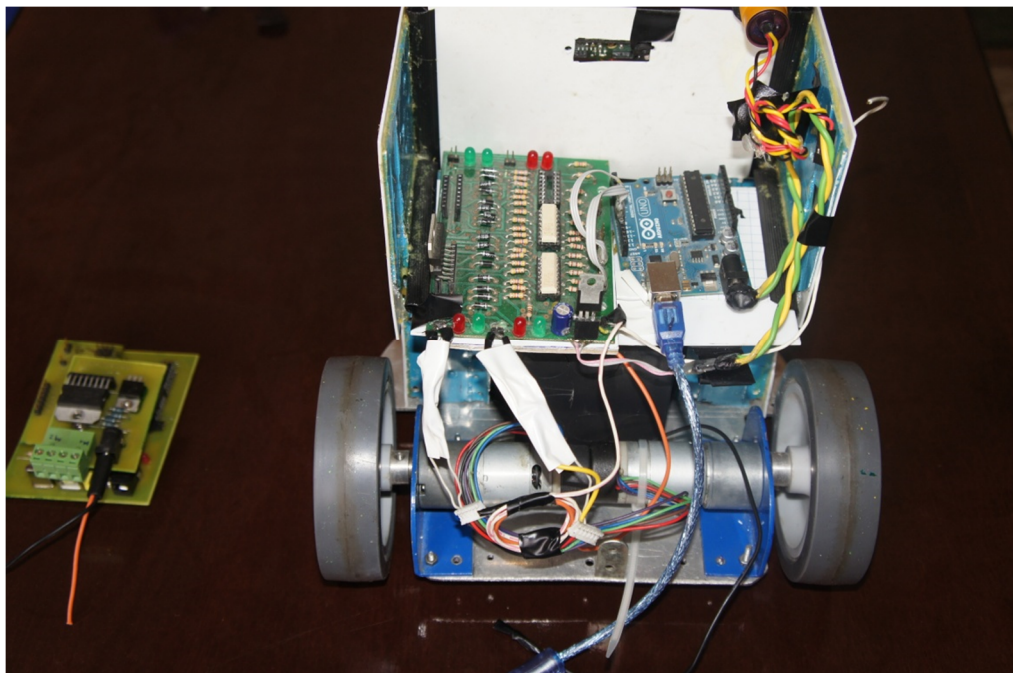


Figure V.1. le robot face arrière.

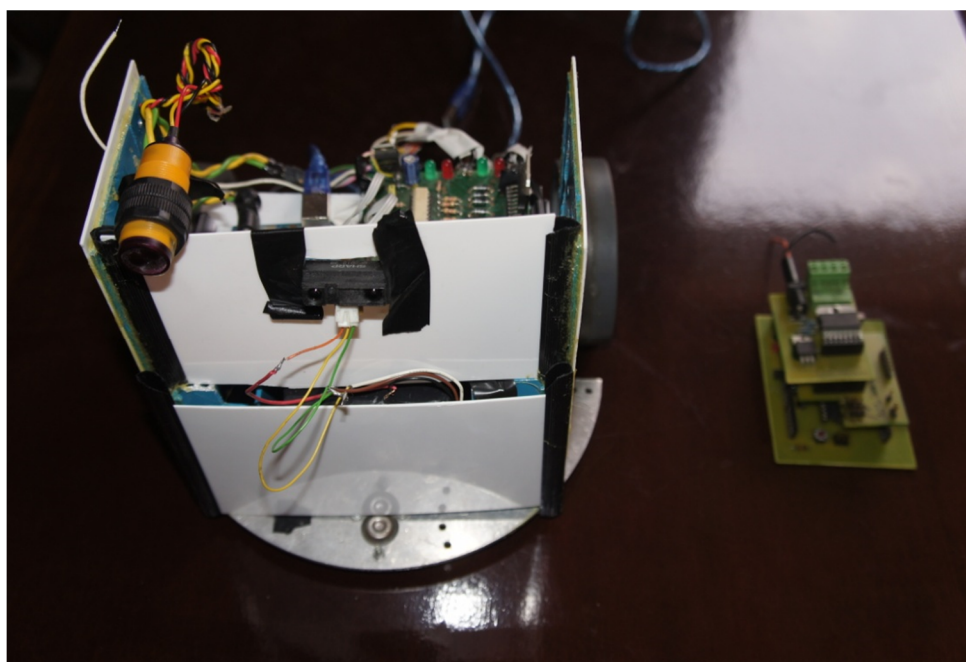


Figure V.2.le robot face avant.

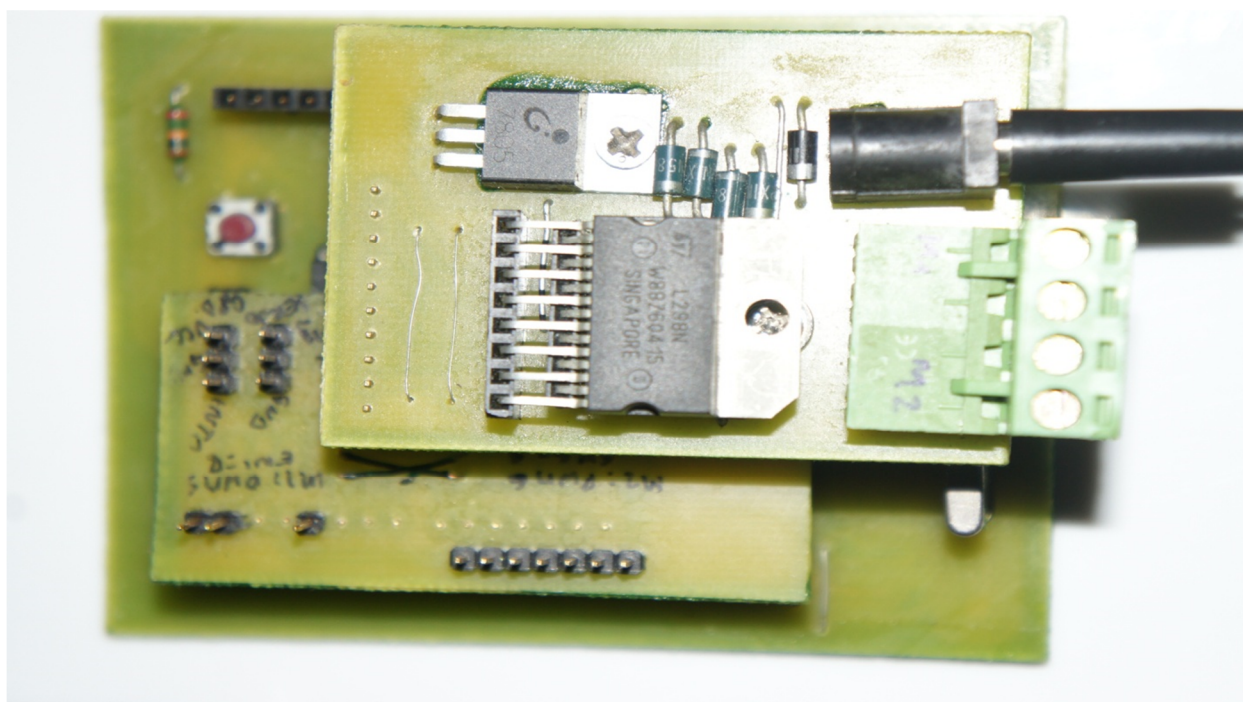


Figure V.3.les carte electronique prépare

V.I.Introduction :

Pour mettre en oeuvre tout ce travail, le module ARDUINO doit être programmé selon le cahier de charge que ona fait.

. Le langage utilisé pour programmé c'est le langage (C).

V.2.communication avec les capteurs :**V.2.1.Evitement d'obstacle :**

Le capture infrarouges GP2D15 numériques qui génèrent une interruption dès que le robot se rapproche de 24 cm d'un obstacle. Quand cela arrive il attend un certain temps (5 secondes),puis s'il y a toujours un obstacle en face de lui il change complètement de direction grâce à la distance récupérée par les encodeurs.

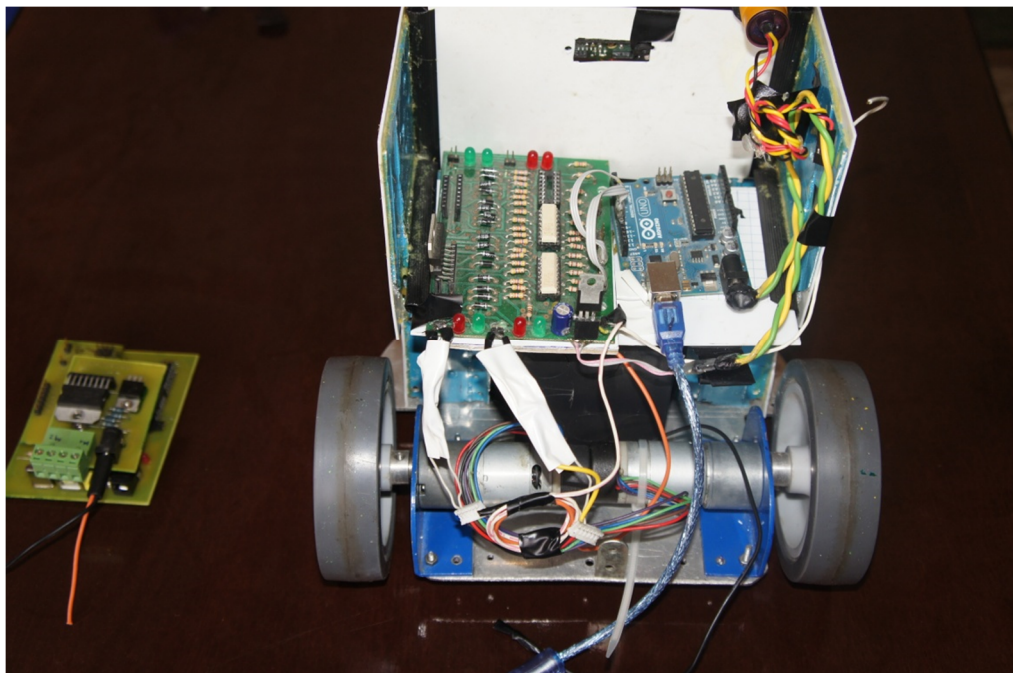


Figure V.1. le robot face arrière.

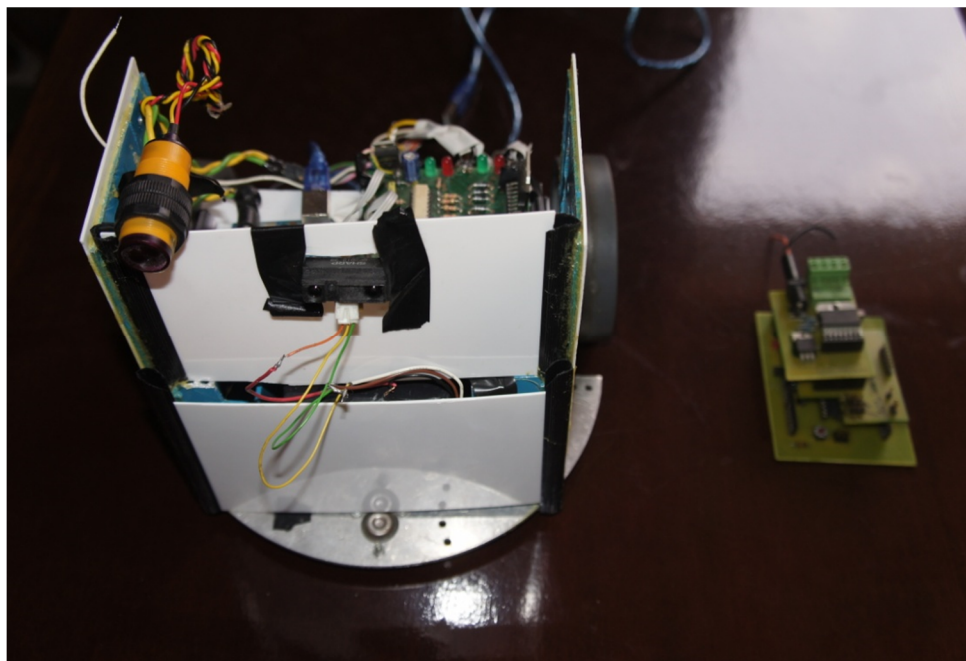


Figure V.2.le robot face avant.

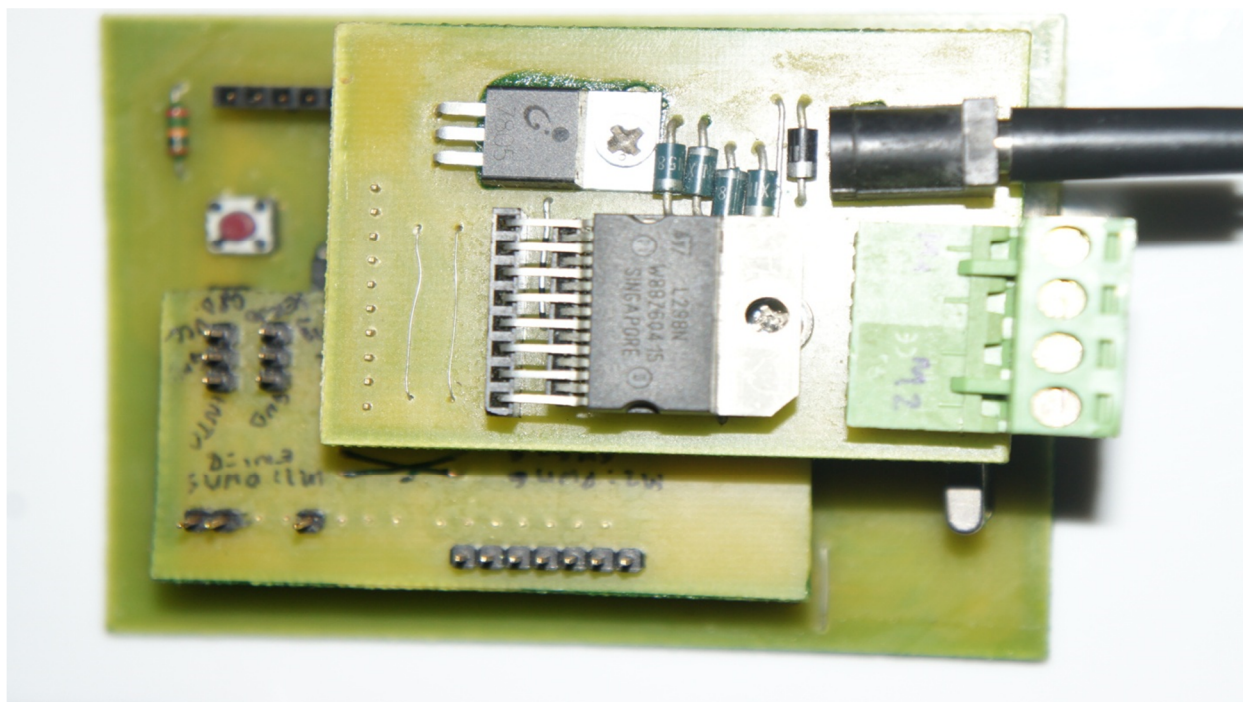


Figure V.3.les carte electronique prépare

V.I.Introduction :

Pour mettre en oeuvre tout ce travail, le module ARDUINO doit être programmé selon le cahier de charge que on a fait.

. Le langage utilisé pour programmé c'est le langage (C).

V.2.communication avec les capteurs :

V.2.1.Evitement d'obstacle :

Le capture infrarouges GP2D15 numériques qui génèrent une interruption dès que le robot se rapproche de 24 cm d'un obstacle. Quand cela arrive il attend un certain temps (5 secondes), puis s'il y a toujours un obstacle en face de lui il change complètement de direction grâce à la distance récupérée par les encodeurs.

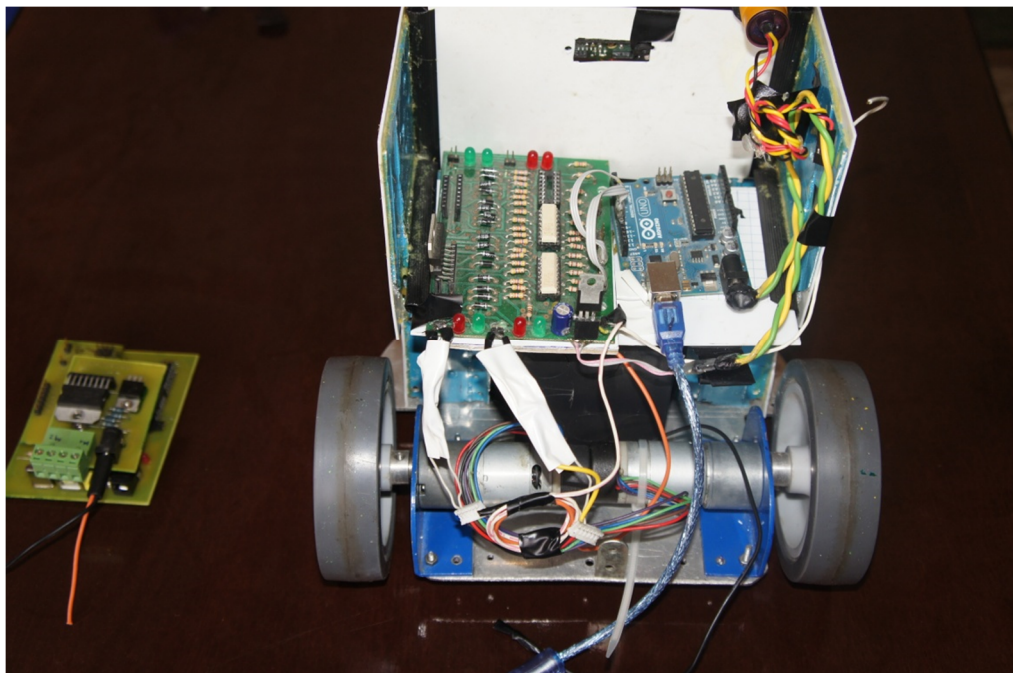


Figure V.1. le robot face arrière.

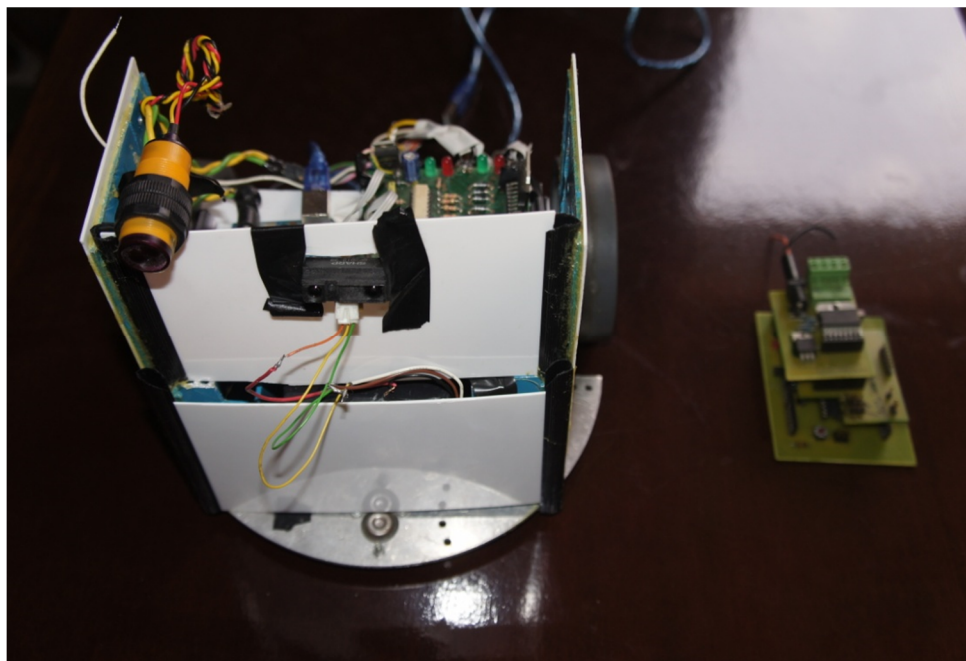


Figure V.2.le robot face avant.

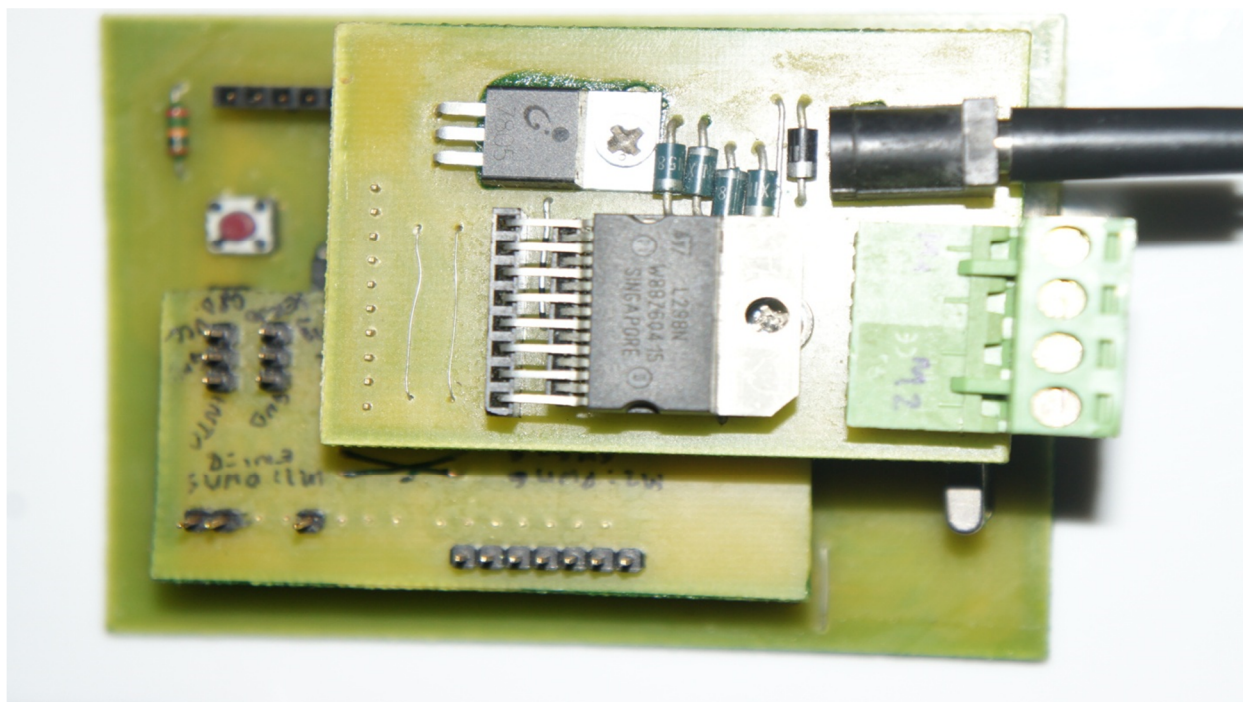


Figure V.3.les carte electronique prépare

Conclusion générale

Dans ce mémoire nous avons fait la conception et la réalisation d'un robot mobile. Nous avons avant tout, appris à bâtir un projet en respectant un cahier des charges stricte, et ce par la construction d'une stratégie idée par idée jusqu'à maturité.

La partie réalisation nous a été d'un profit inestimable, notamment pour la partie mécanique, qui a nécessitée l'usinage des pièces, la soudure, et différentes fonctions de tournage. De même pour la partie électronique et informatique, dans lesquelles nous avons pu mettre en pratique nos connaissances théoriques acquises tout au long de notre cursus de formation.

Dans un premier temps nous avons acquis quelques notions de base sur la robotique mobile , Deuxièmement nous avons conçu et réalisé notre système constitué d'une partie mécanique, commandée par différentes cartes électroniques.. Nous avons travaillé avec le microcontrôleur arduino sur notre système, pour contrôle et générer les signaux PWM qui en est besoin dans notre robot.

La réalisation de ce système nous a permis de nous familiariser avec le monde du traitement d'image, le monde de la robotique . Elle nous a permis aussi de travailler avec des nouveaux outils, tel que la bibliothèque du traitement d'image OPEN CV (Open Computer Vision), qui contient toutes les fonctions nécessaires, pour l'acquisition et le traitement d'image en temps réel. Nous avons utilisé l'environnement Visual c++ pour la programmation, et avec le logiciel MATLAB qui contient plusieurs fonctions pour le contrôle du système. Ce projet nous a permis aussi de se familiariser avec les microcontrôleurs, et avec le logiciel ARDUINO de la programmation, qui est basé sur le langage C.

Nous pouvons améliorer ce système dans le futur, faire des comparaisons entre les différents microcontrôleurs (rovin ,pic, automatte,.....)

Conclusion générale

Dans ce mémoire nous avons fait la conception et la réalisation d'un robot mobile. Nous avons avant tout, appris à bâtir un projet en respectant un cahier des charges stricte, et ce par la construction d'une stratégie idée par idée jusqu'à maturité.

La partie réalisation nous a été d'un profit inestimable, notamment pour la partie mécanique, qui a nécessitée l'usinage des pièces, la soudure, et différentes fonctions de tournage. De même pour la partie électronique et informatique, dans lesquelles nous avons pu mettre en pratique nos connaissances théoriques acquises tout au long de notre cursus de formation.

Dans un premier temps nous avons acquis quelques notions de base sur la robotique mobile , Deuxièmement nous avons conçu et réalisé notre système constitué d'une partie mécanique, commandée par différentes cartes électroniques.. Nous avons travaillé avec le microcontrôleur arduino sur notre système, pour contrôle et générer les signaux PWM qui en est besoin dans notre robot.

La réalisation de ce système nous a permis de nous familiariser avec le monde du traitement d'image, le monde de la robotique . Elle nous a permis aussi de travailler avec des nouveaux outils, tel que la bibliothèque du traitement d'image OPEN CV (Open Computer Vision), qui contient toutes les fonctions nécessaires, pour l'acquisition et le traitement d'image en temps réel. Nous avons utilisé l'environnement Visual c++ pour la programmation, et avec le logiciel MATLAB qui contient plusieurs fonctions pour le contrôle du système. Ce projet nous a permis aussi de se familiariser avec les microcontrôleurs, et avec le logiciel ARDUINO de la programmation, qui est basé sur le langage C.

Nous pouvons améliorer ce système dans le futur, faire des comparaisons entre les différents microcontrôleurs (rovin ,pic, automatte,.....)

Conclusion générale

Dans ce mémoire nous avons fait la conception et la réalisation d'un robot mobile. Nous avons avant tout, appris à bâtir un projet en respectant un cahier des charges stricte, et ce par la construction d'une stratégie idée par idée jusqu'à maturité.

La partie réalisation nous a été d'un profit inestimable, notamment pour la partie mécanique, qui a nécessitée l'usinage des pièces, la soudure, et différentes fonctions de tournage. De même pour la partie électronique et informatique, dans lesquelles nous avons pu mettre en pratique nos connaissances théoriques acquises tout au long de notre cursus de formation.

Dans un premier temps nous avons acquis quelques notions de base sur la robotique mobile , Deuxièmement nous avons conçu et réalisé notre système constitué d'une partie mécanique, commandée par différentes cartes électroniques.. Nous avons travaillé avec le microcontrôleur arduino sur notre système, pour contrôle et générer les signaux PWM qui en est besoin dans notre robot.

La réalisation de ce système nous a permis de nous familiariser avec le monde du traitement d'image, le monde de la robotique . Elle nous a permis aussi de travailler avec des nouveaux outils, tel que la bibliothèque du traitement d'image OPEN CV (Open Computer Vision), qui contient toutes les fonctions nécessaires, pour l'acquisition et le traitement d'image en temps réel. Nous avons utilisé l'environnement Visual c++ pour la programmation, et avec le logiciel MATLAB qui contient plusieurs fonctions pour le contrôle du système. Ce projet nous a permis aussi de se familiariser avec les microcontrôleurs, et avec le logiciel ARDUINO de la programmation, qui est basé sur le langage C.

Nous pouvons améliorer ce système dans le futur, faire des comparaisons entre les différents microcontrôleurs (rovin ,pic, automatte,.....)

Chapitre I : Notions de robotique et de robots mobiles.

Chapitre I : Notions de robotique et de robots mobiles.

Chapitre I : Notions de robotique et de robots mobiles.

CHAPITRE II :
PARTIE
MECANIQUE.

CHAPITRE II :
PARTIE
MECANIQUE.

CHAPITRE II :
PARTIE
MECANIQUE.

CHAPITRE III :

PARTIE

ELECTRONIQUE.

CHAPITRE III :

PARTIE

ELECTRONIQUE.

CHAPITRE III :

PARTIE

ELECTRONIQUE.

CHAPITRE IV : **ODOMETRIE.**

CHAPITRE IV : **ODOMETRIE.**

CHAPITRE IV : **ODOMETRIE.**