

TABLE DES MATIERES

REMERCIEMENT

DEDICACES

LISTES DES FIGURES

LISTE D'ABREVIATIONS

INTRODUCTION GENERALE 1

CHAPITRE 01 : Introduction à la robotique mobile 2

1. Introduction	2
2. La robotique	2
3. Historique	3
4. Domaines d'application	6
4.1.Domaine industriel	6
4.2.Domaine militaire	7
4.3.Domaine médical	10
4.4.Domaine domestique	11
4.5.Domaine spatiale	12
5. Type des robots	12
6. Architecture d'un robot mobile	13
7. Conclusion	15

CHAPITRE 02 : Robot différentiel 16

1. Introduction	16
2. Choix du robot	16
3. Robot différentiel	17
4. Modèle géométrique	18
5. Conception mécanique	21
5.1.Cahier de charge	21
6. Conception électrique	25
6.1.Structure global	26
6.2.Alimentation	26
7. Architecture de contrôle	27

7.1. Contrôle du système	28
8. Détection des obstacles	28
9. Localisation et cartographie	32
9.1. Marier les données multi capteurs et cartographie	32
10. Suivi de trajectoire	32
11. Evitement des obstacles	32
12. Méthode de navigation	33
12.1. Navigation métrique	33
12.2. Navigation géométrique	33
12.3. Navigation topologique	34
12.4. Navigation par action associé à un lieu	34
12.5. Navigation par guidage	34
12.6. Navigation par approche d'un objet	34
13. Conclusion	35
CHAPITRE 03 : Conception, simulation et réalisation	36
1. Introduction	36
2. La conception du robot	36
3. Cahier de charge	37
4. Les composants hardwares	37
4.1. Carte Arduino UNO	38
4.2. Circuit L293D	40
4.3. Capteur de distance	40
4.4. Module bluetooth	42
4.5. Plaque d'essais et câbles jumpers	43
4.6. Moteurs à courant continu	44
4.7. Batteries d'alimentation	45
4.8. Châssis de véhicule	46
5. MODE 01 : Line follower robot	47
5.1. Simulation du line follower par V-REP Pro EDU	48
5.2. Circuit du suiveur de ligne	51
5.3. Programme Arduino	51
5.4. Simulation du système suiveur de ligne par ISIS	53
6. MODE 02 : Remote controlled via Blueooth	54
6.1. Circuit Arduino-Bluetooth HC-05	55
6.2. Circuit RC via Blueooth	56
6.3. Programme Arduino	57

6.4.Montage du RC via Bluetooth	59
7. MODE 03 : Intelligent & Obstacle avoidance	59
7.1.Circuit d'évitement des obstacles	60
7.2.Programme du teste	61
7.3.Simulation du capteur ultrason et Arduino sur ISIS	62
7.4.Circuit du robot intelligent	62
7.5.Programme Arduino du robo intelligent	63
7.6.Simulation du robot intelligent par ISIS	65
7.7.Montage du robot intelligent	66
8. Conclusion	66
CONCLUSION GENERALE	67
REFERENCES BIBLIOGRAPHIE	68
ANNEXES	69

REMERCIEMENTS

Nous tenons à remercier Mme GHOMRI Latéfa, Maître de conférences à l'Université de Tlemcen, pour l'honneur qu'il nous a fait en acceptant de présider le jury.

Nous remercions vivement Mme HOUBAD Yamina, Maître assistante à l'Université de Tlemcen et Mme KHEDIM Amaria, Maître assistante à l'Université de Tlemcen, pour avoir accepté d'examiner notre travail et l'intérêt qu'ils y ont porté.

Nous adressons nos vifs remerciements à monsieur MKADDER Mohammed El Amine, Ingénieur de laboratoire de recherche à l'Université de Tlemcen, pour nous avoir co-encadré et accueillie dans les laboratoires de recherche et pour toute l'aide et le soutien qu'il nous a apporté.

Nous tenons à exprimer notre profonde gratitude à Monsieur HASSAM Ahmed, Maître de conférences à l'Université de Tlemcen, pour avoir encadré et dirigé ce travail, pour toute l'aide et le soutien qu'il nous a apporté et pour sa patience durant la période de mémoire, ainsi que pour les nombreuses discussions inspirationnelles que nous avons eues.

Enfin, nous tenons à remercier tout particulièrement nos parents, pour leur soutien inconditionnel tout au long de ces longues années d'études, et aussi tous nos collègues et amis de la filière pour la solidarité et le soutien moral dont ils ont toujours fait preuve.

DEDICACES

Je dédie ce travail à :

Mes parents, que dieu me les garde,

A mes sœurs Amina, Radhia et Nafissa,

A mes frères Oussama et Aimen,

A la Mémoire de ma Grande mère, et mon Grand père

A ma chère Hanane

A Mes Amies : Rekia, Hamza, Walid, Nori, Amine, et Cheikh

*Et les membres du Royaume B18 : Ahmed, Dali, Mohammed et
Brahim*

A Mes collègues de la filière

A tous mes professeurs au long de ma carrière académique

...

YOUSFI Hadj Seddik

LISTE DES FIGURES

Fig.1.1 La tortue de Grey Walter.....	3
Fig.1.2 Robot Beast.....	4
Fig.1.3 Shakey de Stanford.....	4
Fig.1.4 Stanford Cart.....	5
Fig.1.5 Robot Hilare.	5
Fig.1.6 Genghis de Rodney Brooks.....	6
Fig.1.7 Bras souderur ABB.....	7
Fig.1.8 Robot éclaireur.....	8
Fig.1.9 Drone à courte portée.....	9
Fig.1.10 Drone tactique.....	9
Fig.1.11 Robot Mule DARPA.....	10
Fig.1.12 Robot Da Vinci.....	11
Fig.1.13 Robot aspirateur.....	11
Fig.1.14 Rover du Mars.....	12
Fig.2.1 Robot mobile différentiel.....	18
Fig.2.2 Cinématique du rebot différentiel.....	19
Fig.2.3 La motorisé d'un rebot mobile	20
Fig.2.4 Exemple d'un châssis	23
Fig.2.5 Vue de profil des roues motrices	24
Fig.2.6 Vue dessous des roues	25
Fig.2.7 La structure globale électrique et électronique	26
Fig.2.8 Différents types de contrôleurs	28
Fig.2.9 Principe de fonctionnement des capteurs IR	30
Fig.2.10 Capteur ultrason	30

Fig.2.11 Zone aveugle d'un capteur ultrason	31
Fig.2.12 Télémètre NEATO	31
Fig.3.1 Schéma des différentes unités du robot	36
Fig.3.2 Carte Arduino Uno	38
Fig.3.3 Propriétés de la carte Arduino Uno	39
Fig. 3.4 Circuit L293D et son schéma	40
Fig.3.5 Capteur Ultrason HC-SR04	41
Fig.3.6 Fonctionnement du capteur ultrason	41
Fig.3.7 Capteur CNY70	42
Fig.3.8 Module Bluetooth HC-05	43
Fig.3.9 Breadboard microtivity IB401 & Jumpers	44
Fig.3.10 Les deux moteurs DC	45
Fig.3.11 Batteries d'alimentation	45
Fig.3.12 Châssis et roues du robot	46
Fig.3.13 Création de routage par V-REP	48
Fig.3.14 Routage avec des virages ouverts	48
Fig.3.15 Routage finale du rebot	49
Fig.3.16 LineTracer avec ses trois capteurs	50
Fig.3.17 Simulation du line follower sur V-Rep	50
Fig.3.18 Montage suiveur de ligne sur Fritzing	51
Fig.3.19 Simulation du suiveur de ligne par ISIS	54
Fig.3.20 Montage bluetooth-Arduino	55
Fig.3.21 Montage réel du bluetooth avec Arduino	56
Fig.3.22 Montage du RC via bleutooth	56
Fig.3.23 Robot RC	59
Fig.3.24 Schéma du capteur ultrason avec Arduino	60
Fig.3.25 Montage réel du capteur ultrason avec Arduino	61

Liste des figures

Fig.3.26 Simulation ISIS du capteur ultrason et Arduino	62
Fig.3.27 Circuit du robot intelligent par Fritzing	63
Fig.3.28 Simulation du robot intelligent par ISIS	65
Fig.3.29 Montage final du robot intelligent	66

LISTE DES ABREVIATIONS

AMR: Automated mobile robot

AGV: Auto-guided vehicle

UGV: Unmanned ground vehicles

UAV: unmanned aerial vehicle

MIT: Massachusetts Institute of Technology

ICC: Instantaneous Center of Curvature

CC: Courant continu

DC: Direct current

SLA: Sealed Lead Acid

GPS: Global positioning system

RC: Remote controlled

ICSP: In-Circuit Serial Programming

PWM: Pulse width modulation

DF: Decision factor

Introduction générale

Les entreprises viennent dans un environnement concurrentiel dynamique et imprévu à cause du changement des exigences des clients comme la personnalisation et la date due d'obtention des produits ou services, ce qui rend l'aptitude de s'adapter à ces changements crucial et un paramètre important dans l'équation de réussite et d'échec.

La robotique est considérée comme une discipline indispensable dans l'industrie moderne car elle offre des opportunités diverses. Son apport a un lien direct avec le domaine opérationnel des systèmes de production qui peut être un facteur crucial affectant le succès du système global soit au terme de précision, de cout ou de temps opératoire.

L'un des aspects de cet apport est la robotique mobile, qui est le thème de notre projet fin d'étude. L'importance de ce domaine est due à son déploiement dans les systèmes manufacturiers récents comme les systèmes de production flexibles (FMS) où un robot mobile parcourt un routage pour assurer la personnalisation des produits, et dans les méthodes de livraison modernes comme celle établie par l'empire Amazon pour délivrer des produits commandés à sa clientèle.

Le premier chapitre présente une introduction à la robotique mobile en donnant quelque généralité sur le domaine, les points marquants de son histoire et ses types et domaine d'application.

Le deuxième chapitre fournit une compréhension du type différentiel en donnant son modèle géométrique et dynamique. On a présenté la conception mécanique avec ses composantes d'une manière générale, la conception électronique globale puis les aspects de contrôle comme les stratégies de navigation et le système sensoriel.

Le troisième chapitre est dédié à la réalisation des trois modes du robot. Une visualisation du comportement suiveur de ligne avait lieu par logiciel V-Rep. Pour la suite du chapitre on a simulé le circuit de chaque mode par ISIS pour confirmer sa performance avant l'installation du circuit réel.

Une conclusion générale résume, à la fin, l'ensemble du projet et fournit des perspectives d'amélioration de ce travail.

Chapitre

01

1. Introduction :

L'objet de la robotique est l'automatisation de systèmes mécaniques. En dotant le système de capacités de perception, d'action et de décision, l'objectif est de lui permettre d'interagir rationnellement avec son environnement, et de façon autonome. La robotique est un domaine de recherche qui se situe au carrefour de l'intelligence artificielle, de l'automatique, de l'informatique et de la perception par ordinateur ; cette interdisciplinarité est à l'origine d'une certaine complexité. Des applications dans des domaines aussi variés que l'industrie manufacturière, le spatial, l'automobile ou plus récemment les loisirs et le secteur médical, démontrent aujourd'hui l'intérêt économique et sociale.

Ce chapitre a pour but d'introduire le sujet de ce travail en situant brièvement quelques généralités concernant la robotique mobile en général ainsi que tous les points devant être abordés pour une meilleure compréhension de la navigation en robotique mobile. Ce chapitre commence par une étude des robots mobiles, de leur architecture et des différents concepts nécessaires au problème de la navigation. Nous présentons ensuite une introduction aux différentes méthodes de localisation permettant d'estimer la configuration d'un robot.

2. La robotique :

La robotique est l'ensemble des techniques pour concevoir et réaliser des machines automatiques ou simplement des robots. Un système robotique appartient aux familles des systèmes mécatroniques.

- Système mécatronique

La mécatronique est un domaine d'ingénierie interdisciplinaire, elle regroupe la mécanique, l'électronique et l'informatique pour concevoir des systèmes automatiques puissants.

La norme NF E 01-010 (2008) définit la mécatronique comme une « démarche visant l'intégration en synergie de la mécanique, l'électronique et l'informatique et l'automatique dans la conception et la fabrication d'un produit en vue d'augmenter et/ou d'optimiser sa fonctionnalité. [1]

- Etymologie

Le terme robot est d'origine **Slave**. Dans la langue russe **pa6oTa** (rabota) veut dire main d'œuvre ou travail.

Le sens actuel du mot a été introduit par le dramatiste tchécoslovaque Karel Capek (1890-1930) en 1920, dans une pièce appelée R.U.R. (Rossum's Universal Robots) où il a créé des substituts autonomes -avec un look humain et l'aptitude d'avoir des émotions humaines-

comme des serviteurs dociles et efficace pour les tâches pénibles mais qui déjà vont se rebeller contre leurs créateurs. [2]

- Définition

Un robot est une machine mécanique qui peut performer des tâches de manipulation ou de locomotion sous un contrôle automatique.

Le robot intelligent est celui qui a la capacité d'être programmé de prendre des décisions de performance en se basant sur l'input des capteurs.

3. Historique :

Considéré comme l'un des premiers robots mobiles autonomes, la tortue construite par Walter Grey dans le début des années 1950 (Fig.1.1). Ce robot n'utilise que des composantes analogiques dont des tubes à vide, mais il était capable de se diriger vers une lumière qui marque un but, de s'arrêter face à des obstacles et de recharger ses batteries lorsqu'il arrive dans sa niche. [3]

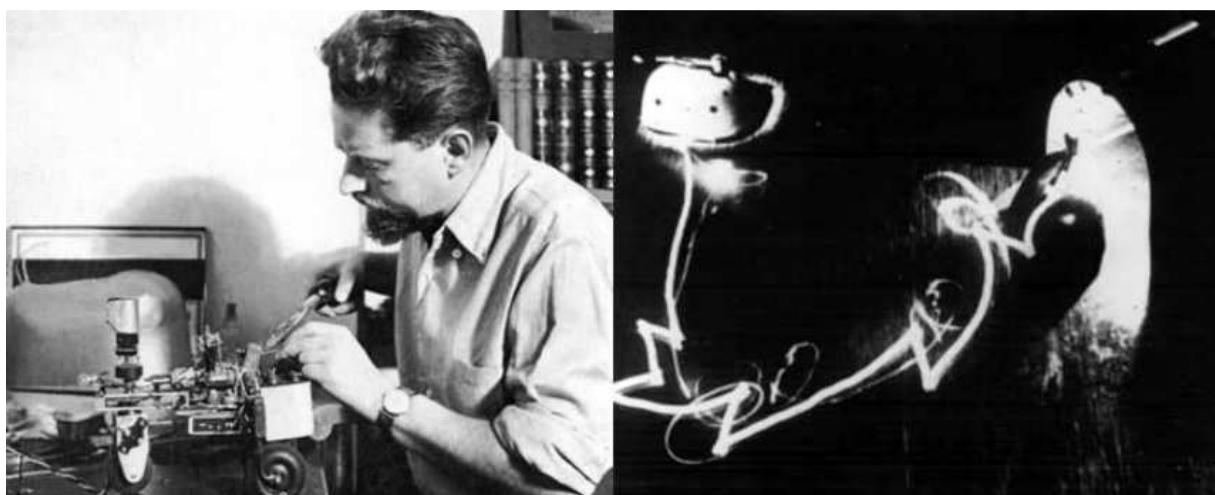


Fig.1.1 La tortue de Grey Walter (Machina Speculatrix ou Elsie) avec l'illustration de la trajectoire vers sa niche

Dans les années 1960 les étudiants de l'université John Hopkins ont créé le robot Beast (Fig.1.2). Ce dernier, en utilisant des capteurs ultrasons est capable de se déplacer au centre des couloirs, de localiser des prises électriques noires dans des murs blancs et de s'y recharger.

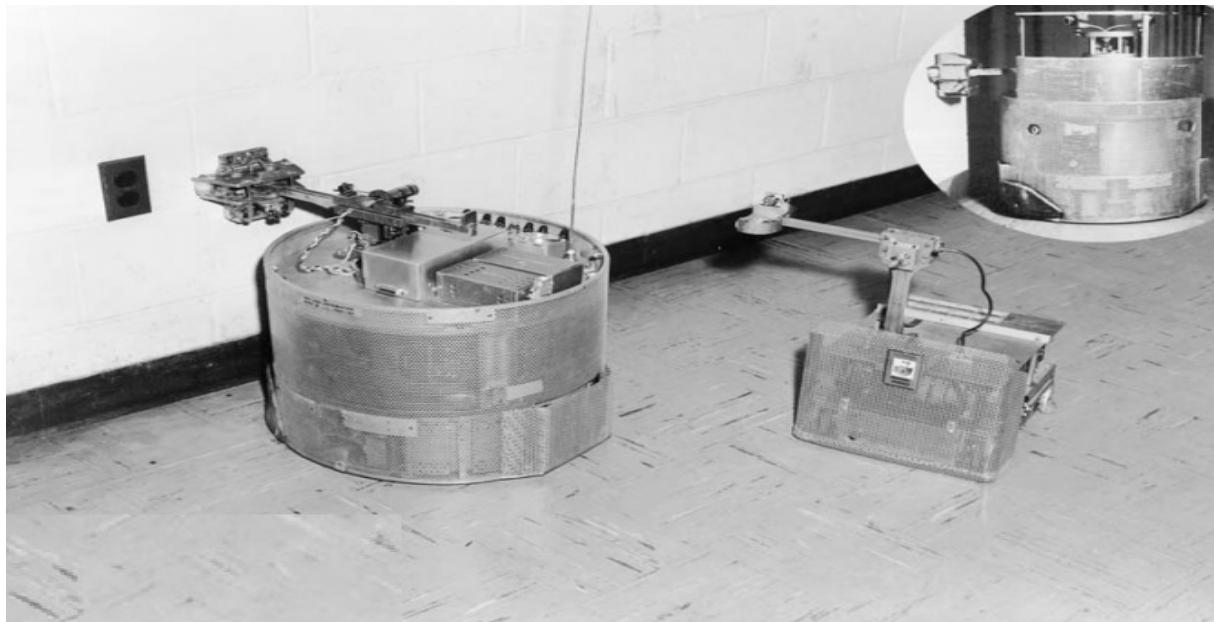


Fig.1.2 Robot « Beast » de l'université John Hopkins

Les premiers liens entre la recherche en intelligence artificielle et la robotique apparaissent à Stanford en 1969 avec Shakey (Fig.1.3). Ce robot utilise des télémètres à ultrason et une caméra et sert de plate-forme pour la recherche en intelligence artificielle.



Fig.1.3 Shakey de Stanford : Une plate-forme pour les recherches en intelligence artificielle [3]

Le Stanford Cart date de la fin des années 1970 (Fig.1.4), avec notamment les premières utilisations de la stéréo-vision pour la détection d'obstacles et la modélisation de l'environnement.

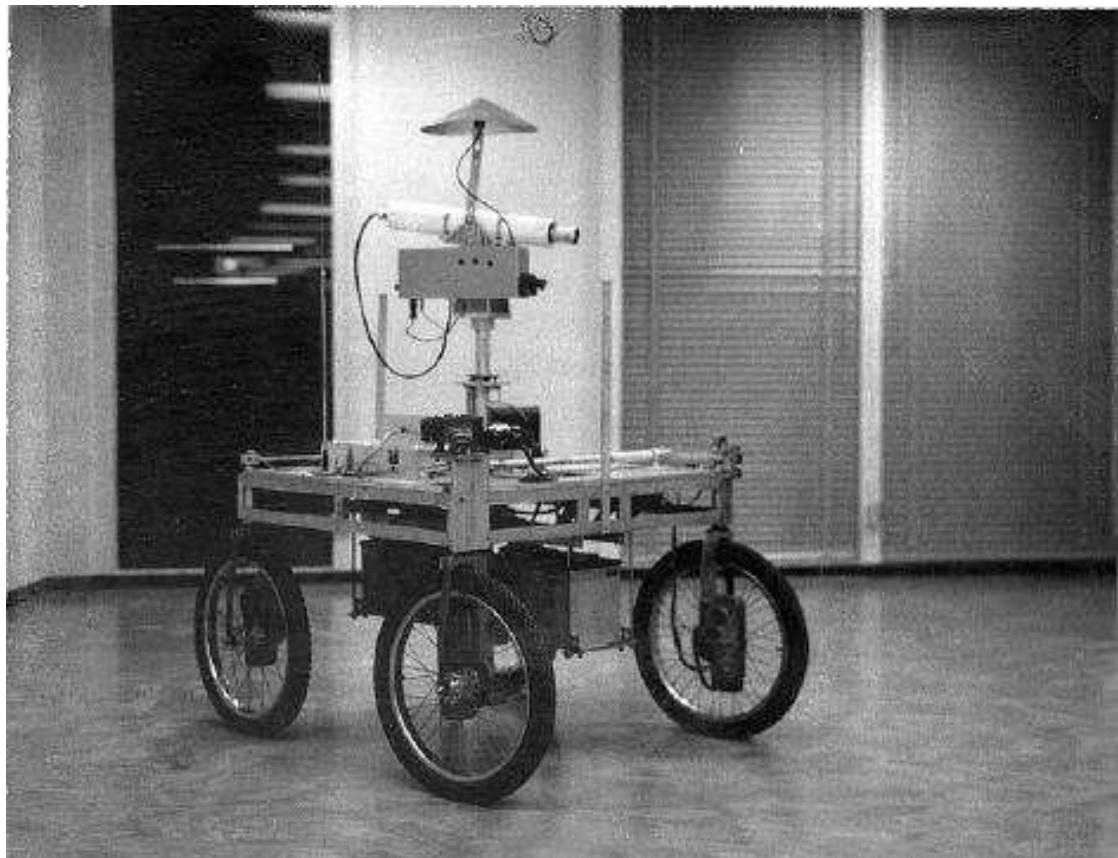


Fig.1.4 Le Stanford Cart de la fin des années 1970

En France, le robot Hilaire était le premier robot construit au LAAS à Toulouse (Fig.1.5).

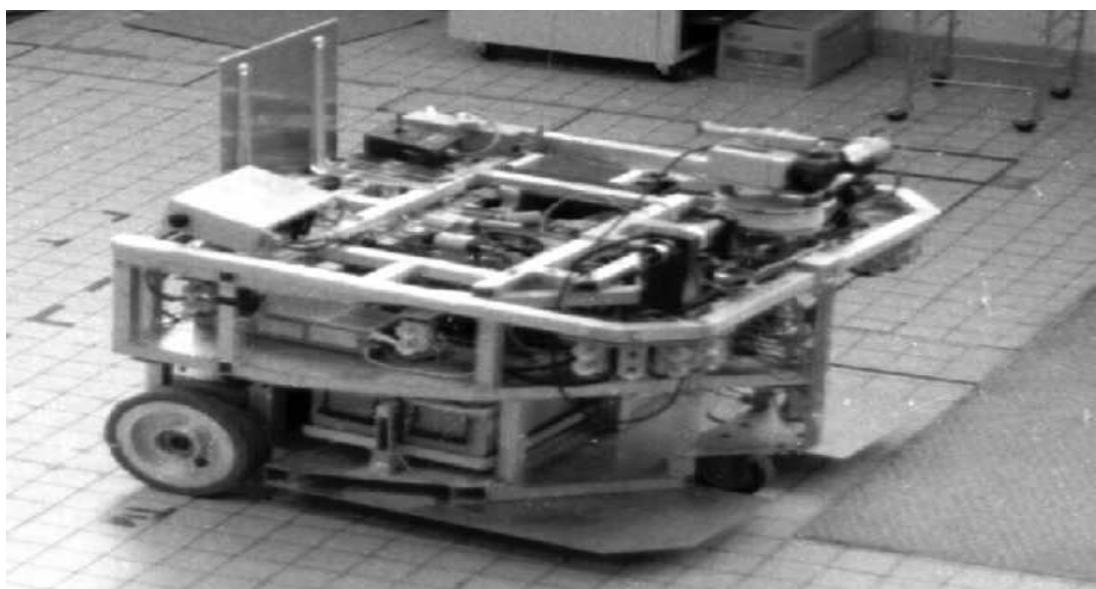


Fig.1.5 Robot Hilare du LAAS 1977

Une étape importante est à signaler au début des années 1990 avec la mise en avant de la robotique réactive, représentée par Genghis (Fig.1.6), développé par Rodney Brooks au MIT. Cette nouvelle approche de la robotique, qui met la perception au centre de la problématique, a permis de passer de gros robots très lents à de petits robots beaucoup plus réactifs et adaptés à leur environnement. [3]

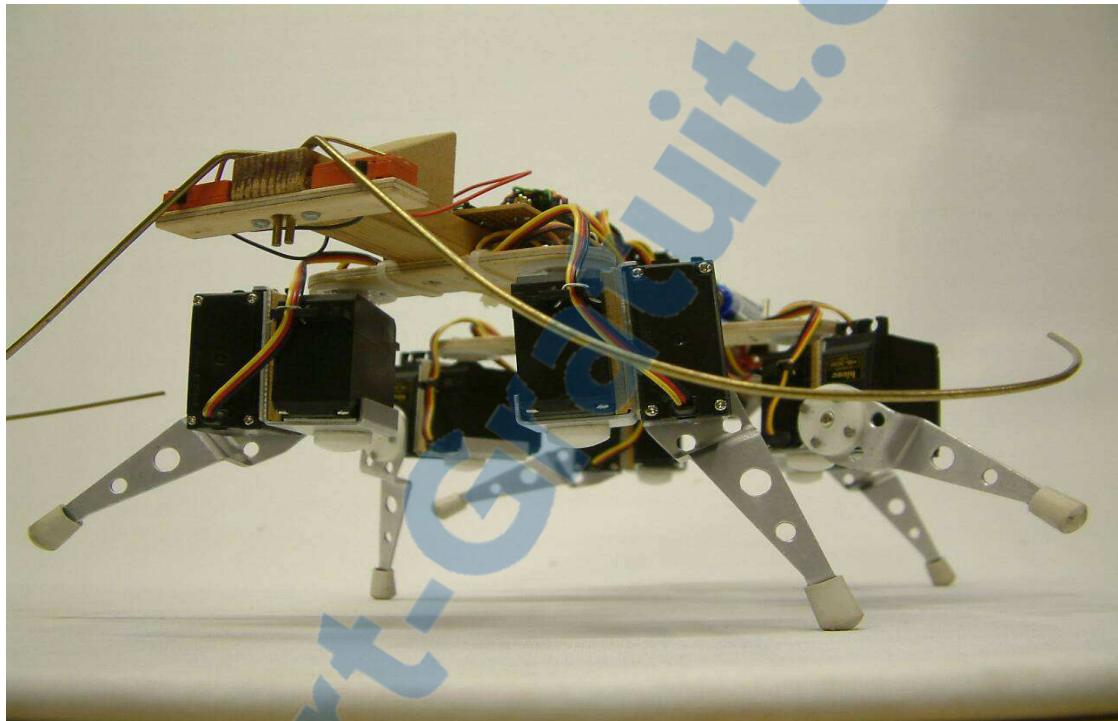


Fig.1.6 Genghis de Rodney Brooks 1990

4. Domaines d'application :

Le but premier d'un robot est d'effectuer des tâches répétitives et/ou précises. Les robots permettent également d'effectuer des tâches dans des environnements de travail trop dangereux pour l'Homme. On peut les regrouper par Dull-Dirty-Dangerous.

Les robots sont généralement présents dans différents domaines d'application tels que l'industriel, médical, militaire et spatiale.

4.1. Domaine industriel :

L'objectif de ces robots est de remplacer l'homme dans des activités fastidieuses ou onéreuses pour l'employeur. Les robots ont donc commencé à être utilisés dans les chaînes d'assemblage industrielles. Dans ces chaînes d'assemblage, on retrouve des robots soudeurs, manipulateurs, peintres ... (Fig.1.7)

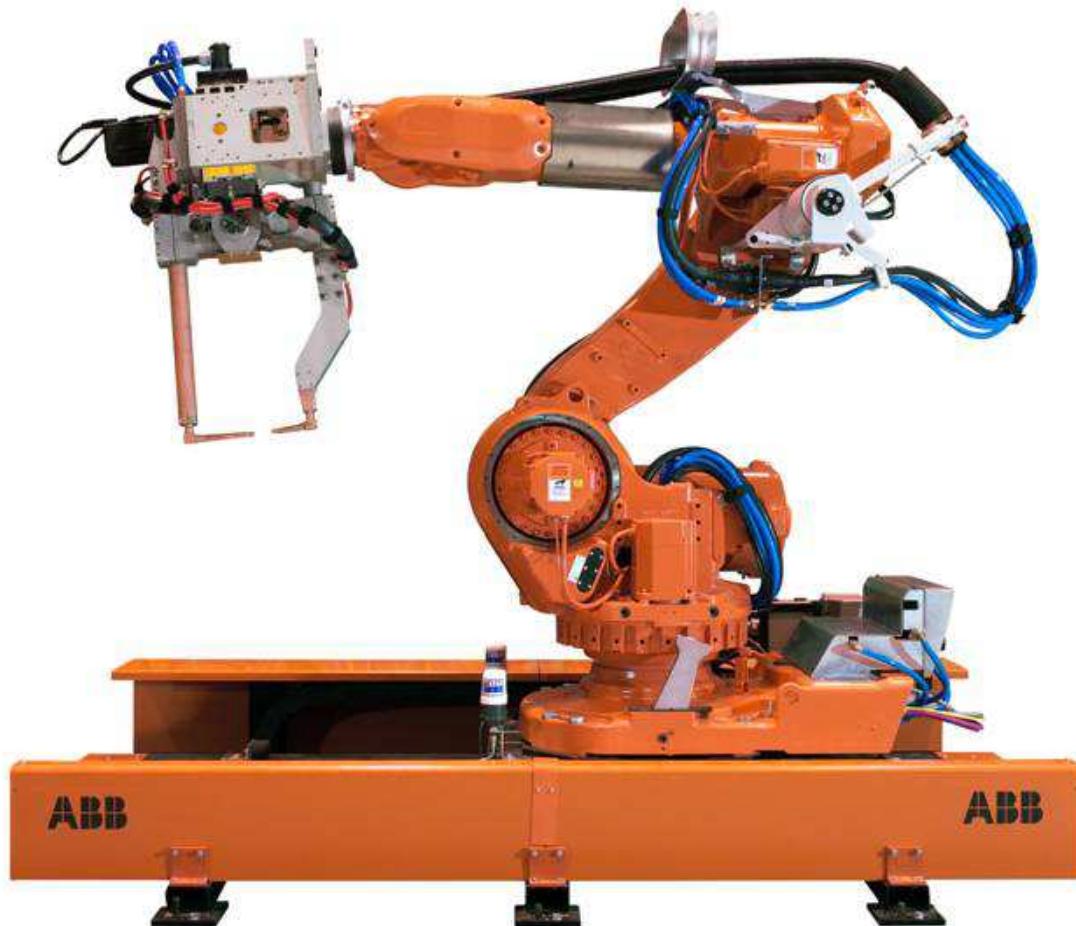


Fig.1.7 Bras soudeur d'ABB

4.2. Domaine militaire :

Les robots sont de plus en plus utilisés dans le domaine militaire. En effet, la miniaturisation permet aujourd’hui de créer des robots discrets mais dotés de nombreux capteurs, ce qui est idéal pour des missions d’espionnage ou d’éclairement.

- Robot éclaireur tout-terrain :

L’image captée par la caméra est transmise par une liaison sans fil aux fantassins ou au poste de commandement. (Fig.1.8)



Fig.1.8 Robot éclaireur militaire

- **Drone à courte portée pour le fantassin :**

Il est doté d'une caméra jour/nuit à transmission numérique, pour effectuer des missions de reconnaissance. Son poids de 4kg lui permet d'être transporté facilement, jusqu'aux abords des lieux à surveiller, par les militaires. (Fig.1.9)



Fig.1.9 Drone à courte portée

- **Drone tactique :**

D'une autonomie moyenne de 3 à 5h, il peut être équipé d'un radar, d'une caméra ou d'outils de mesures météorologiques. Sa portée de transmission de 100km lui permet un grand rayon

d'action. (Fig.1.10)



Fig.1.10 Drone tactique

- Le robot Mule, de l'armée américaine :

Initialement conçu pour transporter les paquetages des soldats américains, ce véhicule autonome est également doté de radars, sonars, d'une caméra infrarouge... Il est capable de mesurer la distance entre tout objet fixe ou mobile et de détecter tout mouvement ou source de chaleur. Il est entraîné par un moteur diesel, mais un moteur électrique prend le relais s'il doit se faire discret malgré ses 2 tonnes. Il devrait à terme être équipé d'un groupe électrogène et d'un détecteur de mines pour assister les soldats en temps réel. (Fig.1.11)



Fig.1.11 Robot mule de DARPA

4.3. Domaine médical :

Les robots commencent à être de plus en plus dans le domaine médical, qu'il s'agisse de « simples » échographies ou d'opérations chirurgicales plus délicates. En fait ces robots ne sont pas complètement autonomes mais ils assistent les médecins ou chirurgiens, jusqu'à permettre des opérations médicales à distance (télémedecine). On parle de surgétique c'est-à-dire tout ce qui consiste à introduire les derniers outils des technologies informatiques et robotiques dans la pratique médico-chirurgicale. Cette pratique de « chirurgie assistée » est émergente donc bien que peu répandue, elle est en phase de devenir la chirurgie du futur. (Fig.1.12) [4]

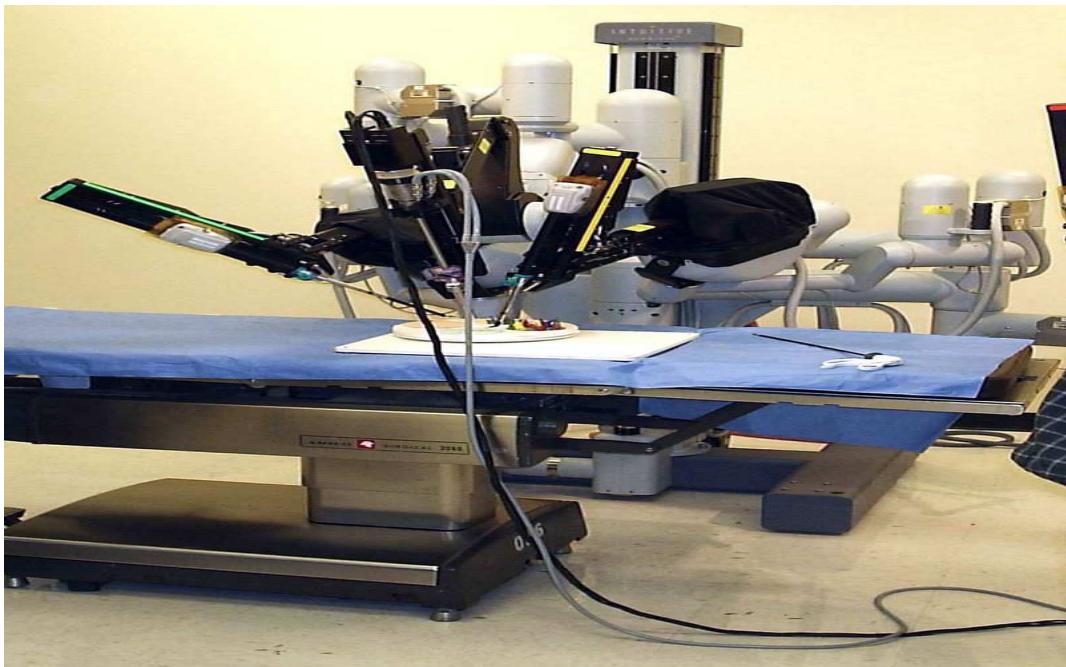


Fig.1.12 Robot Da Vinci

4.4. Domaine domestique :

Un robot domestique est un robot de service personnel utilisé pour des tâches ménagères. Les robots domestiques sont utilisés par exemple en vaisselle, en repassage, en nettoyage. Ils peuvent également être utilisés dans le domaine de la restauration. (Fig.1.13)



Fig.1.13 Robot aspirateur

4.5. Domaine spatiale :

Les scientifiques ont pris le parti d'envoyer des robots pour explorer le système solaire, dans des environnements souvent mortels pour l'homme. (Fig.1.14)

L'autonomie d'un robot d'exploration spatiale est obligatoire, et doit être d'autant plus grande qu'il est éloigné de la Terre, du fait du temps qui s'écoule entre l'envoi d'une commande depuis la Terre, et la réception de cette commande par le Robot. Celui-ci doit donc être capable de réagir tout seul aux évènements qui peuvent surgir dans cet intervalle de temps.



Fig.1.14 Rover de l'exploration du Mars

5. Type des robots :

Ils existent plusieurs types qui ont lien avec la fonctionnalité du robot.

- Manipulateur :

La majorité de ces robots sont à base fixe. Quand la base n'est pas fixe, elle est généralement montée à un rail. Il est un mécanisme qui consiste à une série de segments articulés et en glissant relativement l'un à l'autre pour la manutention des objets dans des différents degrés de liberté. Commandé par un ordi ou un opérateur. On retrouve dans cette catégorie les robots de manipulation, type « Pick And Place », des robots soudeurs ou encore des robots de peintures. [2]

- **Humanoïde :**

Catégorie la plus connue, en grande partie grâce à leur promotion faite par la science fiction, elle regroupe tous les robots anthropomorphes, ceux dont la forme rappelle la morphologie humaine. Ces robots ont généralement un torse, une tête, deux bras et deux jambes. On peut citer le robot Asimo de Honda. Parfois, certains de ces robots ne représentent qu'une partie du corps, comme le robot Nexi développé par le MIT. Lorsqu'un robot anthropomorphe imite non seulement l'apparence physique, mais aussi les comportements humains, on l'appelle un androïde. Un parfait exemple d'androïde est l'Actroid-DER de la société Kokoro. [5]

- **Mobile :**

Un robot mobile a la capacité de se déplacer dans un environnement. Cette catégorie englobe tous les robots à base mobile. Les robots mobiles peuvent être autonomes (AMR) car ils sont capables de naviguer dans un environnement non contrôlé sans avoir besoin des outils de guidance électromécanique. Ils peuvent aussi compter sur ses outils de guidance qu'ils les permettent de se déplacer dans un routage prédéfini (AGV). UGV (Unmanned ground vehicles) ça regroupe les robots actionnés par roues ou chenilles généralement exploités pour faire l'exploration (appelés rover : vagabond en En) et à pattes : D'abord, il y a la question du nombre de pattes et comment elles assurent la stabilité quand le robot est en mouvement. Ou plus critique, la stabilité quand il est immobile. Puis il y a la question de savoir comment les jambes propulsent le robot en avant ou en arrière. Les robots à pattes ou jambes sont un défi à concevoir et à construire, mais ils fournissent un niveau supplémentaire de mobilité par rapport aux robots sur roues. Les robots sur roues peuvent avoir un passage difficile lorsqu'ils se déplacent sur un terrain accidenté, mais un robot sur pattes bien conçu peut facilement marcher sur les petits fossés et passer des obstacles (bipèdes : difficultés à assurer l'équilibre et le contrôle / quadrupèdes : 4 pattes, plus faciles à équilibrer / hexapodes : 6 pattes, sont capables de marcher à des vitesses rapides sans tomber, et sont plus que capables de faire des virages et de se déplacer sur un terrain accidenté.)

- **Drones :**

Un drone ou unmanned aerial vehicle (UAV) est un aéronef sans passager ni pilote qui peut voler de façon autonome ou être contrôlé depuis le sol.

La taille d'un drone aérien peut aller de quelques centimètres pour les modèles miniatures à plusieurs mètres pour les drones spécialisés (surveillance, renseignement, combat, transport et loisirs). L'autonomie en vol va de quelques minutes à plus de 40 heures pour ceux de longue endurance.

Les drones militaires, ce concept a émergé durant la première guerre mondiale. A l'origine, le drone était un avion-cible à vocation militaire. Son développement a suivi le rythme des grands conflits du XX^e siècle : Seconde guerre mondiale, guerre de Corée et les conflits du Moyen-Orient. Les drones sont plus économiques tout en évitant de mettre en jeu la vie des pilotes et de déployer des troupes terrestres notamment pour les missions de reconnaissances, de surveillance et des attaques ciblées. Leur utilisation aux seins des armées est devenue prépondérante. [6]

Dans le civile, de nombreux domaines (Cinéma, télévision, agriculture, environnement, etc.). On a vu les drones susciter des applications inédites grâce à leur capacité à embarquer des appareils photos, des caméras, des caméras infrarouges ou des capteurs environnementaux. Plusieurs sociétés spécialisées dans le transport (DHL, UPS, Allship, La poste) ainsi que le géant de e-commerce Amazon travaillent sur des concepts de drone-livreurs. Ce type de service a été introduit en 2015 aux Émirats Arabes Unis pour la livraison des documents officiels.

6. Architecture d'un robot mobile :

- La structure mécanique et la motricité :

Roues, Chenilles et Pattes – motricité : les déplacements des robots sont réalisés par des moteurs électriques, thermiques ou hydrauliques. L'énergie électrique la plus fréquemment employée offre l'avantage d'une commande aisée. Par contre le transport et la génération présentent des difficultés. Plusieurs méthodes sont employées :

Par batteries qui sont soit rechargées périodiquement de manière automatique ou manuelle, soit par un échange avec d'autres lorsqu'elles sont déchargées.

Par groupe électrogène embarqué dont l'inconvénient constitue la masse élevée. L'énergie de base est alors thermique.

- Les organes de sécurité :

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

Le traitement de la détection s'effectue selon plusieurs cas. Si le capteur à contact est sollicité, le robot s'immobilise soit définitivement soit tant que le contact persiste, ou il effectue un mouvement opposé au contact. Par contre si un proximètre détecte une présence, la stratégie consiste soit à immobiliser le robot en attendant que la personne s'éloigne, soit à ralentir le mouvement si la personne n'est pas trop proche, soit à choisir un autre chemin qui l'éloigne de la personne.

- Localisation :

La localisation consiste à définir la position en termes de coordonnées d'un point du mobile par rapport à un référentiel de base. Les techniques employées sont de deux types :

La localisation **relative**, basée sur l'utilisation de capteurs proprioceptifs. Odométrie utilisée pour les robots à roues. Système inertiel : Utilisation d'accélération et de vitesse.

La localisation **absolue**, basée sur l'utilisation de capteurs extéroceptifs. Localisation absolue. Elles assurent la mesure de la position et de l'orientation du mobile à tout instant.

7. Conclusion :

La robotique est une discipline qui a subi un développement en parallèle à celui de l'informatique, de l'électronique et de la mécanique. Ces trois domaines sont indispensables dans un système robotique, ce qui était la raison de la situer dans le spectre des systèmes mécatroniques.

Dans ce chapitre on avait défini la robotique, on avait cité les points marquants dans son histoire et ces différents domaines d'application.

A la fin on a concentré sur l'architecture d'un robot mobile en citant d'une façon générique sa structure électromécanique, les équipements de sécurité, et ses différentes stratégies de navigation et localisation.

Dans le chapitre prochain, on va parler sur le robot différentiel et sa conception mécanique et électronique.

Chapitre 02

Rapport.Gatuit.Com

1. Introduction :

Dans ce chapitre on va définir la configuration d'un robot à roues différentielles comme le robot élaboré lors de notre projet en justifiant ce choix parmi les robots mobiles relativement aux éléments de l'environnement.

Ce chapitre présente dans un premier temps quelques généralités concernant les robots mobiles à roues et ainsi tous les points qui doivent être cités pour une meilleure perception de cette catégorie. On commence par l'étude théorique du modèle géométrique de robots différentiels, leur architecture cinématique, dynamique et tous les concepts nécessaires qui permettent d'établir un modèle contrôlable.

Nous présentons ensuite la conception mécanique et la citation des composantes mécaniques. Cette partie comprenne les propriétés voulues du châssis Et le choix de roues motrices et des moteurs.

L'étape prochaine consiste à définir la conception électronique du robot. Ça commence par donner la structure globale du système électronique. La conception électronique se termine par une présentation de la partie d'alimentation et le choix des batteries.

Pour comprendre l'architecture de contrôle de robot, on va donner des rappels théoriques sur les techniques de navigation utilisés généralement pour le pilotage des robots mobiles.

Puis, la détection des obstacles en citant les différentes parties du système sensoriel d'un robot mobile, cette partie sera faite par l'illustration des capteurs et leurs principes de fonctionnement. On va aussi parler sur la localisation du robot mobile.

Finalement, Ce chapitre se termine par une présentation des méthodes de navigation ce qui va nous mener à atteindre l'objectif du déplacement d'un robot mobile.

2. Choix de robot :

Pour la conception d'un robot mobile à roues, il existe une diversité de configurations faisables. On peut mentionner parmi les configurations candidates :

- Robot tricycle :

Dans ce modèle on trouve, un moteur de propulsion couplé à 2 roues motrices et un moteur de direction couplé à une roue directionnelle. Son centre de masse est le plus près possible de l'axe des roues motrices.

- Robot à vitesse différentielles :

Il existe deux configurations pour le modèle différentiel, **asymétrique** avec 2 moteurs de propulsion et de direction couplés à deux roues motrices. Ces dernières sont disposées en avant ou en arrière et une roue libre assure la stabilité est disposée à l'autre extrémité. Dans le modèle **symétrique**, les roues motrices sont disposées au centre du robot et 2 roues libres assurant la stabilité sont disposées en arrière et en avant. Le centre de masse du premier se trouve le plus près possible de l'axe des roues motrices et pour le deuxième, le plus près du centre du robot.

- Robot à 4 roues motrices et directionnelles :

Avec 4 roues motrices jumelées 2 à 2 (côté gauche ensemble et droite ensemble), 2 moteurs de propulsions (vitesse différentielles), 2 moteurs de direction un pour les roues avant et un pour la roue arrière. Le centre de masse se trouve le plus près possible du centre du robot.

La configuration qui a été choisi est le modèle différentiel asymétrique pour les raisons suivantes :

- Simplicité de contrôle : Dans ce genre de robot, la cinématique et la dynamique qui facilite le développement de contrôleurs haut niveau.
- Bonne fiabilité mécanique : Le nombre de membrures et de composantes mécaniques étant réduit.
- Faible cout de fabrication : La quantité de matériau et de composantes est minimale.

3. Robot différentiel :

Un robot différentiel, est un robot ayant deux roues autopropulsées indépendantes et une ou deux sphères directionnelles. (Fig.2.1)

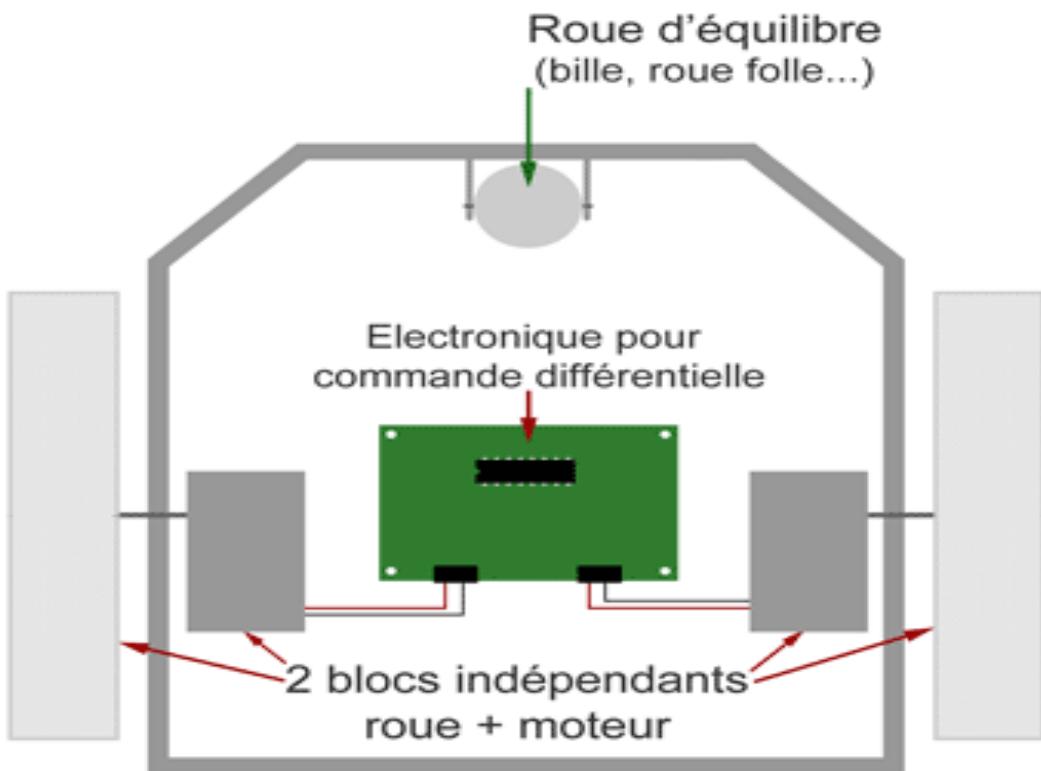


Fig.2.1 Robot mobile différentiel

4. Modèle géométrique :

Le modèle géométrique est une transformation mathématique dont les entrées sont les vitesses angulaires des roues (généralement mesurées avec des codeurs) et la sortie est la pose (position et orientation) du robot mobile dans son espace de travail. C'est tout simplement l'étude des mathématiques de mouvement sans la considération des forces qui causent ou affectent ce mouvement.

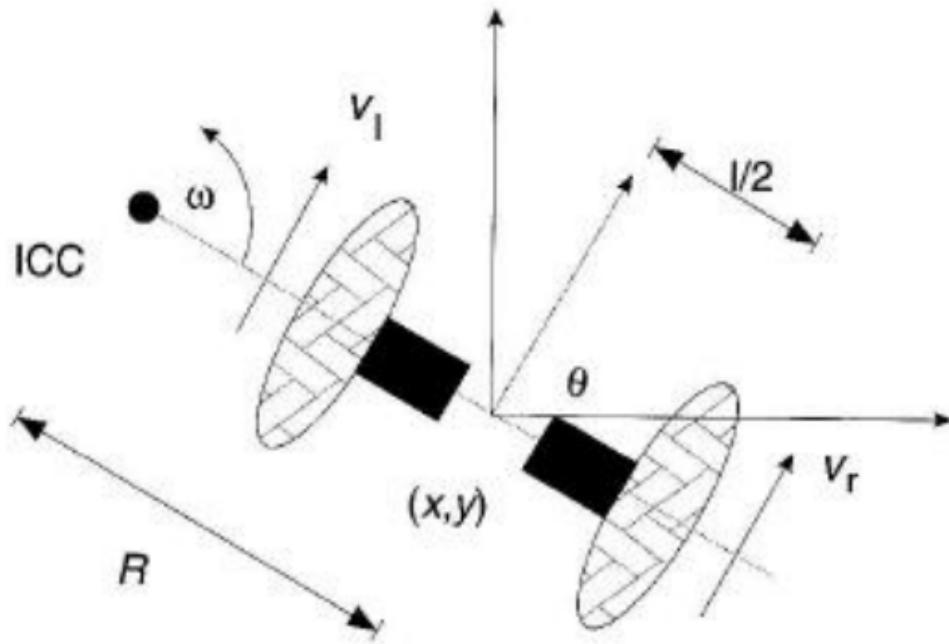


Fig.2.2 Cinématique d'un robot à vitesses différentielles

Notre robot à deux (02) roues motrices disposées sur un axe commun, chaque roue a un moteur de propulsion indépendant qui peut se tourner dans et contrairement au sens des aiguilles pour produire une translation en avant et en arrière. On peut aussi varier la vitesse de moteur associé à chaque roue pour performer un mouvement roulant ou circulaire. Le point auquel le robot tourne sur est connue par ICC (Instantaneous Center of Curvature). Les équations qui suivent donnent la relation entre les vitesses des roues droite et gauche V_r et V_l et la vitesse angulaire de rotation ω sur ICC :

$$\omega(R + l/2) = V_r$$

$$\omega(R + l/2) = V_l$$

Où, l est la distance entre les centres des deux roues, R est la distance qui sépare l'ICC du point de centre entre les roues. A n'importe quel instant t on peut résoudre pour R et ω , ce qui va donner :

$$R = l/2(V_r + V_l)/(V_r - V_l)$$

$$\omega = (V_r - V_l)/l$$

On peut discuter trois cas intéressants:

Dans le cas où $V_r = V_l$ le mouvement linéaire est une droite, R tend vers l'infini et il n'y a pas de rotation donc $\omega = 0$.

Dans le cas où $V_r = -V_l$, $R = 0$ et on a une rotation sur le point centre de l'axe des roues.

Si $V_r = 0$ ou $V_l = 0$ donc on peut observer une rotation autour la roue droite ou gauche respectivement.

- **Equations des vitesses angulaires :**

On pose le diamètre des roues $D = 2r$ et les vitesses angulaires φ_r et φ_l . Donc on a la vitesse de translation du robot par :

$$V = \frac{(V_r + V_l)}{2} = r \frac{(\varphi_r + \varphi_l)}{2}$$

Et la vitesse angulaire ω est donnée par :

$$\omega = r \frac{(\varphi_r - \varphi_l)}{l}$$

On pose X et Y les coordonnées cartésiennes du robot et ϑ son orientation (position angulaire).

- **Calcul des déplacements élémentaires :**

La vitesse linéaire de chaque roue est donnée par :

$$V_r = r \cdot \varphi_r$$

$$V_l = r \cdot \varphi_l$$

La vitesse moyenne du robot est :

$$V = \frac{(V_r + V_l)}{2}$$

La vitesse du robot peut être projetée le long des axes x et y :

$$\Delta X = V \cdot \cos(\theta)$$

$$\Delta Y = V \cdot \sin(\theta)$$

Le déplacement élémentaire du robot est donné par la relation suivante :

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta \theta \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ 2 & -2 \\ \frac{l}{l} & \frac{l}{l} \end{bmatrix} \cdot \begin{bmatrix} \varphi_r \\ \varphi_l \end{bmatrix}$$

- La position absolue :

La position absolue peut être calculée grâce aux équations suivantes :

$$X_i = X_{i-1} + \Delta X$$

$$Y_i = Y_{i-1} + \Delta Y$$

$$\theta_i = \theta_{i-1} + \Delta \theta$$

Où, X_i et Y_i sont les coordonnées cartésiennes du robot à l'instant i et θ_i est l'orientation du robot à l'instant i.

5. Conception mécanique :

5.1. Cahier de charge :

À chaque nouvelle conception, il y a des objectifs de performance qui doivent être fixés pour le robot, afin de décider, quelle configuration on va adopter pour la conception mécanique, parmi les différents critères, on va citer celles qui ont une influence majeure pour un robot à roues différentielles.

- La première concerne les problèmes mécaniques : le robot doit être très léger, facile à prendre et à manipuler, contrôlable et rigide, d'ailleurs, il doit avoir plus d'autonomie de batterie.

- Simplicité de contrôle : la cinématique et la dynamique du robot sont très simples, facilitant le développement de contrôleurs haut niveau.
- Positionnement précis : le positionnement du robot à l'aide de placement de ses senseurs doit être relativement précis.
- Symétrie: ceci constitue le point déterminant de cette structure. La symétrie permet des performances intéressantes en rotation puisque le centre de masse peut être situé au centre du robot, sur l'axe des roues motrices.

La plateforme robotique que nous souhaitons construire est composée de 2 moteurs de propulsion et de direction couplés à deux roues motrices. Les roues motrices sont disposées à l'avant ou à l'arrière et une roue-libre assurant la stabilité est disposée à l'autre extrémité du robot. Le centre de masse se retrouve le plus près possible de l'axe des roues motrices.

La base motorisée de robot peut être le plus difficile sous-système pour concevoir et construire, on peut gagner beaucoup de temps si on concevoir à partir de cette partie du robot. Au lieu de la conception du premier pas et le montage du base motorisé au corps du robot. La base tient généralement tous les composants qui faire fonctionner le robot. Par exemple, la base comporte des roues motrices, ainsi que les batteries (qui peut être assez lourd) et le système de contrôle.



Fig.2.3 La motorisé du robot mobile

- **Choix de matériau de châssis :**

Lors de cette phase en se concentre sur la contrainte du poids, une structure mécanique doit être relativement légère possible. Elle doit également offrir une rigidité suffisante pour supporter le poids important du système et ne pas se déformer lors des mouvements du robot.

On peut résumer les points qu'un châssis doit satisfaire par : [7]

- Légereté.
- Ductilité.
- Résistance de corrosion.

Pour obtenir la meilleure structure et dimensions du châssis, on doit d'abord connaître toutes les éléments qui doivent être retrouvé sur la base mécanique du robot. (Fig.2.4)



Fig.2.4 Exemple d'un châssis d'une plateforme différentielle [8]

En effet lors de la détection d'un obstacle par les capteurs de proximité, le châssis circulaire ou ovale garantit un dégagement par inversion du sens de rotation des roues (en tournant sur place).

Nous devons trouver un moyen d'agencer les roues motrices, les moteurs et la roue folle en utilisant une structure légère et suffisamment rigide. La base mécanique se doit également d'incorporer les batteries du robot puisqu'elles représentent un poids important et il est donc nécessaire de les soutenir par une structure suffisamment rigide.

- **Choix des roues :**

La performance du robot dépend grandement sur leur dimension par exemple, le couple fourni et la vitesse maximale sont deux paramètres qui varient en fonction du rayon de la roue.

Généralement dans cette phase on est censé de faire des compromis, pour avoir une accélération maximale on choisit des roues avec un rayon assez faible car l'inertie de rotation vue par le moteur est directement proportionnelle à ce rayon, par contre la vitesse maximale est inversement proportionnelle au rayon, donc il ne faut pas trop le diminuer.

L'adhérence latérale dépend de la forme des roues ainsi du matériau de la surface de contact avec le sol (préférable d'utiliser crampons de caoutchouc).

On peut aussi citer que la large surface de contact au sol peut amplifier l'erreur sur la mesure de l'orientation du robot. Pour un positionnement précis il est désiré d'utiliser des roues qui offrent une surface de contact minimale.

- **Modélisation des roues :**

Il faut donc: connaître avec précision le diamètre des roues et la distance séparant ces deux roues. Les capteurs odométriques mesurent le déplacement des roues en nombre de tours, pour savoir quand arrêter le mouvement.

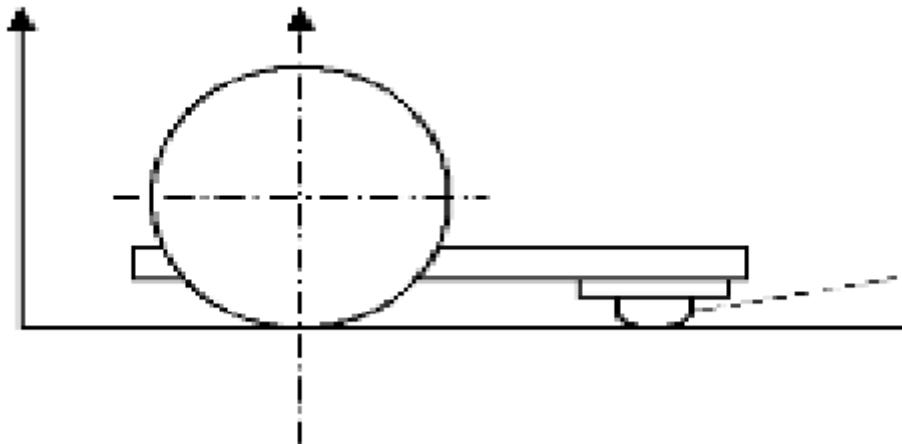


Fig.2.5 Vue de profile des roues motrice. [9]

La figure (Fig.2.5) présente le positionnement des roues. Il est important de déterminer le diamètre des roues car on ne mesure pas directement la distance parcourue par la roue mais le nombre de tours qu'elle fait.

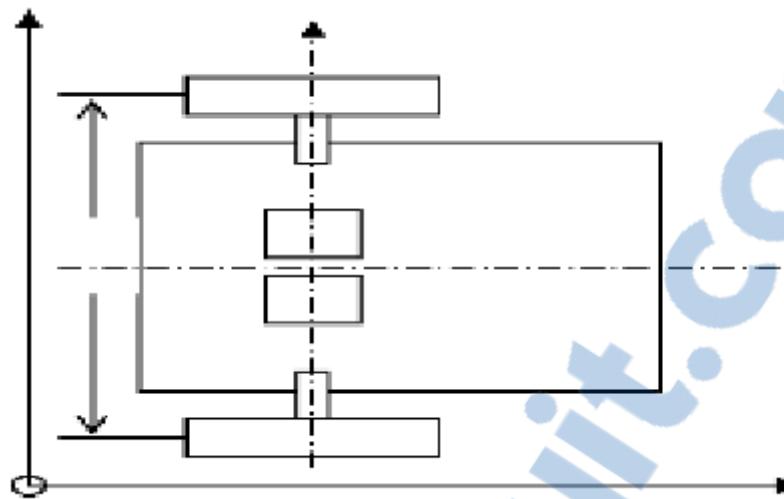


Fig.2.6 Vue dessous des roues. [9]

La figure (Fig.2.6) présente l'écartement des roues motrices.

- Roue libre :

La roue libre assure la stabilité du robot pour les différents mouvements possibles. Lorsque le robot passe d'une vitesse tangentielle positive à une vitesse tangentielle négative, la roue libre engendre des perturbations sur le mouvement du robot. Plus le poids et la friction sont importants, plus ces perturbations n'augmentent.

Donc le choix idéal sera une roue libre plus légère avec un roulement à friction très faible. Elle doit être disposée de façon à ce que les points de rotation de cette roue et les points de contact des roues motrices forment un triangle.

- Choix de moteurs :

On doit déterminer avec précision le travail qui sera accompli par chacun des moteurs, connaître les spécifications que nous voulons atteindre et trouver le bobinage et l'engrenage pour certaines gammes de moteurs qui permettent de répondre aux besoins.

6. Conception électrique et électronique :

Dans cette partie, on va expliquer l'électronique embarquée de notre robot mobile, tout en assurant la symbiose avec les aspects mécaniques, esthétiques et software de la machine.

L'enjeu ici, c'est de poser des choix au niveau de la partie électronique qui soient compatibles avec les autres parties et d'estimer les implications, la faisabilité et le coût.

La conception électrique et électronique doit débuter avec le choix de l'architecture. Le choix de l'architecture du système électrique et électronique constitue en fait l'identification de toutes les composantes qui formeront le système. Nous pouvons diviser cette architecture en deux sections, soit l'alimentation et l'électronique de contrôle.

6.1. Structure globale:

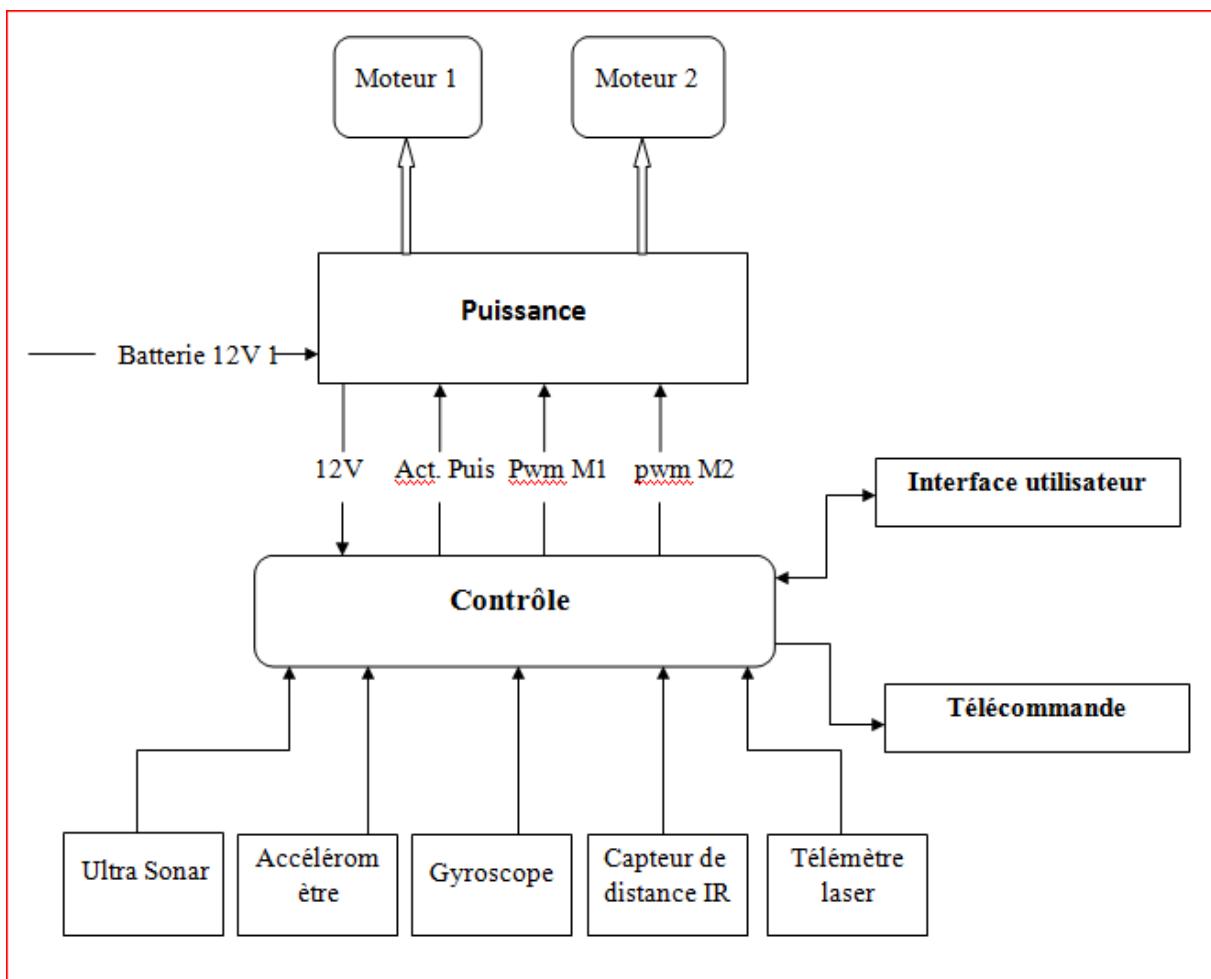


Fig.2.7 La structure globale de la partie électrique et électronique du robot

6.2. Alimentation :

Il faut dans un premier temps, choisir le circuit d'alimentation du robot. Ce circuit doit répondre aux besoins de sécurité et de flexibilité.

- Convertisseurs DC-DC :

L'alimentation de l'électronique est assurée par les convertisseurs de tension continue. Un DC-DC est un module générant une tension de sortie régulée, isolée de la tension d'entrée et plus faible.

Ces DC-DC permettent d'atténuer le bruit électrique et les perturbations, mais surtout de passer d'une tension d'entrée à une autre pour alimenter les différents modules électroniques selon besoins énergétiques.

- Choix de batteries :

Le choix des batteries est crucial lors de la conception d'un robot puisqu'il implique d'importantes contraintes aux niveaux de la conception mécanique et de l'autonomie électrique.

Le volume occupé par les batteries et leurs poids influent sur la conception mécanique, donc il est inévitable de faire des compromis entre l'autonomie électrique et les attributs physiques des batteries pour mener le robot à avoir des performances en accélération satisfaisantes et une autonomie également satisfaisante.

Un robot mobile est généralement décrit par son utilisation cyclique et une décharge relativement rapide. Pour cela, il existe trois (03) familles de batteries rechargeables qui sont appropriées à un robot mobile : les batteries à acide-plomb (SLA), les batteries au Nickel-Cadmium (Ni-Cad) et les batteries aux Nickel-Hybride de métal (Ni-MH). Les deux dernières familles offrent une densité d'énergie plus importante que la première mais ces batteries sont plus dispendieuses que les batteries de SLA et elles demandent une attention plus soignée au niveau de la recharge.

La solution acide-plomb est très peu dispendieuse et sa recharge est assez facile.

7. Architecture de contrôle :

Un robot est un système complexe qui doit satisfaire des exigences variées en utilisant un ensemble logiciel appelé architecture de contrôle du robot. Cette architecture permet donc d'organiser les relations entre les trois grandes fonctions: la perception, la décision (planification) et l'action.

Une architecture doit donc être conçue pour un robot précis, mais en utilisant des modules génériques. De manière plus générale il existe également des règles de conception relativement générales qui permettent de réaliser ces implantations. En fonction de ces règles, les architectures de contrôle peuvent être classées en trois grandes catégories que nous

détaillerons par la suite : les contrôleurs **hiérarchiques**, les contrôleurs **réactifs** et les contrôleurs **hybrides**, toutes ces architectures ne diffèrent pas forcément par les méthodes élémentaires employées mais plutôt par leur agencement et leurs relations. (Fig.2.8)

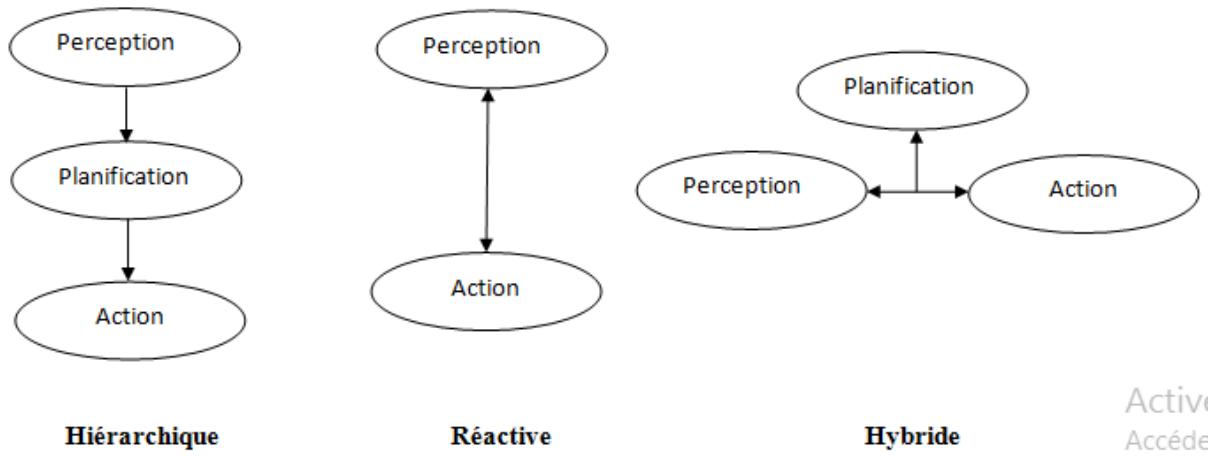


Fig.2.8 Différents types de contrôleurs

7.1. Contrôle du système :

Pour un système quelconque, un contrôleur est un mécanisme qui permet d'ajuster et améliorer l'attitude ou l'état de ce système, pour le considérer comme un contrôleur de système et non seulement un changeur d'état du système, l'output ou le comportement de ce dernier, doit tendre vers un état désiré.

Il est donc intéressant, une fois un modèle de simulation développé, d'utiliser cet outil pour valider les concepts et ajuster les contrôleurs, pour ensuite étudier l'effet de certains paramètres sur le système. Mais il est également bien utile de pouvoir simuler le système en utilisant le programme de contrôle utilisé sur le vrai robot. Il est alors possible de valider l'implantation des algorithmes en simulation et de déterminer sans avoir des risques ou générer des incidents graves sur le système réel.

8. Détection des obstacles :

La commande des robots mobiles est basée sur deux types d'informations importantes; les informations proprioceptives et les informations extéroceptives.

Le système de perception est très important pour la sécurité du robot si l'environnement est encombré d'obstacles fixes ou bien mobiles.

- **Capteurs intéroceptifs (Localisation à l'estime):** fournissent des données sur l'état interne du robot (vitesse, position, orientation,...). Ces informations renseignent le robot en cas de mouvement, sur son déplacement dans l'espace

(la localisation). Ce sont des capteurs que l'on peut utiliser directement, mais ils souffrent d'une dérive au cours du temps, par exemple, les roues peuvent glisser, ce qui vient fausser les lectures, ce qui rend leur utilisation seul inefficace ou avec limitation. Parmi c'est techniques, il y a: l'odomètre, radar, systèmes inertIELS (l'accéléromètre, gyroscope).

- **Capteurs extéroceptifs (Localisation absolue):** ont pour objectif d'acquérir des informations sur l'environnement proche du système. Ils fournissent des mesures caractéristiques de la position que le robot peut acquérir dans son environnement par la détection des objets qui contourne. Ces informations peuvent être de natures très variée. Différentes méthodes peuvent être appliquées : les télémètres à ultrason, infrarouge, laser, les caméras,...etc.
Ce dernier type des capteurs peuvent être classés en deux catégories, passives et actives.
Dans le premier cas, on se contente de recueillir et d'analyser une énergie fournie par l'environnement, typiquement la lumière. Dans le second cas, c'est au capteur de générer une énergie, et de récupérer cette énergie après interaction sur le milieu extérieur.

- Système sensoriel :

Un système sensoriel est l'ensemble des organes qui permet au notre robot de percevoir son environnement. Il est constitué de différent type de capteurs. Pour un robot mobile, généralement un système sensoriel a comme objectif la mesure de proximité et de distance.

Généralement la mesure de distance est par des procédés acoustiques, des dispositifs optiques ou radio électriques. L'appareil permettant de mesurer les distances est appelé télémètre. De même elle joue le rôle d'une liaison entre le dispositif de mesure (le robot) et l'objet cible. Il existe différentes technologies pour réaliser des télémètres.

- **Capteurs infrarouges :** Ils sont les plus réponduS dans le secteur de la robotique. Ils sont hautement sensibles, sélectifs et stable. Leurs durée de vie est relativement longue et ils sont insensibles aux modifications de l'environnement.
Ils sont constitués d'un ensemble émetteur/récepteur fonctionnant avec des radiations non visibles, dont la longueur d'onde est juste inférieure à celle du rouge visible.
(Fig.2.9)

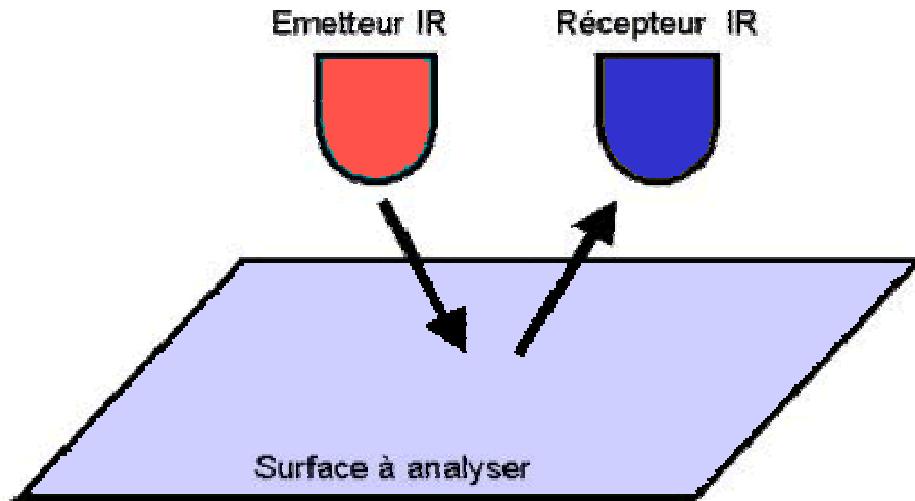


Fig.2.9 Principe de fonctionnement d'un capteur infrarouge

- **Capteurs ultrasonores :** Largement utilisés en robotique. La raison en est qu'ils peuvent fournir une information très importante au robot, à savoir la distance le séparant des obstacles autour de lui, qu'ils sont simple et donc peu chères. (Fig.2.10) Les capteurs ultrasonores utilisent des vibrations sonores dont les fréquences ne sont pas perceptibles par l'oreille humaine. Les fréquences couramment utilisées dans ce type de technologie vont de 20 kHz à 200 kHz.
Les ultrasons émis se propagent dans l'air et sont réfléchis partiellement lorsqu'ils heurtent un corps solide.



Fig.2.10 Capteur ultrason

Les télémètres ultrasons possèdent une zone dite « aveugle » de quelques centimètres

de longueurs, en dessous de laquelle ils ne peuvent détecter les obstacles. Cela est due au fait que le récepteur du capteur d'ultrasons est désactivé pendant un court instant après l'émission de l'onde ultrasonore. (Fig.2.11)

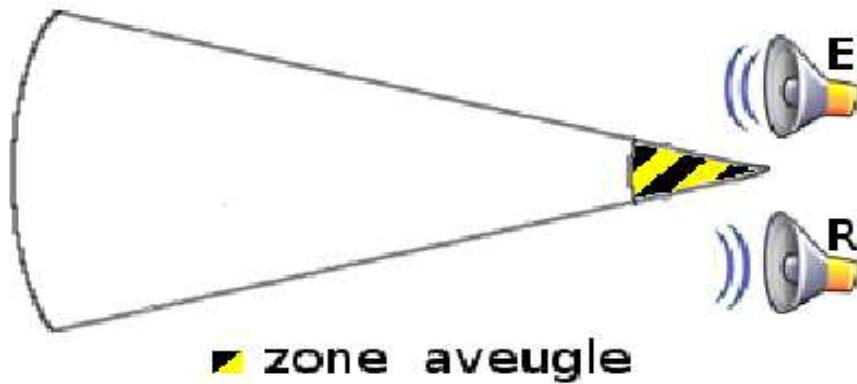


Fig.2.11 Zone aveugle d'un capteur ultrason

- **Télémètre laser :** Un moyen important pour obtenir des mesures précises de distance. Leur principe de fonctionnement est simple, ils estiment une distance à partir d'un calcul de différence de phase.
Ils émettent un rayon laser, c'est-à-dire une onde lumineuse à une fréquence de 800 nm. La réflexion de cette onde donne un écho qui est détecté au bout d'un temps proportionnel à la distance capteur/obstacle.



Fig.2.12 Télémètre NEATO

9. Localisation et cartographie :

Afin d'exécuter le mouvement planifié, le robot doit se localiser dans son environnement en estimant la position et l'orientation par rapport à un repère fixe. L'estimation peut s'effectuer soit par une mesure des déplacements du robot, soit par une mesure de sa position absolue dans l'environnement. Plusieurs approches de localisation sont utilisées en robotique mobile. Elles peuvent être classées en trois catégories: **localisation basée sur une carte (GPS)**, **localisation par rapport à des balises** et **localisation par rapport à d'autres robots**.

9.1. Marier les données multi capteurs et cartographie :

La localisation d'un robot mobile s'effectue par la mise en correspondance de différentes sources extéroceptives et proprioceptives. Généralement il s'agira de confronter les mesures de déplacements prises par odométrie avec une méthode de localisation absolue : soit reconnaissance et calcul de distance par rapport à des balises de position connue, soit mise en correspondance avec une carte construite en ligne et/ou présente dans une base de données, soit encore localisation externe du robot par des capteurs dans l'environnement (GPS).

L'objectif principal de notre système est d'avoir fusionner les deux types de localisation afin de faire un compromis entre la notion de télécommande et la localisation absolue (GPS).

10. Suivi de trajectoire :

Le suivi de trajectoire est une mission importante pour un robot mobile. Il consiste à calculer les commandes envoyées aux actionneurs permettant de réaliser le mouvement planifié.

Un robot est un système dynamique commandé par bouclage pour la poursuite de sa trajectoire de référence.

11. Evitement des obstacles :

Le suivi de la trajectoire planifiée ne permet pas de garantir l'absence des collisions avec les objets statiques ou dynamique existants. L'évitement d'obstacles est un comportement de base présent dans quasiment tous les mouvements des robots mobiles. Ces collisions peuvent se produire lors de l'exécution de la trajectoire, dues à une localisation imparfaite, un plan imprécis ou des obstacles qui n'étaient pas dans le modèle de l'environnement utilisé pour la planification de trajectoire. Il faut donc ajouter des senseurs qui permettront au robot de détecter et d'éviter les obstacles. La première chose à faire est de

connaitre l'étendue de la zone de détection souhaitée afin de choisir les senseurs à utiliser. Comme mis en évidence au précédent, plusieurs types de senseurs comme un laser, des sonars ou des détecteurs de distance infrarouge sont couramment utilisées pour cette tache. Le laser a l'avantage d'être très précis et il couvre une zone importante. Les sonars ont l'avantage de couvrir une zone plus large que les infrarouges qui détectent seulement en ligne droite.

Etant donné les limitations sur chacun des capteurs (précision, exactitude, taux de rafraîchissement, couverture, détection de différents matériaux..), il s'avère utile d'en combiner plusieurs pour assurer une détection plus fiable et robuste.

12. Méthodes de navigation :

Il est possible au robot de se situer dans l'environnement d'opération et de planifier un trajet pour se rendre d'un point A à un point B. Pour y arriver, il est besoin de suivre l'une des stratégies de navigation permettant à un robot mobile de se déplacer pour rejoindre son but.

Les approches utilisables actuellement cherchent plutôt à exploiter des cartes préétablies.

Pour être en mesure de localiser le robot sur l'environnement, il faut choisir entre différentes approches suivantes : [3]

12.1. Navigation métrique: Cette stratégie permet de calculer le chemin le plus court entre deux lieux mémorisés, en divisant l'environnement en plusieurs cellules de dimensions identiques. Une valeur est attribuée à chaque case en fonction de la probabilité qu'un obstacle empêche le robot de se déplacer au centre de celle-ci, ça permet même de planifier des raccourcis au sein de zones inexplorées de l'environnement.

Cette technique permet au robot de planifier des chemins au sein de zones inexplorées de son environnement.

Elle mémorise pour cela les positions métriques relatives des différents lieux, en plus de la possibilité de passer de l'un à l'autre. Ces positions relatives permettent, par simple composition de vecteurs, de calculer une trajectoire allant d'un lieu à un autre, même si la possibilité de ce déplacement n'a pas été mémorisée sous forme d'un lien.

12.2. Navigation géométrique : L'environnement est alors représenté par une série de formes géométriques simples comme des points ou des droites. Le localisateur prend en compte les lectures des capteurs de proximité et il essaie de retirer les formes géométriques qui sont présentes autour du robot. La position des différentes formes et la position de celles-ci les unes par rapport aux autres sont les critères pour identifier la position du robot.

Le problème majeur est qu'il est parfois difficile ou même impossible de créer une carte précise due à la présence de formes trop complexes à modéliser. De plus, les

différents obstacles sont définis par leur forme ainsi que par leur position dans l'environnement (coordonnées absolues).

12.3. Navigation topologique : Cette stratégie permet de mémoriser un ensemble de lieux et les possibilités de passer de l'un à l'autre, indépendamment de tout but. Pour rejoindre un but, il faut alors une étape de planification qui permet de rechercher, parmi tous les chemins possibles, le chemin rejoignant le but. Dans cette technique, des points de repères sont utilisés ici comme références pour identifier des lieux dans l'environnement, et un graphe d'adjacence vient représenter les accès possibles à ces lieux.

12.4. Navigation par action associé à un lieu : cette méthode requiert une représentation interne de l'environnement qui consiste à définir des lieux comme des zones de l'espace dans lesquelles les perceptions restent similaires, et à associer une action à effectuer à chacun de ces lieux.

L'enchaînement des actions associées à chacun des lieux reconnus définit une route qui permet de rejoindre le but. Ces modèles permettent donc une autonomie plus importante mais sont limités à un but fixé. Une route qui permet de rejoindre un but ne pourra en effet pas être utilisée pour rejoindre un but différent. Changer de but entraînera l'apprentissage d'une nouvelle route.

12.5. Navigation par guidage : cette capacité permet d'atteindre un but qui n'est pas un objet matériel directement visible, mais un point de l'espace caractérisé par la configuration spatiale d'un ensemble d'objets remarquables. La stratégie de navigation, souvent une descente de gradient également, consiste alors à se diriger dans la direction qui permet de reproduire cette configuration. Cette stratégie utilise également des actions réflexes et réalise une navigation locale qui requiert que les amers caractérisant le but soient visibles.

12.6. Navigation par approche d'un objet : cette capacité de base permet de se diriger vers un objet visible depuis la position courante du robot. Elle est en général réalisée par une remontée de gradient basée sur la perception de l'objet. Cette stratégie utilise des actions réflexes, dans lesquelles chaque perception est directement associée à une action. C'est une stratégie locale, c'est-à-dire fonctionnelle uniquement dans la zone de l'environnement pour laquelle le but est visible.

13. Conclusion :

Ce chapitre nous a permis d'illustrer toutes les informations et les généralités acquises sur la conception (mécanique, électronique ou de contrôles) et les différents concepts indispensables comme la navigation et l'évitement des obstacles qui vont donner au robot sa fonctionnalité souhaitée.

Le prochain chapitre va inclure l'étapes finale où la définition des composantes utilisées, les testes et simulations, le montage finale du robot vont avoir lieu.

Chapitre

03

1. Introduction :

Dans ce chapitre, on s'intéresse à la réalisation du robot et à l'implémentation de l'étude théorique dans notre modèle.

D'abord, nous avons montré les pièces que nous aurons besoin et quelles fonctionnalités que nous voulions mettre en œuvre dans notre robot.

Puis on passe à la conception des circuits des différentes parties du robot où on va montré comment raccorder et tester les composantes du robot.

Nous allons introduire l'environnement de développement Arduino et enfin écrire quelques programmes pour démontrer comment contrôler chacune des parties du robot. Ensuite, on va élaborer un software complet pour assembler toutes les croquis Arduino dans un seul sketch, l'étape qui va nous facilite la manipulation dans le programme final sur Arduino.

2. La conception du robot :

Ce schéma montre la structure globale qu'on va suivre pour la réalisation de notre robot différentiel : (Fig.3.1)

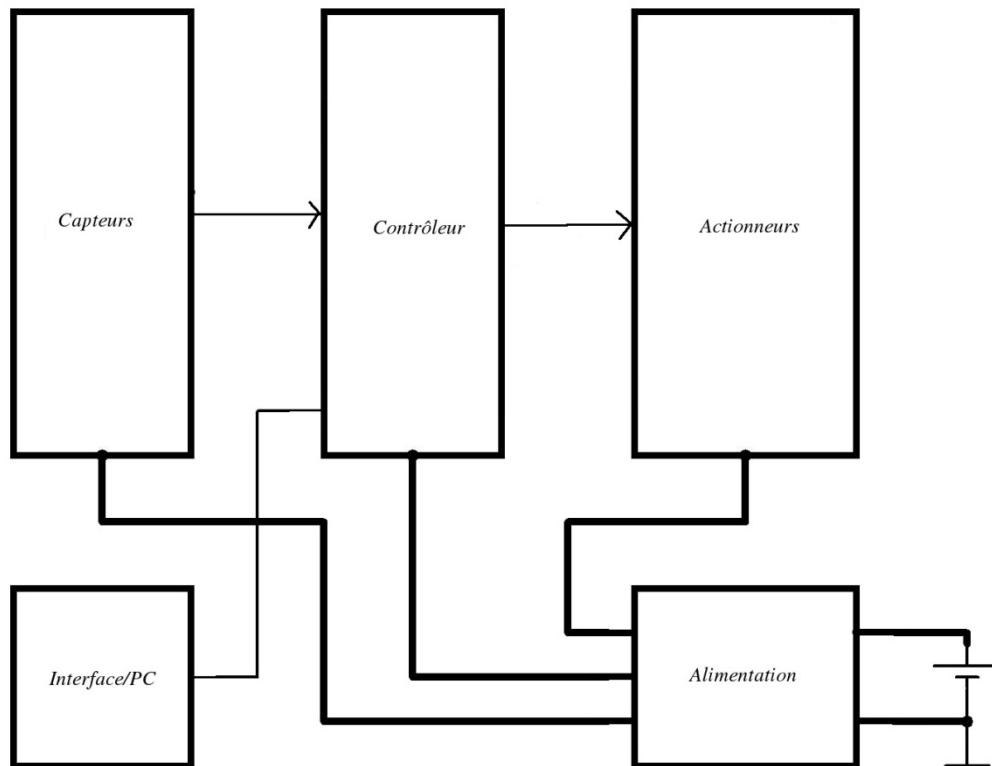


Fig.3.1 Schéma des différentes unités du robot
LE NUMERO 1 MONDIAL DU MÉMOIRES

3. Cahier de charge :

Notre objectif est de réaliser un robot à vitesses différentielles qui a les caractéristiques suivantes :

- Il devrait être un véhicule qui peut aller en l'avant, en arrière et tourner.
- Il devrait être facile à monter et à démonter.
- Il devrait y avoir un mode de déplacement avec la capacité de détecter des obstacles avant et les éviter.
- Il devrait avoir un mode de suivi d'une ligne à l'aide des capteurs de réflecteur IR disposés dans le châssis.
- Il devrait avoir un mode dans lequel il peut être entièrement contrôlé à partir d'un Smartphone Android.
- Il devrait être facile de manipuler, modifier et améliorer.

On peut déjà remarquer l'émergence de trois (03) modes :

- Mode suiveur de ligne (Line follower robot)
- Mode remote controlled (RC)
- Mode intelligent

4. Les composants hardwares :

Pour la plate-forme hardware, on a décidé de choisir le microcontrôleur Arduino avant de commencer l'élaboration de la base mécanique de robot, c'était le choix optimal pour notre robot car on peut facilement agencer les autres composantes autour de lui grâce à son plateforme d'entrées/sorties permettant de dialoguer (en temps réel) et exécuter les commandes, et aussi pour son interface qui est largement utilisé partout dans le monde ce qui signifie qu'il y'a beaucoup d'informations et de ressources disponibles.

Avec la liste des fonctionnalités qu'on établit au dessus, on peut aller chercher les pièces nécessaires pour la construction de robot.

Voici l'ensemble des composants qu'on va utiliser pour la conception de robot :

- Carte Arduino Uno
- Circuit intégré (L293D)
- Capteur de distance (Capteur ultrason HC-SR04)
- Liaison Bluetooth (HC-05)
- Plaque d'essai (breadboard) et câbles
- Les moteurs CC.
- Source d'alimentation (batteries)

- Kit de véhicule (châssis et roues)

4.1. Carte Arduino UNO :

La carte Arduino Uno sera le cerveau du robot, car il sera en cours d'exécution le composant qui permettra de contrôler toutes les autres parties. (Fig.3.2)

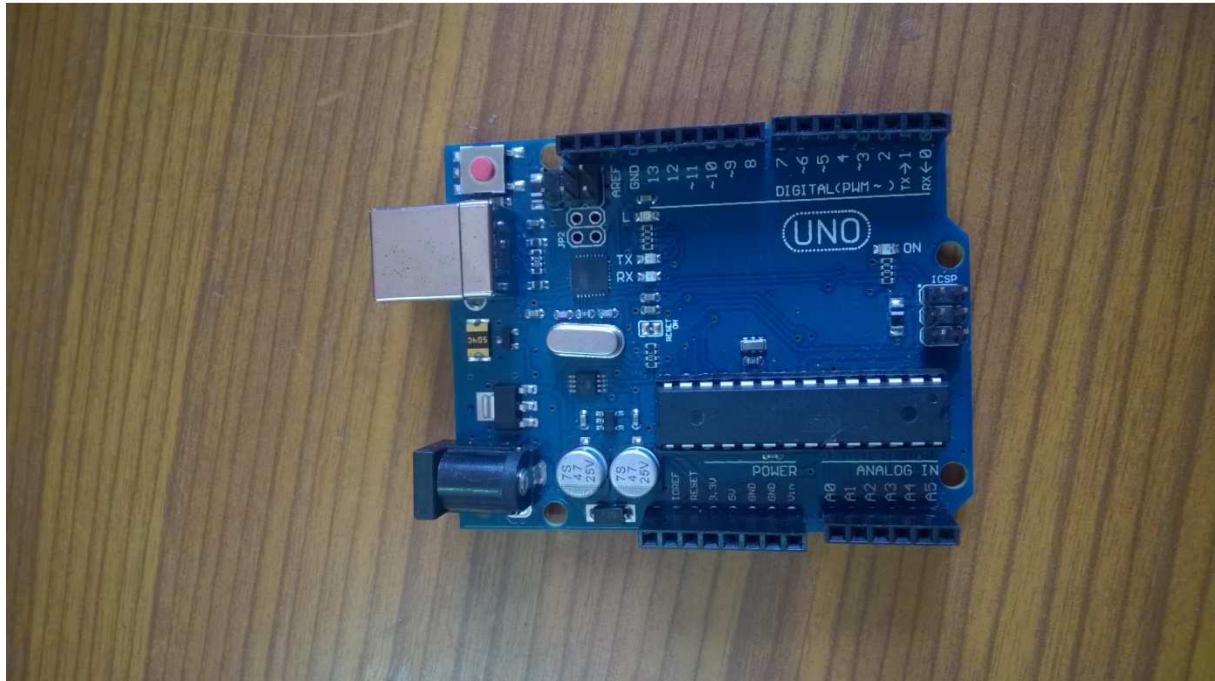


Fig.3.2 Carte Arduino Uno

L'Arduino Uno est une carte microcontrôleur basée sur l'ATmega328P. Il dispose de 14 broches numériques d'entrée / sortie (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un cristal de quartz 16 MHz, une connexion USB, une prise d'alimentation, d'une embase ICSP et un bouton de réinitialisation. (Fig.3.3)

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Fig.3.3 Propriétés de la carte Arduino Uno [10]

4.2. Circuit L293D :

La carte Arduino ne peut pas contrôler directement un moteur. La partie délicate est d'être en mesure de faire fonctionner le moteur de manière sélective vers l'avant ou vers l'arrière, ce qui nécessite l'échange des entrées d'alimentation et de masse dans le moteur.

On utilise pour notre projet le L293D est très ancien et bien documenté, très facile à mettre en œuvre. Il s'agit d'un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur.

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts grâce à ses 4 canaux. (Fig.3.4)

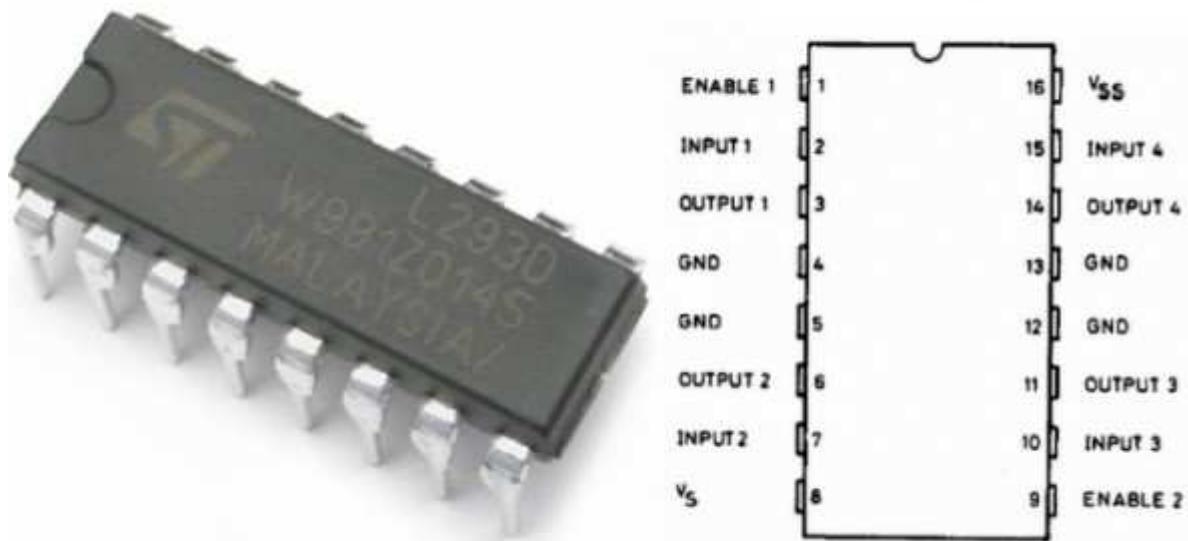


Fig. 3.4 Circuit L293D et son schéma [11]

4.3. Capteur de distance :

On va utiliser ce dispositif pour empêcher le robot de frapper les murs ou d'autres obstacles sur son chemin.

Nous avons choisi le capteur ultrason **HC-SR04**, car il dispose simplement de 4 pins: VCC, TRIG, ECHO, GND. Il est donc très facile de l'interfacer à un microcontrôleur.

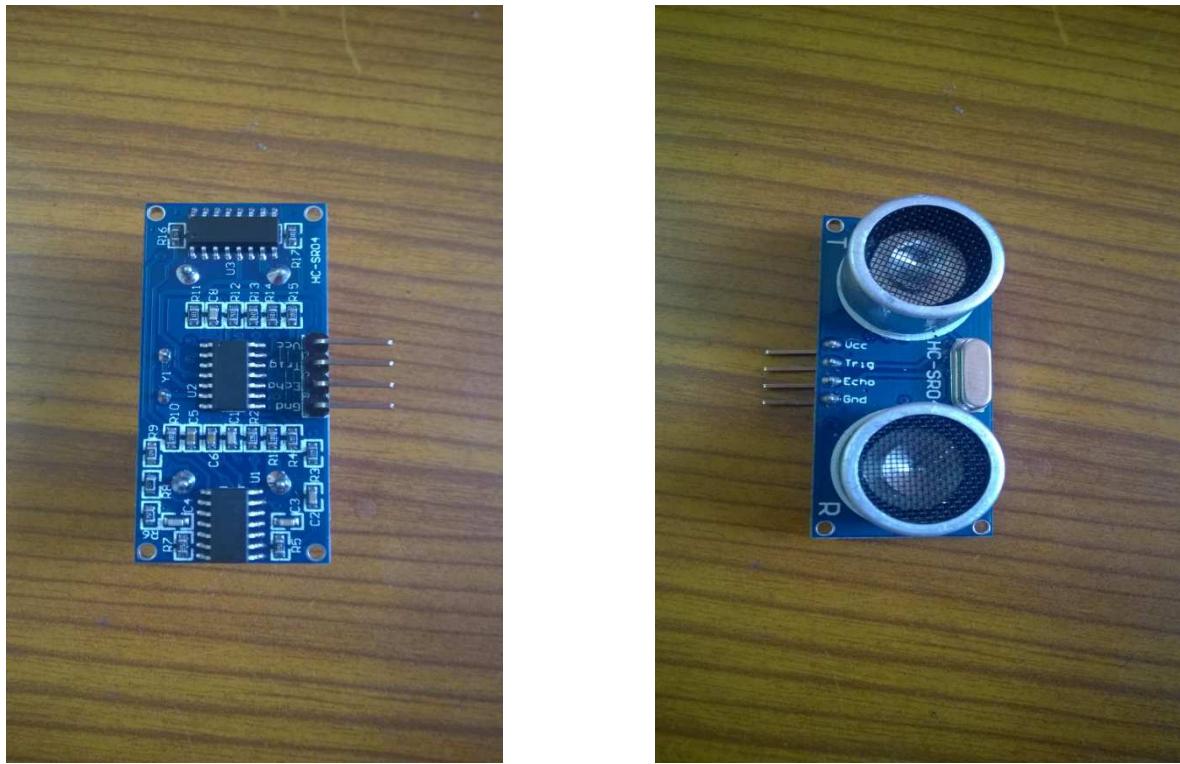


Fig.3.5 Capteur Ultrason HC-SR04

Le fonctionnement du module est le suivant :

Il faut envoyer une impulsion niveau haut (5v) pendant au moins $10 \mu\text{s}$ sur la broche ‘**Trig Input**’ cela déclenche la mesure. En retour la sortie ‘**Output**’ ou ‘**Echo**’ va fournir une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet.

Afin de pouvoir calculer la distance en cm, on utilisera la formule suivante :

Distance = (durée de l'impulsion) (en μs) / 58 (basée sur la formule de la vitesse du son)

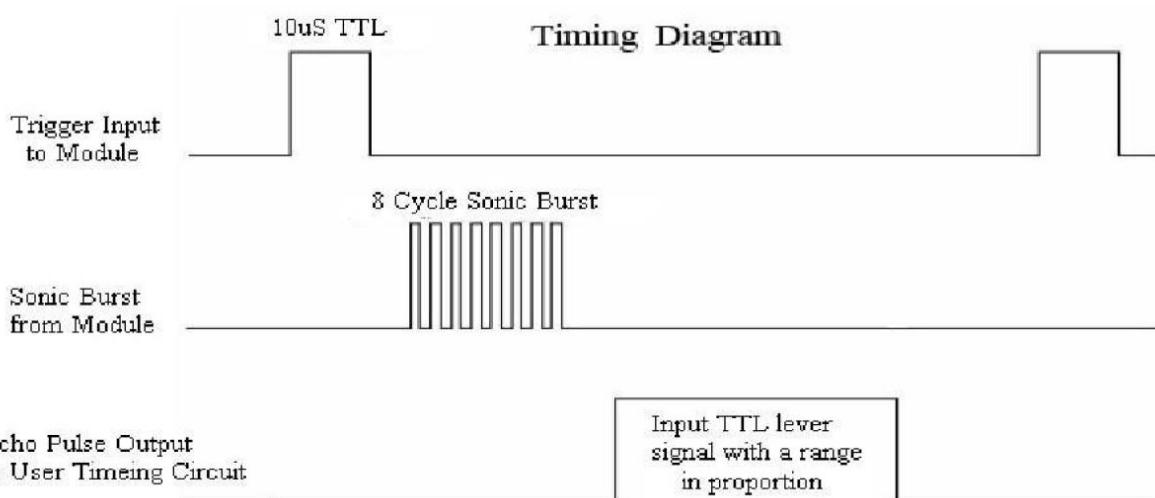


Fig.3.6 Fonctionnement du capteur ultrason

4.4. Capteur de couleurs :

Le CNY70 est un capteur de réflexion qui comprend une connexion infrarouge émetteur et phototransistor dans un paquet au plomb qui bloque la lumière visible. (Fig.3.7)



Fig.3.7 Capteur CNY70

4.5. Module Bluetooth :

La meilleure façon de contrôler le robot à partir d'un Smartphone est via l'interface Bluetooth que tous les Smartphones modernes ont. Le téléphone va agir comme émetteur de commandes, donc on aura besoin d'un récepteur Bluetooth dans le robot.

Le HC-05 agit comme un port série à travers lequel nous pouvons envoyer et recevoir des données. Donc, en utilisant un terminal série ou une application Bluetooth personnalisée sur l'ordinateur ou par téléphone, nous pouvons contrôler et surveiller notre projet. (Fig.3.8)

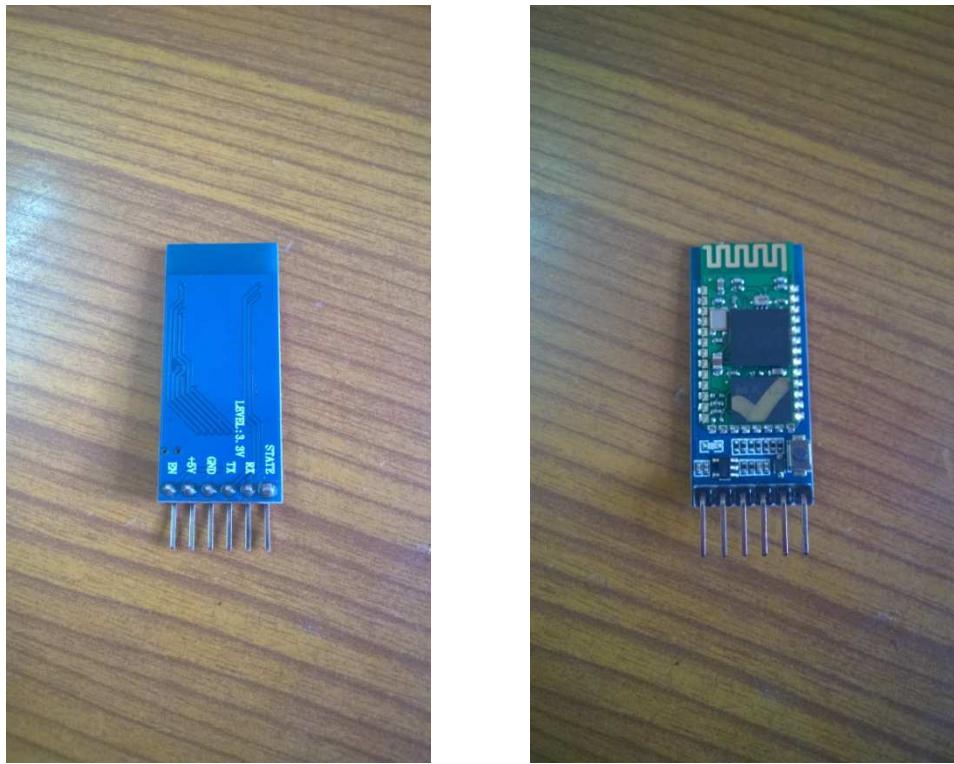


Fig.3.8 Module Bluetooth HC-05

4.6. Plaque d'essais et câbles :

L'une des restrictions que nous avons décidé d'imposer dans ce projet est que nous ne ferions pas des opérations de soudage, de sorte que nous pouvons assembler et à démonter le robot sans détruire les pièces.

Ensuite, il nous fallait une plate-forme où l'on peut facilement connecter tous les composants ensemble. Pour ce genre de tâche, nous avons utilisé une plaque d'essai (Breadboard microtivity IB401).

C'est un dispositif qui permet de réaliser le prototype d'un circuit électronique et de le tester. L'avantage de ce système est d'être totalement réutilisable, car il ne nécessite pas de soudure. (Fig.3.9)

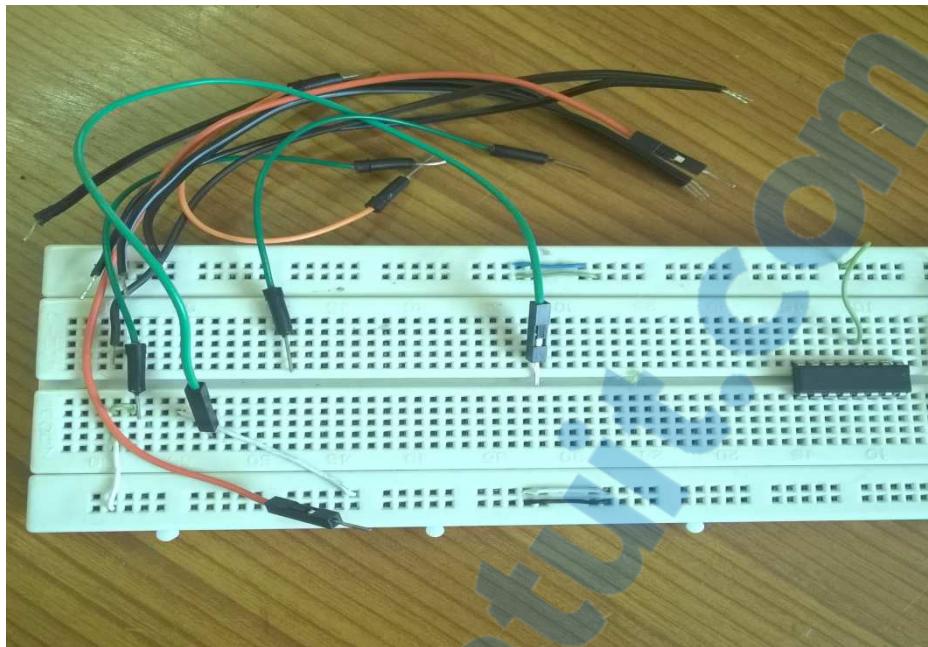


Fig.3.9 Breadboard microtivity IB401 & Jumpers

4.7. Moteurs à courant continu :

Parce qu'on a décidé de réaliser un robot différentiel, alors on utilise deux moteurs à courant continu pour commander les deux roues arrière.

Évidemment, pour pouvoir valider un moteur, il faut connaître les spécifications que nous voulons atteindre.

On utilise pour notre projet les moteurs à CC RS-775, ils sont facile à utiliser car ils sont programmables depuis une très large gamme de langages de programmation (C#, Java, Python, Labview, Matlab...) (Fig.3.10)

Ses spécifications :

- Courant Continu (A):1-30A
- Puissance de sortie:5-300W
- Type: Micro moteur
- Construction: Aimant permanent
- Certification: CCC, CE, ROHS
- Couple:0.1-5NM
- Vitesse(RPM):1000-32000rpm
- Tension (V):12V
- Dimensions: 47*60mm



Fig.3.10 Les deux moteur DC

4.8. Batteries d'alimentation :

Pour alimenter notre robot on va utiliser un module teadStudio de porte piles AA avec 4 cellules, de configuration carrée pour donné une source externe aux moteurs et une batterie 9v pour alimenté la carte Arduino. On à, aussi, utilisé une pile à transistor 9 volts Metama en raison de ses paramètres idéal pour notre configuration (volume d'environ 21 cm³). (Fig.3.11)

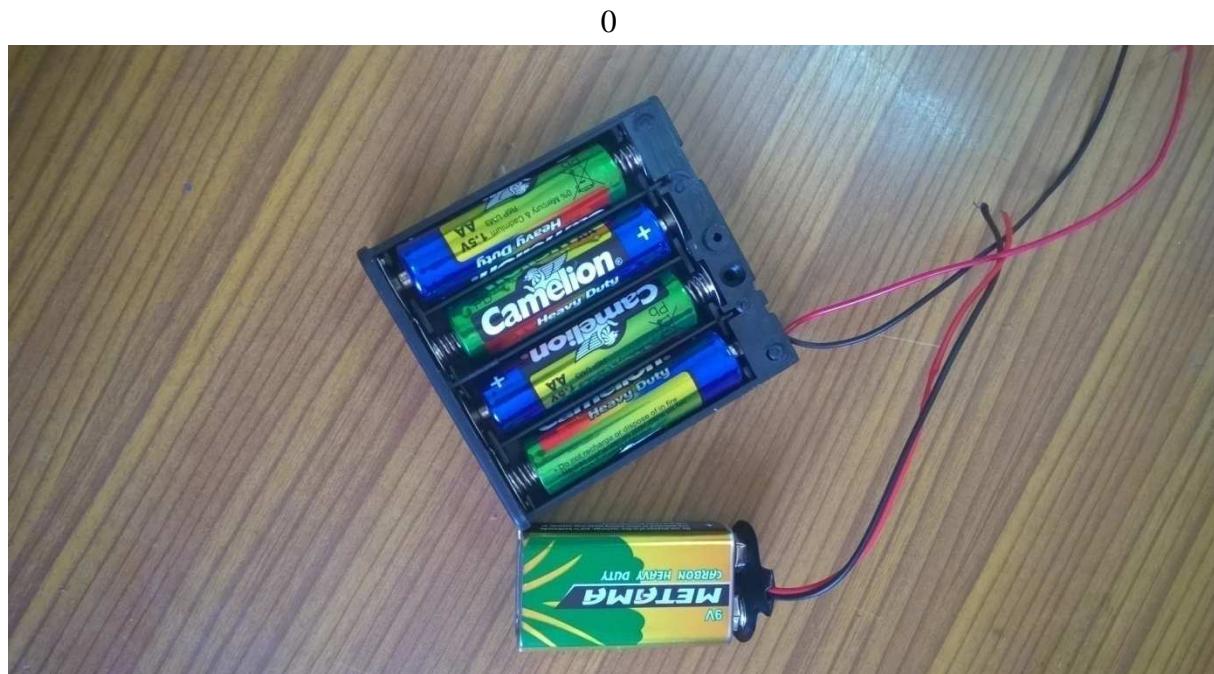


Fig.3.11 Batteries d'alimentation

4.9. Châssis de véhicule :

Il y a beaucoup de choix pour notre corps de robot. Nos seules exigences étaient qu'il disposait d'une grande plate-forme où toutes les parties peuvent être montées et qu'il est compatibles avec l'emplacement des roues et des moteurs. Sans être très sophistiqué, le châssis définit le type de stratégie choisie face à la structure globale utilisée, lors de la création de notre robot.

Il faut organiser tous les éléments pour optimiser le robot : centre de gravité, position des capteurs, moteurs, pneus, batteries, cartes Arduino, plaque d'essai. Il y a une infinité de solutions possibles ayant chacune ses avantages et inconvénients.

On a fabriqué le châssis en plexiglas pour ses propriétés physiques (excellente résistance aux agents atmosphériques, légèreté : masse volumique de $1,19 \text{ g}\cdot\text{cm}^{-3}$ [4] (beaucoup plus léger que le verre)).

Les dimensions sont de 24 cm x 15 cm de côté, sans limitation de hauteur, pour un poids de 275 g.

Le robot est équipé par deux roues parallèles et coaxiales, elles sont munies de pneus en caoutchouc qui permettent de rouler sans glisser. Chaque roue est actionnée séparément par un moteur à CC. (Fig.3.12)



5. MODE 01 : Line follower robot

La fonctionnalité de ce mode est que le rebot soit capable de suivre une ligne tracée au sol en utilisant ses capteurs de vision (vision sensors).

Le modèle qu'on a choisi est constitué de trois (03) capteurs de couleurs CNY70 monté à l'avant du châssis (Leftsensor, Middlesensor et Rightsensor). Lorsque le capteur est en face du sol (blanc), son émetteur reçoit de la lumière et transmet cette information au micro contrôleur. Dans le cas où le capteur est sur une surface noire, aucune information ne serait transmise.

Les moteurs du rebot vont être commandés en se basant des informations reçues des capteurs par le micro contrôleur.

Le guidage de notre robot est fait par une bande noire tracée sur une surface blanche. Pour que le rebot fasse un mouvement tout droit l'OUTPUT des capteurs doit être en état normal. On a défini cette état par 101 où les chiffres 0 et 1 sont l'information transmise au micro contrôleur par les capteurs. L'état normal consiste à Leftsensor et Rightsensor sont face au sol et le Middlesensor est face à la bande noire. Lorsque cet état est perturbé, dans le cas des virages ou courbe, une action corrective aura lieu par la procédure suivante :

Si le Leftsensor commence à transmettre un 0 ce qui est lié à un virage à gauche, le moteur gauche s'arrête et le rebot tourne à gauche jusqu'à l'état normal est atteinte.

Si le Rightsensor change son OUTPUT en 0 donc le rebot doit tourner à droite en coupant le signal du moteur droit jusqu'à l'obtention de l'état normal.

5.1. Simulation du line follower par V-REP Pro EDU :

L'illustration du fonctionnement du mode suiveur de ligne est faite par logiciel V-REP

V-REP est un simulateur de robot avec un environnement de développement intégré. Cette application peut être utilisée pour: le développement d'algorithmes rapides, les simulations d'automatisation d'usine, le prototypage rapide et la vérification, la robotique éducation, la surveillance à distance liées, la sécurité de double-checking, etc. Les contrôleurs peuvent être écrites en C / C, Python, Java, Lua, Matlab, Octave ou Urbi. [12]

Au début on doit créer une scène de simulation où on peut tester la logique de notre robot suiveur de ligne. Ce modèle va nous permettre de voir la simulation en temps réel même avant d'injecter le programme Arduino.

Après on doit définir le routage sur lequel notre rebot doit utiliser comme une infrastructure de guidage. On clique avec la droite de souris et choisir « Add » « Path » « Segment type ». On ouvre le Path Edit Mode. La création d'un chemin est faite par l'ajout des nouveaux points « Insert new control point after selection » et la translation par rapport du point précédent. (Fig.3.13)

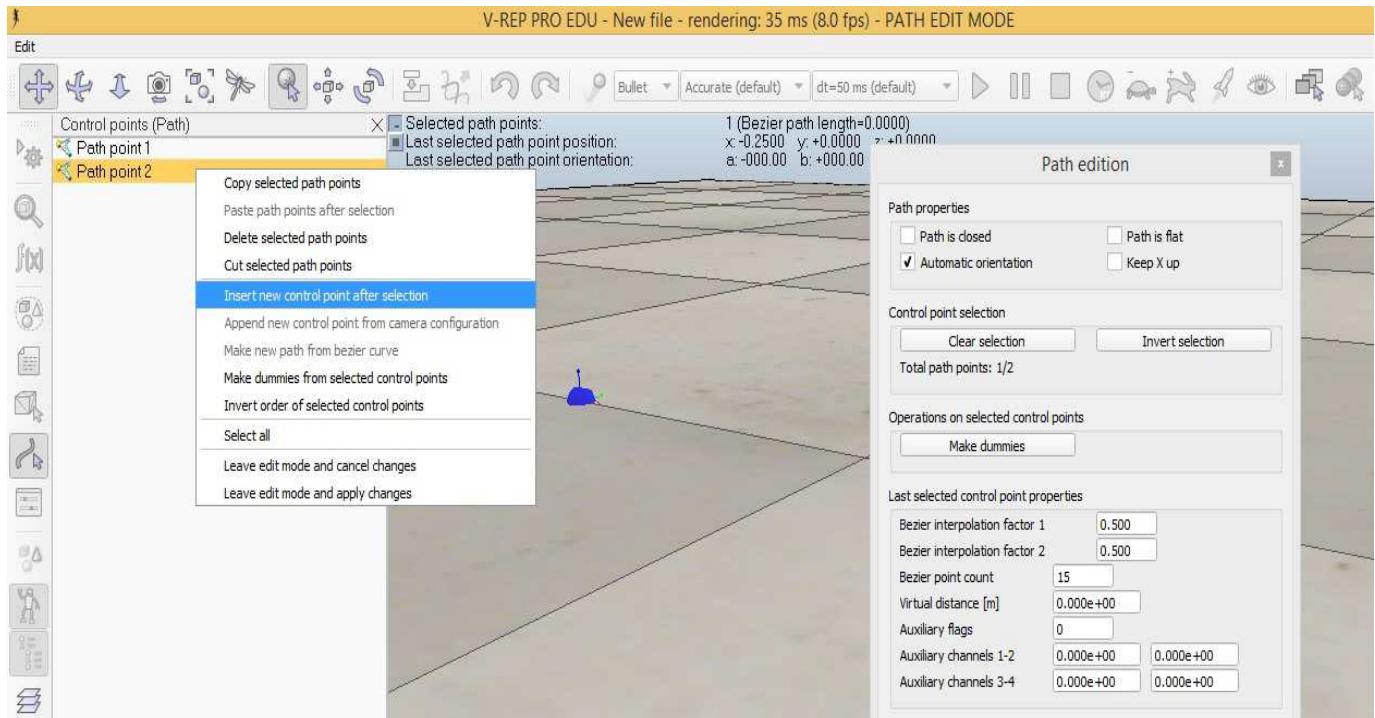


Fig.3.13 Création de routage par V-REP

Après qu'on a crée plusieurs point de contrôle on doit cliquer sur « Path is closed » sur la fenêtre du Path edition pour avoir un routage fermé.

Pour éviter d'avoir des virages serrés qui peuvent introduire une instabilité au robot on peut changer le « Bezier interpolation factors » pour qu'ils soient dans l'intervalle [0.8 0.99] (Fig.3.14)

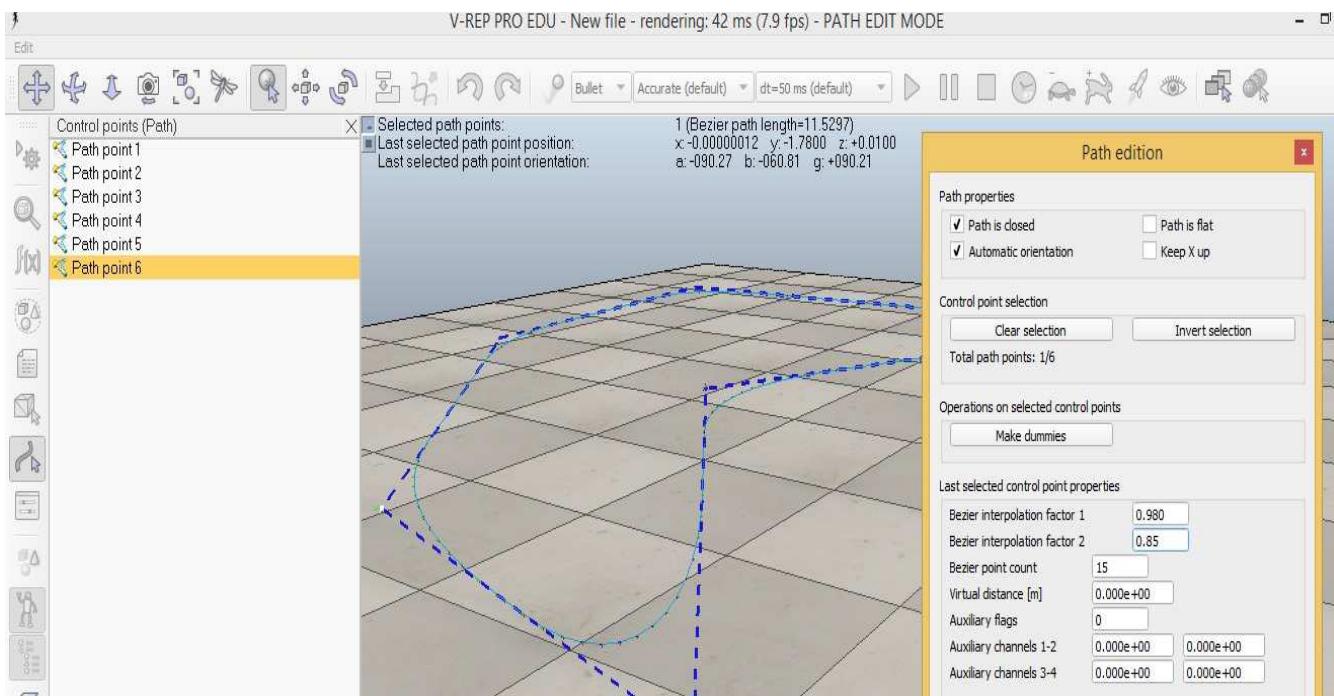


Fig.3.14 Routage avec des virages ouverts

L'étape suivante est de modifier ce routage pour qu'il soit une bande tracée sur le sol. Pour cela on doit ouvrir la fenêtre du « Path Shapping Parameters », de cliquer sur « Path shipping enabled », de changer la couleur par « Adjust color » et de changer le type en «Horizontal Segment » avec un « Scaling factor » égale à 5. (Fig.3.15)

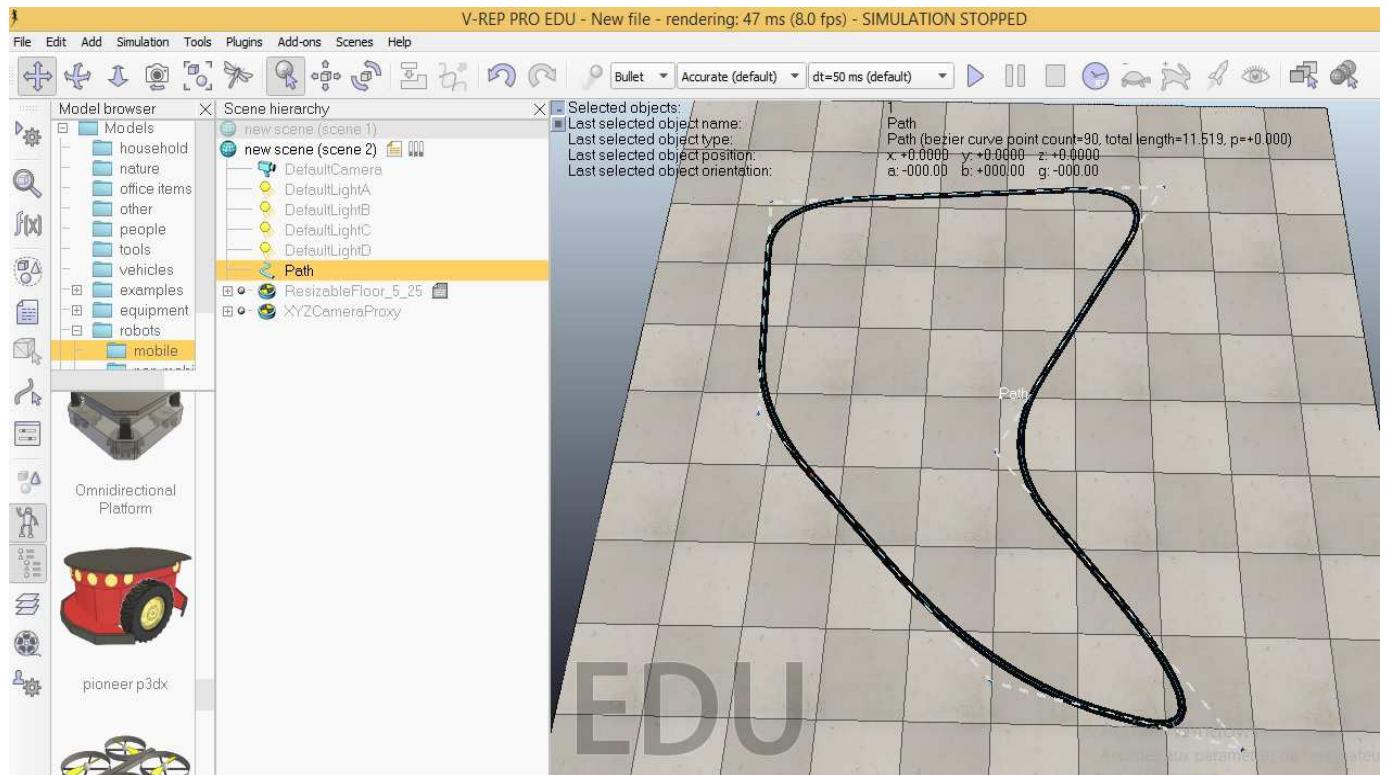


Fig.3.15 Routage finale du rebot

Maintenant que le routage est installé, On doit importé le rebot. Dans le V-Rep il y'a des bibliothèques des rebots différents prédefinis et ces modèles peuvent être modifiés.

On a choisi le modèle LineTracer, ce modèle est une plate-forme différentielle. Dans ce modèle on peut implémenter nos capteurs par « Add » « Vision sensor » « Perspective type ». On doit changer leurs orientation vers le sol et de les attacher au véhicule par les sélectionner (Le lineTracer le dernier) et de cliquer « Make last selected object parent ». (Fig.3.16)

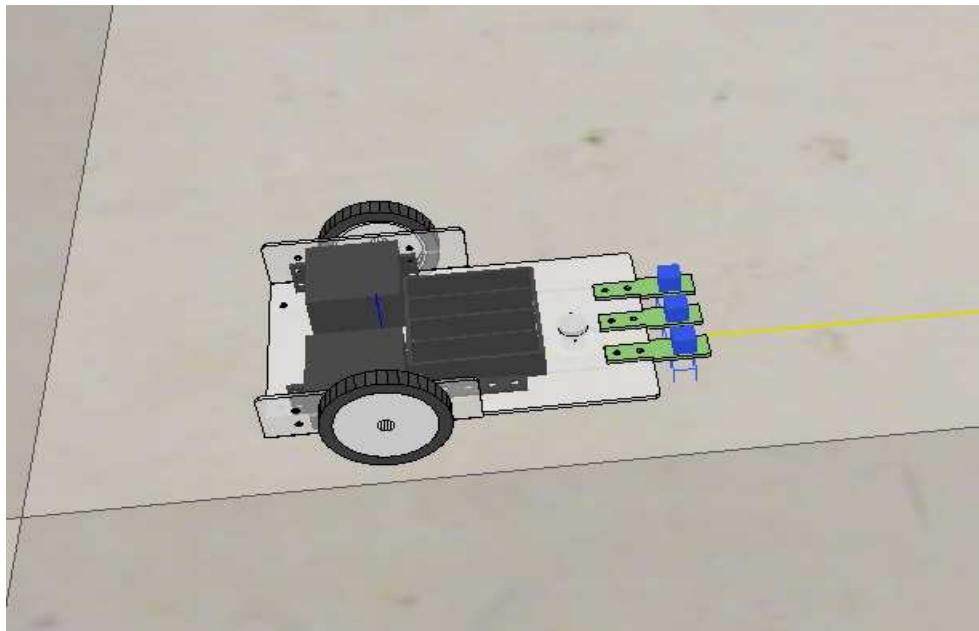


Fig.3.16 LineTracer avec ses trois capteurs

Pour illustrer la vision des capteurs on ajoute une « Floating view ». On fait la liaison avec les capteurs par les sélectionner et de cliquer sur « View » « Associate view with selected camera ».

Finalement, on peut voir le comportement du robot en cliquant sur « Run » de la simulation. (Fig.3.17)

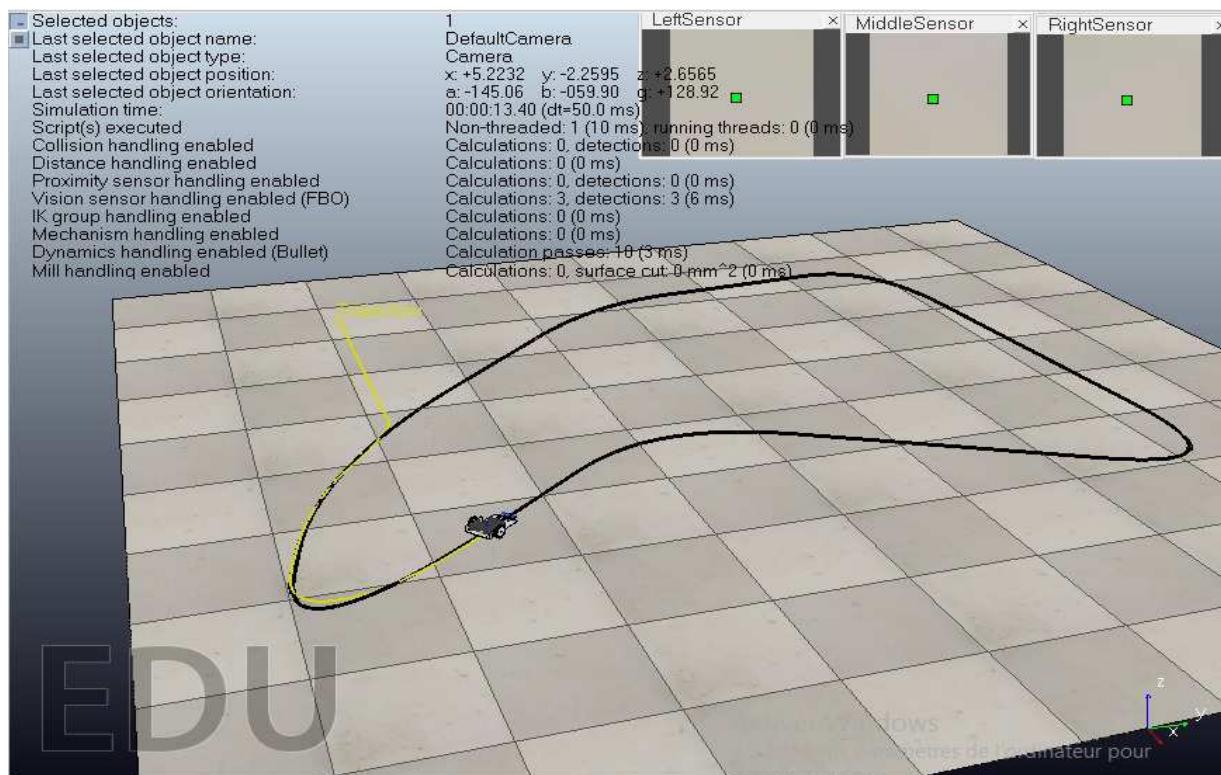


Fig.3.17 Simulation du line follower sur V-Rep

5.2. Circuit du suiveur de ligne :

Le suiveur de ligne contient la carte Arduino, deux (02) moteurs à courant continu, un circuit intégré L293D, trois (03) capteurs CNY70, trois (03) résistance 220 ohm, trois (03) résistance 10k ohm, une batterie.

Le raccordement des résistances avec les capteurs CNY70 peut être fait par plusieurs configurations. On a choisi celle-ci : la cathode du LED IR avec une résistance 220 ohm et la cathode du phototransistor avec une résistance 10 k ohm.

Les deux Vcc doivent être liés avec la sortie 5v d'Arduino, et la valeur OUTPUT du phototransistor avec un PIN digital d'Arduino.

On a illustré le montage complet du suiveur de ligne sur logiciel Fritzing (Fig.3.18)

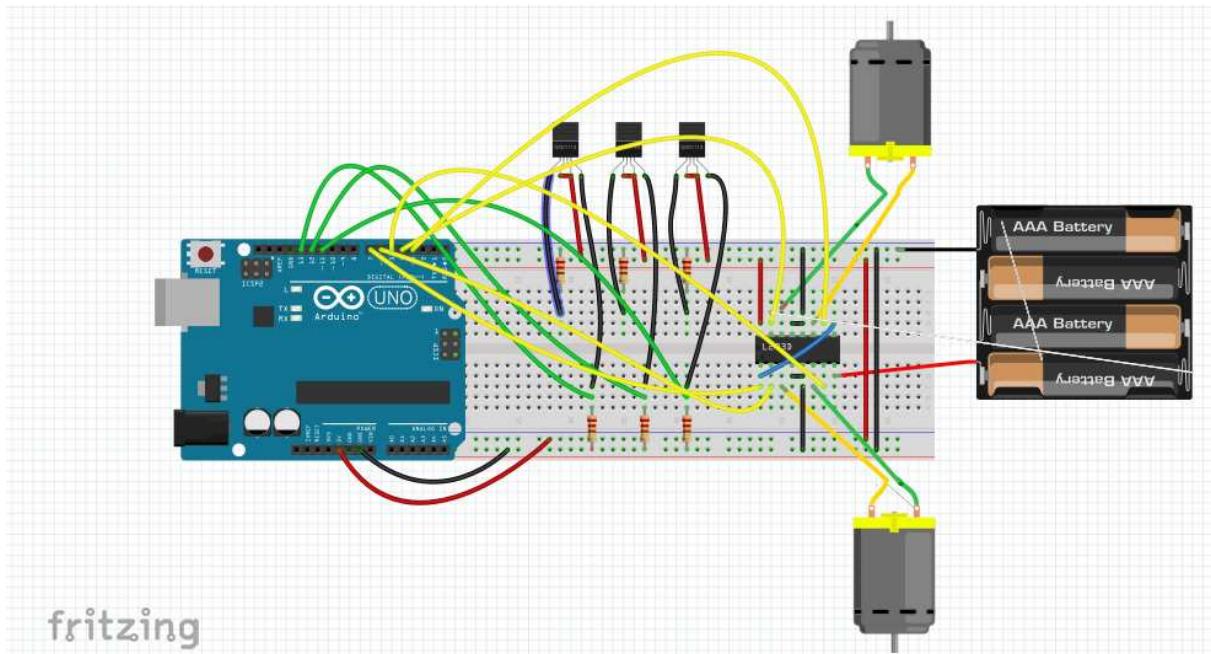


Fig.3.18 Montage suiveur de ligne sur Fritzing

5.3. Programme Arduino :

La dernière partie est celle de l'écriture du code qui va être injecté au robot. On commence par définir les PINs utilisés :

```
// Line Follower Code

// Normal State Definition:
int L=1;
int M=0;
int R=1;

// CNY70 sensors PINs definition:
int Leftsensor=13;
int Middlesensor=12;
int Rightsensor=11;

// L293D Control PINs definition:
int en=7;          // Enable of L293 to PIN7
int m11=6;         // Left motor forward signal
int m12=5;         // Left motor backward signal
int m21=4;         // Right motor forward signal
int m22=3;         // Right motor backward signal
```

Après on doit spécifier le type de ces PINs et comment L'Arduino va les utiliser (comme OUTPUT ou INPUT) donc le « void setup » :

```
void setup() {
    Serial.begin(9600); // Baud speed
    // Used pins types
    pinMode(en,OUTPUT);
    pinMode(m11,OUTPUT);
    pinMode(m12,OUTPUT);
    pinMode(m21,OUTPUT);
    pinMode(m22,OUTPUT);
    pinMode(Leftsensor, INPUT);
    pinMode(Rightsensor, INPUT);
    pinMode(Middlesensor, INPUT);
}
```

Dans le « void loop » on doit décrire le comportement de notre robot. Cette partie doit inclure les réactions du robot face à l'état normale et aux perturbations des virages :

```

void loop() {
    L = digitalRead(Leftsensor);
    M = digitalRead(Middlesensor);
    R = digitalRead(Rightsensor);
    digitalWrite(en, HIGH);

    if ((L==1) && (M==0) && (R==1)) {
        // Normal State : Sensors reading 101
        digitalWrite(m11, HIGH);
        digitalWrite(m12, LOW);
        digitalWrite(m21, HIGH);
        digitalWrite(m22, LOW);
    }
    else if (L==0) {
        // Left turn :
        digitalWrite(m11, LOW);
        digitalWrite(m12, LOW);
        digitalWrite(m21, HIGH);
        digitalWrite(m22, LOW); }
    if (R==0) {
        // Right turn :
        digitalWrite(m11, HIGH);
        digitalWrite(m12, LOW);
        digitalWrite(m21, LOW);
        digitalWrite(m22, LOW); }
}
.

```

5.4. Simulation du système suiveur de ligne par ISIS :

ISIS fait partie de la suite logicielle Proteus permettant de la CAO électronique éditée par la société Labcenter electronics. ISIS est la partie qui permet la création des schémas et la simulation électronique. [13]

Pour notre système on a téléchargé la bibliothèque ARDUINO Uno sur ISIS. On a fait la connexion avec le circuit intégré L293D et ses deux moteurs.

On sait que les trois capteurs CNY70 transmettent une information binaire à la carte Aduino, 0 en cas d'une surface noire et 1 si la surface est blanche. Pour cela on a choisi de les modéliser comme des Logic toggle. (Fig.3.19)

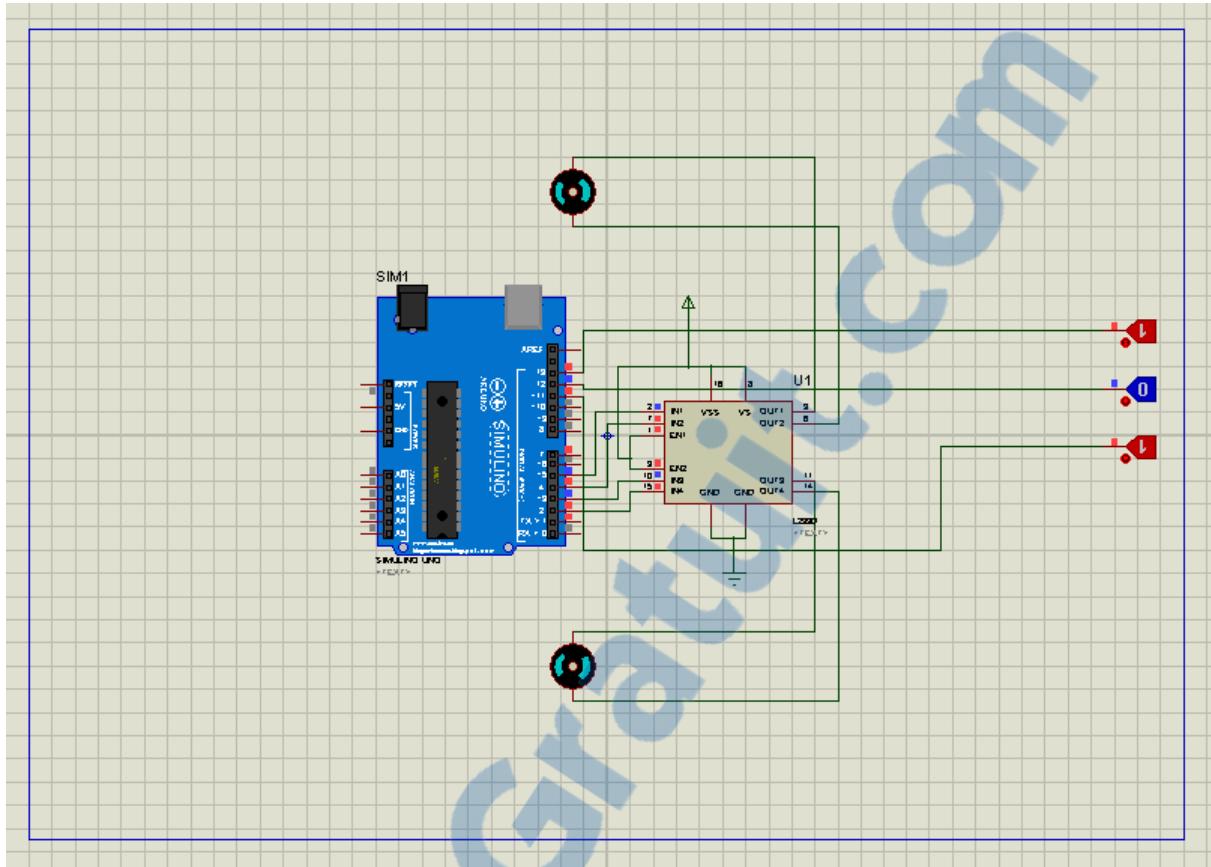


Fig.3.19 Simulation du suiveur de ligne par ISIS

Après avoir finir le schéma, on importe le fichier du programme Arduino avec l’extension « .hex ». Le code du sketch Arduino passe par un compilateur (avr-gcc) qui va transmettre le code lisible par nous à des instructions ou des fichiers objets compris par la machine. Le résultat de cette transformation est un fichier « intel hex » qui contient les bits spécifiques injectés à la mémoire du carte Arduino. [14]

6. MODE 02 : Remote controlled via Bluetooth

On pensait que la communication Bluetooth serait l’aspect le plus complexe du programme, mais après avoir lu la documentation qui est venu avec notre appareil. Nous avons compris que le module Bluetooth fonctionne sur les protocoles de communication standard dans la carte Arduino « Serial », ce qui signifie que nous pouvons utiliser les fonctions de la classe « Serial ».

Une chose importante, Il est impossible de télécharger un croquis sur USB tandis que le module Bluetooth est connecté à la broche 0 et 1, parce que ces broches sont utilisées par la communication USB/Serial.

Lorsque nous avons commencé à écrire le code de commande à distance, nous sommes devenus un peu frustré d'avoir se connecter en permanence et déconnecter Bluetooth et USB. Pour éviter tout ça, nous avons décidé d'utiliser la bibliothèque « SoftwareSerial.h » qui vient

avec le logiciel Arduino pour déplacer la connexion Bluetooth à un ensemble différent de broches.

Le changement est assez simple. Nous avons déplacé les connexions à partir du module Bluetooth qui étaient dans les broches 0 et 1 aux broches digitales 2 et 3.

6.1. Circuit Arduino-Bluetooth HC-05 :

Sur le HC-05 on utilise juste quatre (04) liaisons : 5V, GROUND, Tx et Rx et ils doivent être liés avec la carte Arduino comme suivant :

- Le GROUND avec le PIN GROUND sur Arduino.
- 5V avec la ressource 5 V sur Arduino.
- Tx avec le PIN déclaré Rx dans le croquis.
- Rx avec le PIN déclaré Tx dans le croquis en passant par un diviseur de tension qui va diminuer sa tension avec un tiers. (Le Rx du bluetooth accepte une tension de 3,3 V comme HIGH)

On a illustré le montage du bluetooth avec Arduino sur Fritzing. (Fig.3.20)

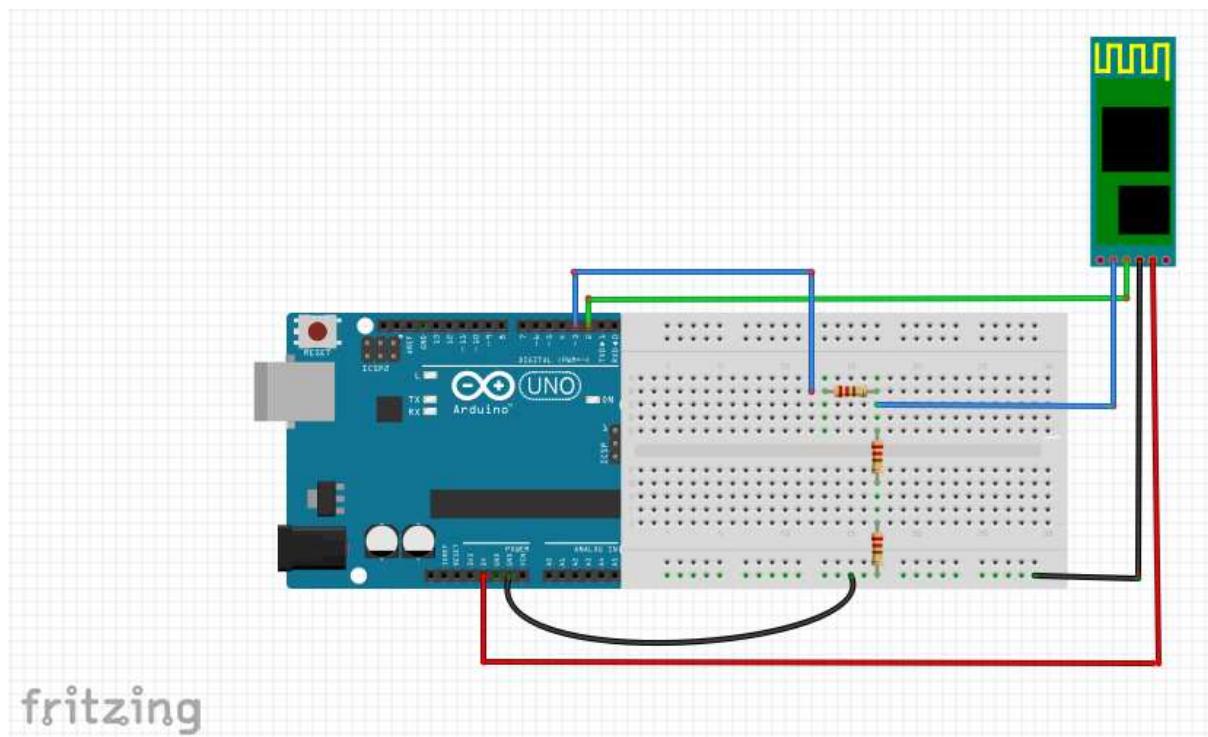


Fig.3.20 Montage bluetooth-Arduino

On a utilisé trois (03) résistance de 1 k ohm comme un diviseur de tension (Voltage divider). Deux (02) résistances en série pour donner une résistance équivalente

$R_{eq} = 1 k\ ohm$ connectée d'une part avec le GROUND et de l'autre part avec le Rx du bluetooth. L'autre résistance est connectée avec le PIN 3 d'Arduino et avec le Rx du bluetooth. (Fig.3.21)

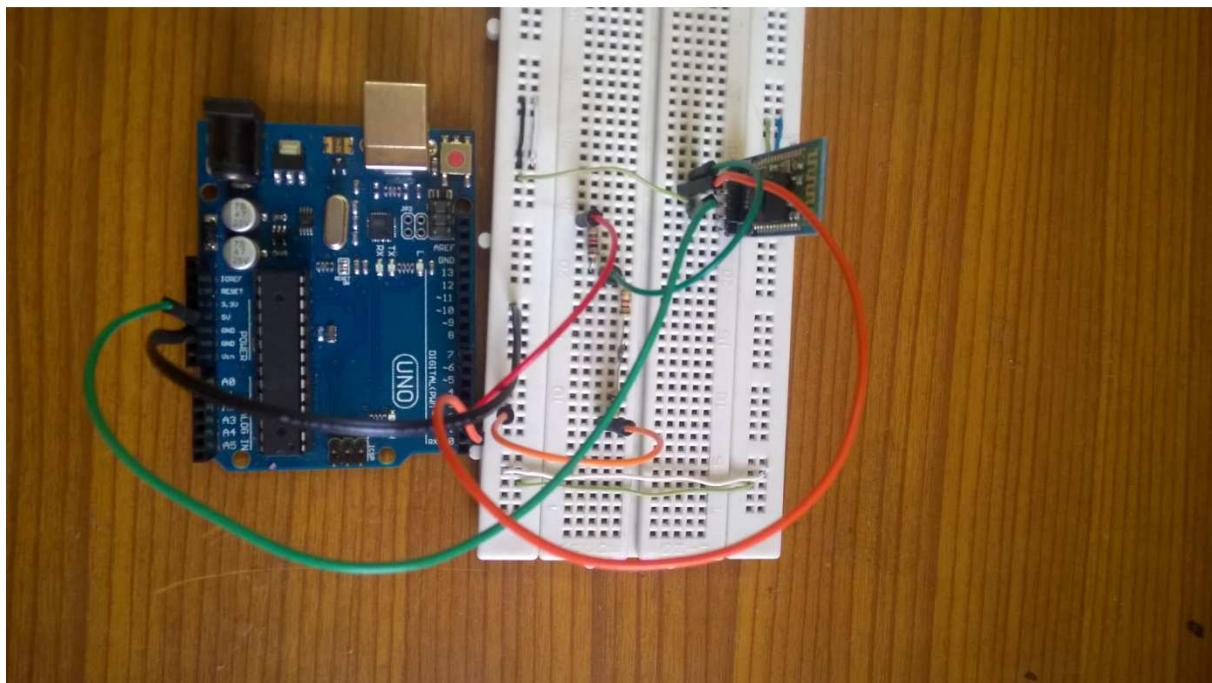
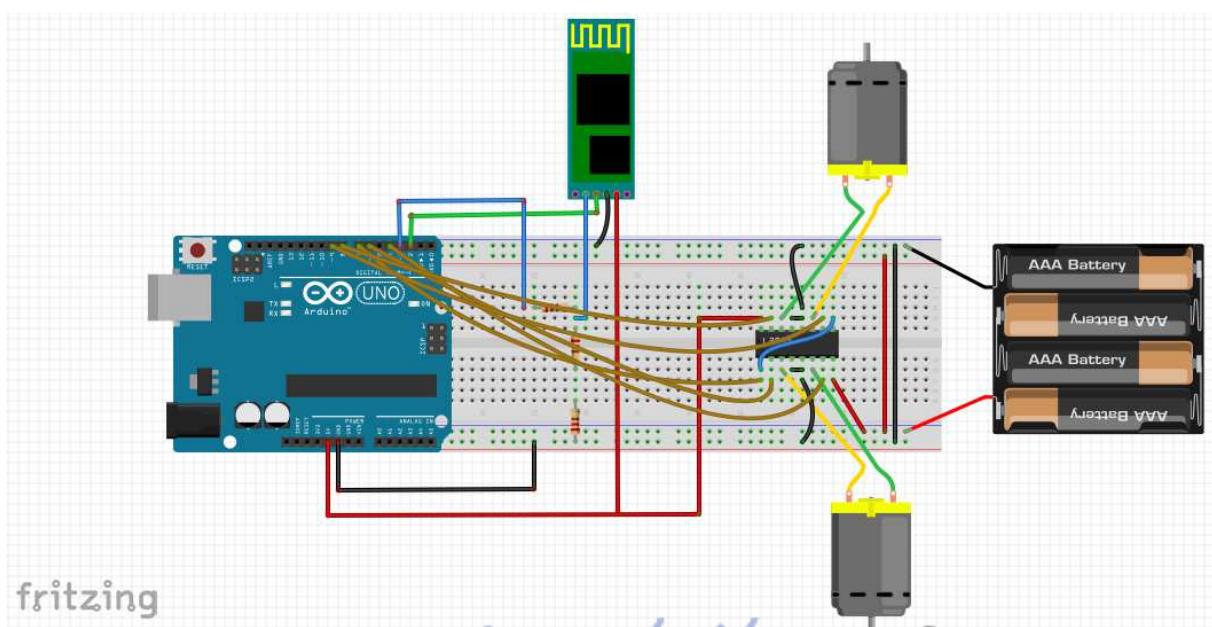


Fig.3.21 Montage réel du bluetooth avec Arduino

6.2. Circuit du RC via Bluetooth :

Pour cela, on a conservé le montage des moteurs avec le circuit intégré L293D tout comme le robot suiveur de ligne. Donc il sera simple d'ajouter le Bluetooth. (Fig.3.22)



Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Le Bleutooth est alimenté avec la source 5v d'Arduino et le deux moteurs par la batterie 12v. Les GROUND du L293D et celui du bluetooth avec le GROUND d'Arduino.

6.3. Programme Arduino :

On a déjà mentionné qu'on a utilisé la bibliothèque « SoftwareSerial.h » pour éviter les problèmes lors du téléversement. On commence par définir les PINs Rx et Tx, la création d'une chaîne de caractère et la définition des PINs responsable au control du circuit L293D :

```
// RC via Bluetooth Code

#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3); // RX | TX

// Creating strings chain
char getstr;

// L293D Control PINs definition:
int en=7;
int m11=6;
int m12=5;
int en2=8;
int m21=9;
int m22=10;
```

On a décidé de déterminer des fonctions qui peuvent décrire le comportement du moteur et les appeler au cours du programme :

```
// Creating moving forward function:
void _forward(int m11, int m12, int m21,int m22)
{
    digitalWrite(m11,HIGH);
    digitalWrite(m12,LOW);
    digitalWrite(m21,HIGH);
    digitalWrite(m22,LOW);
}

// Creating moving backward function:
void _backward(int m11, int m12, int m21,int m22)
{
    digitalWrite(m11,LOW);
    digitalWrite(m12,HIGH);
    digitalWrite(m21,LOW);
    digitalWrite(m22,HIGH);
}

// Creating turning right function:
void _right(int m11, int m12, int m21,int m22)
{
    digitalWrite(m11,HIGH);
    digitalWrite(m12,LOW);
    digitalWrite(m21,LOW);
    digitalWrite(m22,HIGH);
}
```

```

// Creating turning left function:
void _left(int m11, int m12, int m21,int m22)
{
    digitalWrite(m11,LOW);
    digitalWrite(m12,HIGH);
    digitalWrite(m21,HIGH);
    digitalWrite(m22,LOW);
}
// Creating stoping function:
void _Stop(int m11, int m12, int m21,int m22)
{
    digitalWrite(m11,LOW);
    digitalWrite(m12,LOW);
    digitalWrite(m21,LOW);
    digitalWrite(m22,LOW);
}

```

Dans le « void setup » on doit définir le rôle de chaque PIN et le Baud de communication:

```

void setup() {
    BTserial.begin(38400);
    Serial.begin(38400);
    pinMode(en, OUTPUT);
    pinMode(m11,OUTPUT);
    pinMode(m12,OUTPUT);
    pinMode(m21,OUTPUT);
    pinMode(m22,OUTPUT);

}

```

Dans le « void loop » on peut écrire notre programme où il sera exécuté en boucle. Il faut vérifier l'existence des données reçues par Arduino et de les enregistrer dans une chaîne de caractère. On a défini les caractères de commande comme « f » pour que le robot fait un mouvement en avant et « s » pour qu'il s'arrêtera.

```

void loop() {
    if (BTserial.available()>0){
        getstr=BTserial.read(); }

```

```

if(getstr=='f') {
  Serial.println("go forward!");
  _forward(m11,m12,m21,m22);
}
if(getstr=='b')
{
  Serial.println("go backward!");
  _forward(m11,m12,m21,m22);
}
if (getstr=='r') {
  Serial.println("go right!");
  _right(m11,m12,m21,m22);
}
if (getstr=='l') {
  Serial.println("go left!");
  _left(m11,m12,m21,m22);
}

else if(getstr=='s'){
Serial.println("Stop!");
_Stop(m11,m12,m21,m22);
}
}

```

La commande du robot ce fait par un smartphone en utilisant une application du Bluetooth Terminal.

6.4. Montage du RC via Bleutooth : (Fig.3.23)

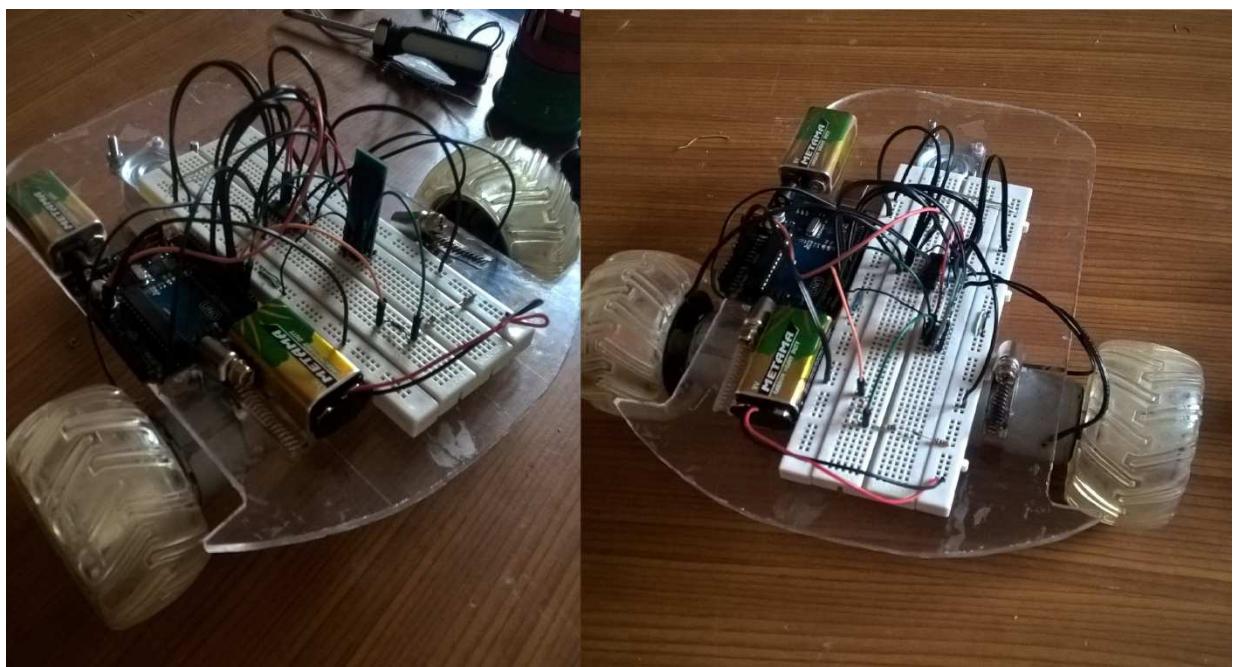


Fig.3.23 Robot RC

7. MODE 03 : Intelligent & Obstacle avoidance :

Dans cette partie on s'intéresse à la réalisation d'un robot qui sera capable de déplacer tout en évitant les obstacles. On a choisi un capteur de distance du type Ulstrasonic HC-SR04.

Ce rebot est constitué d'un capteur Ultrason, une carte Arduino, deux moteurs à courant continu, un circuit L293D, deux batteries de 9V.

7.1. Circuit d'évitement d'obstacle :

Le capteur doit être alimenté par une tension de 5V et son GROUND doit être connecté avec le GROUND d'Arduino. Son ECHO et TRIG sont connectés à des PINs digitales. (Fig.3.24) (Fig.3.25)

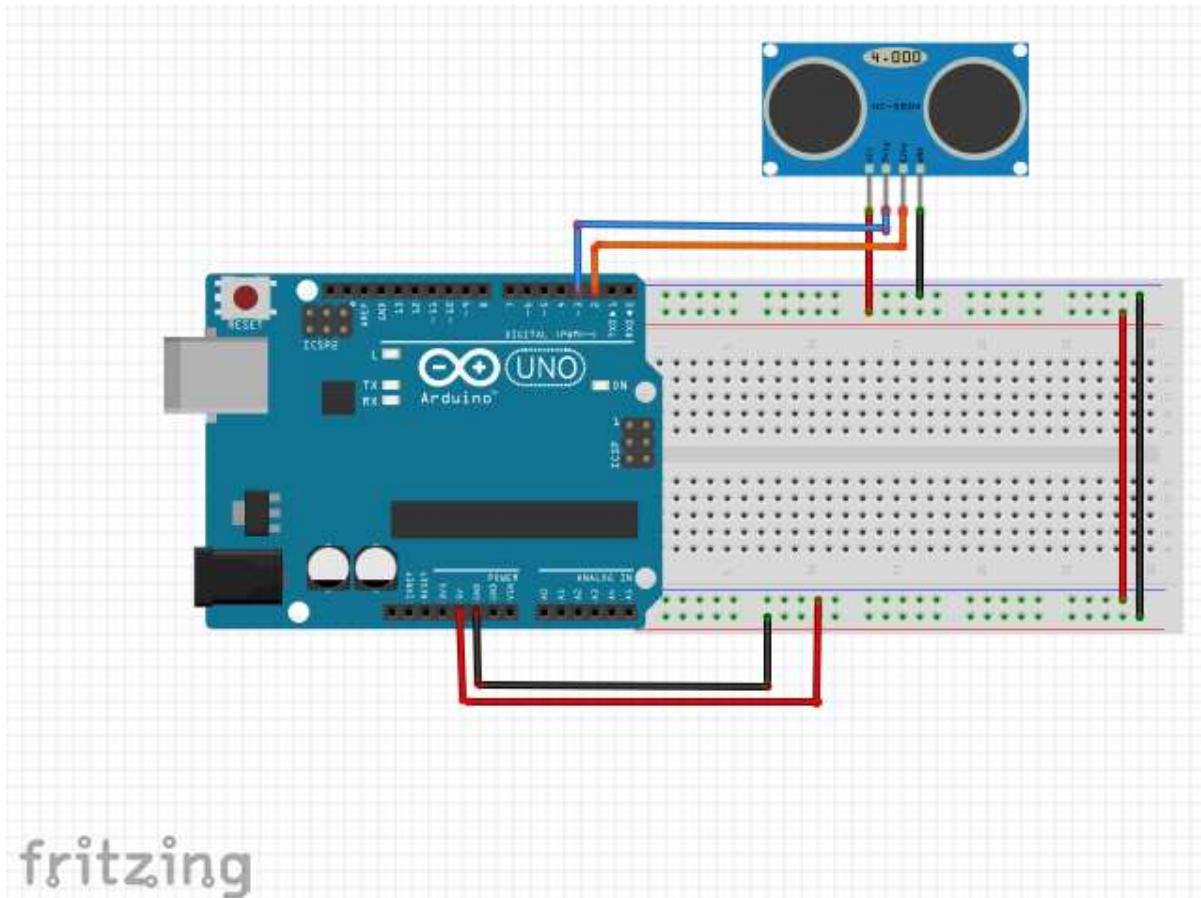


Fig.3.24 Schéma du capteur ultrason avec Arduino

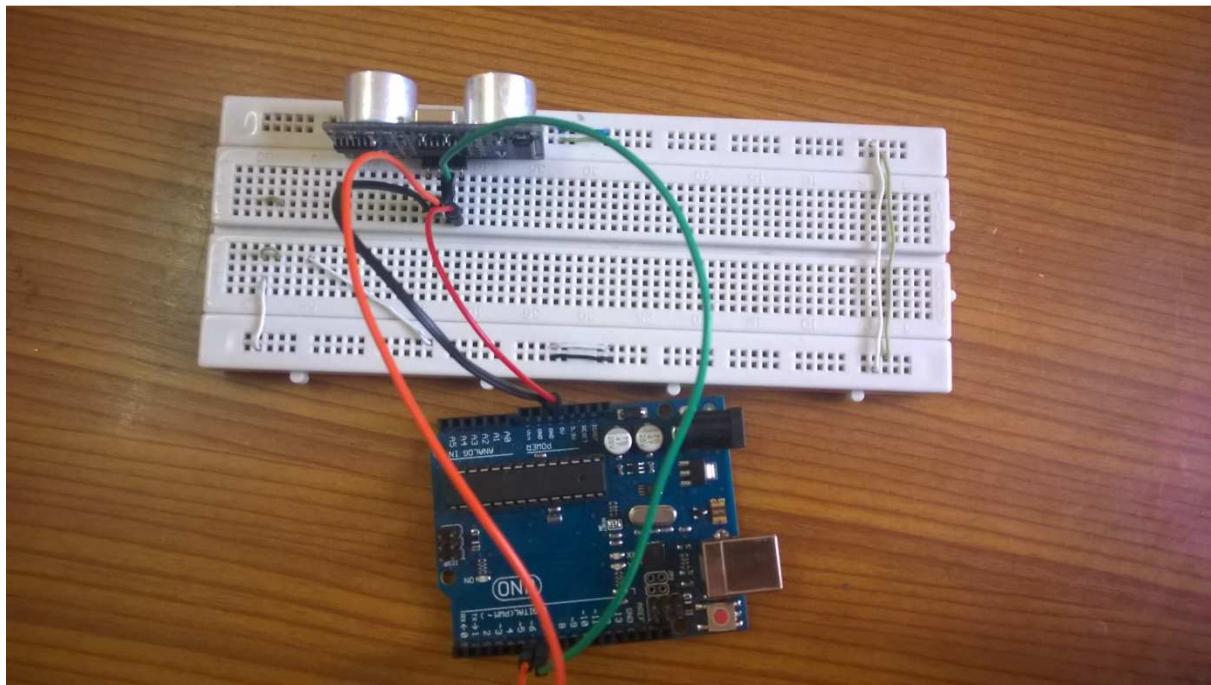


Fig.3.25 Montage réel du capteur ultrason avec Arduino

7.2. Programme du teste :

On commence par définir les PINs connectés aux TRIG et ECHO du capteur ultrason avec ses types dans le « void setup » :

```
// Defining PINs Of Ultrasonic sensor
#define echoPin 2
#define trigPin 3

void setup()
{
    Serial.begin(9600);
    // Ultrasonic PINs Types
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
```

Dans le « void loop » on fait le calcul de la distance par :

```
void loop()
{
    // Calculating Obstacle Distance
    int distance,duration;
    digitalWrite(trigPin, HIGH);
    delay(200);
    digitalWrite(trigPin, LOW);
    duration=pulseIn(echoPin, HIGH);
    distance=(duration/2)/29.1;
    Serial.println(distance);
}
```

La fonction « Serial.println » va nous monter la valeur de la distance dans le moniteur d'Arduino.

7.3. Simulation du capteur ultrason et Arduino sur ISIS :

Pour cette simulation on a utilisé la bibliothèque Arduino et Ultrasonic sensor. On a dû utiliser une résistance variable pour simuler la présence d'un obstacle et un terminal virtuel où son Rx est connecté avec le PIN digital Tx (PIN 1). Après on doit injecter le fichier de l'extension « .hex » du sketch compilé. (Fig.3.26)

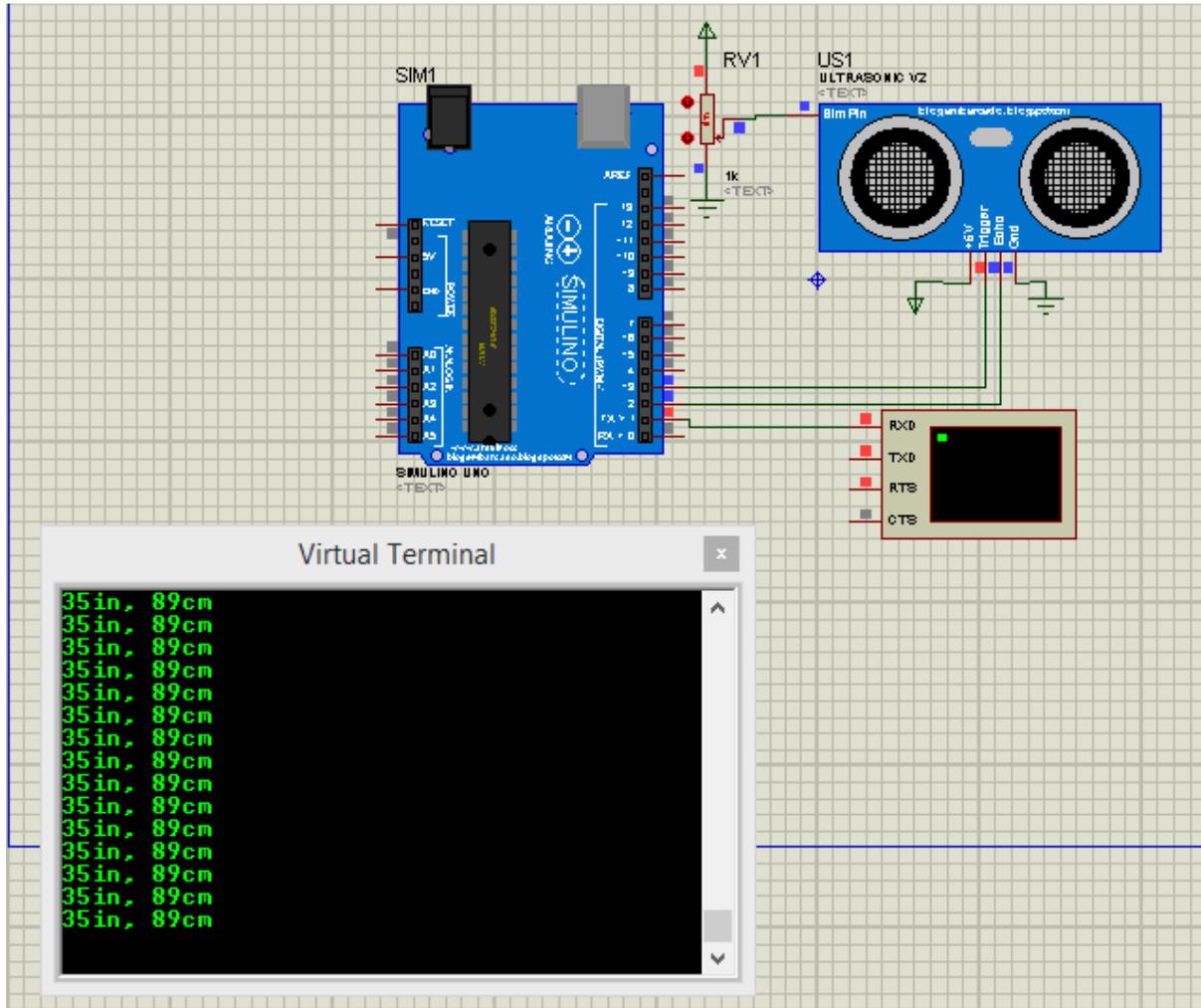


Fig.3.26 Simulation ISIS du capteur ultrason et Arduino

7.4. Circuit du robot intelligent :

Dans ce circuit on doit intégrer le capteur ultrason avec le circuit intégré L293D qui va commander les deux moteurs. (Fig.3.27)

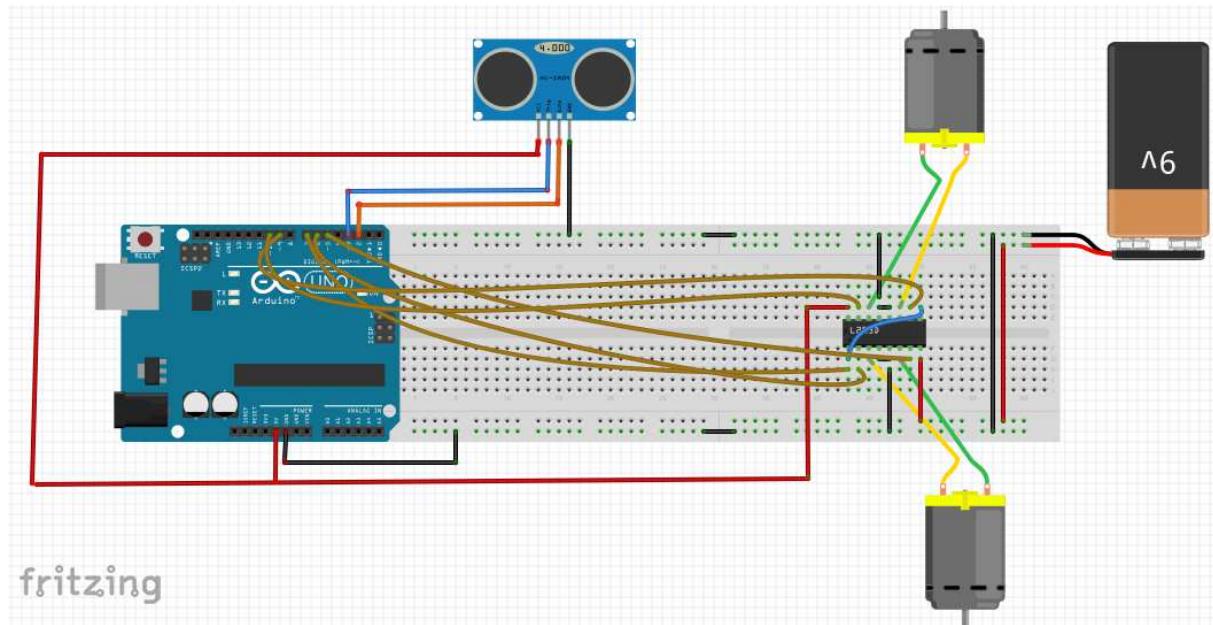


Fig.3.27 Circuit du robot intelligent par Fritzing

7.5. Programme Aduino du robot intelligent :

On fait un simple ajout des éléments responsables au contrôle des moteurs comme les deux modes précédents.

Pour que notre robot soit performant en évitant des obstacles on a décidé qu'il tourne autour de lui-même. C'est-à-dire un moteur commence à tourner dans le sens inverse du l'autre (Dans le cas de simulation dans le sens car leurs couples ont des sens opposés).

On a défini un facteur de décision DF. DF est choisi d'une manière aléatoire avec la fonction « random » dans l'intervalle [0,1] si le robot rencontre un obstacle. Le robot tourne à gauche si $DF < 0.5$ et à droite dans l'autre cas.

```
// Intelligent Robot Code
// Defining PINs Of Ultrasonic sensor
#define echoPin 2
#define trigPin 3

int en=7;           // Enable of L293 to PIN7
int m11=6;          // Left motor forward signal
int m12=5;          // Left motor backward signal
int m21=9;          // Right motor forward signal
int m22=10;         // Right motor backward signal
int DF=0;           // Decision Factor

void setup()
{
  Serial.begin(9600);
  // Ultrasonic PINs Types
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // L293D Controlling PINs Types:
  pinMode(en,OUTPUT);
  pinMode(m11,OUTPUT);
  pinMode(m12,OUTPUT);
  pinMode(m21,OUTPUT);
  pinMode(m22,OUTPUT);
}

void loop()
{
  // Calculating Obstacle Distance
  int distance,duration;
  digitalWrite(trigPin, HIGH);
  delay(200);
  digitalWrite(trigPin, LOW);
  duration=pulseIn(echoPin, HIGH);
  distance=(duration/2)/29.1;
  Serial.println(distance);

  digitalWrite(en, HIGH);

  if(distance >10) // For Going Forward
  {
    digitalWrite(m11, HIGH);
    digitalWrite(m12, LOW);
    digitalWrite(m21, HIGH);
    digitalWrite(m22, LOW);
  }

  else { DF=random(0,2);
  Serial.println(DF);
}
```

```

else { DF=random(0,2);
Serial.println(DF);

if (DF<=0.5) { // For Turning Right
  digitalWrite(m11, HIGH);
  digitalWrite(m12, LOW);
  digitalWrite(m21, LOW);
  digitalWrite(m22, HIGH);
  delay(2000);
}

else { // For Turning Left
  digitalWrite(m11, LOW);
  digitalWrite(m12, HIGH);
  digitalWrite(m21, HIGH);
  digitalWrite(m22, LOW);
  delay(2000);
}
}
}

```

7.6. Simulation du robot intelligent par ISIS :

Après avoir terminé le schéma du système on injecte la carte Adruino par le fichier « .hex » du sketch global et le capteur ultrason par celui qui est censé le commander. (Fig.3.28)

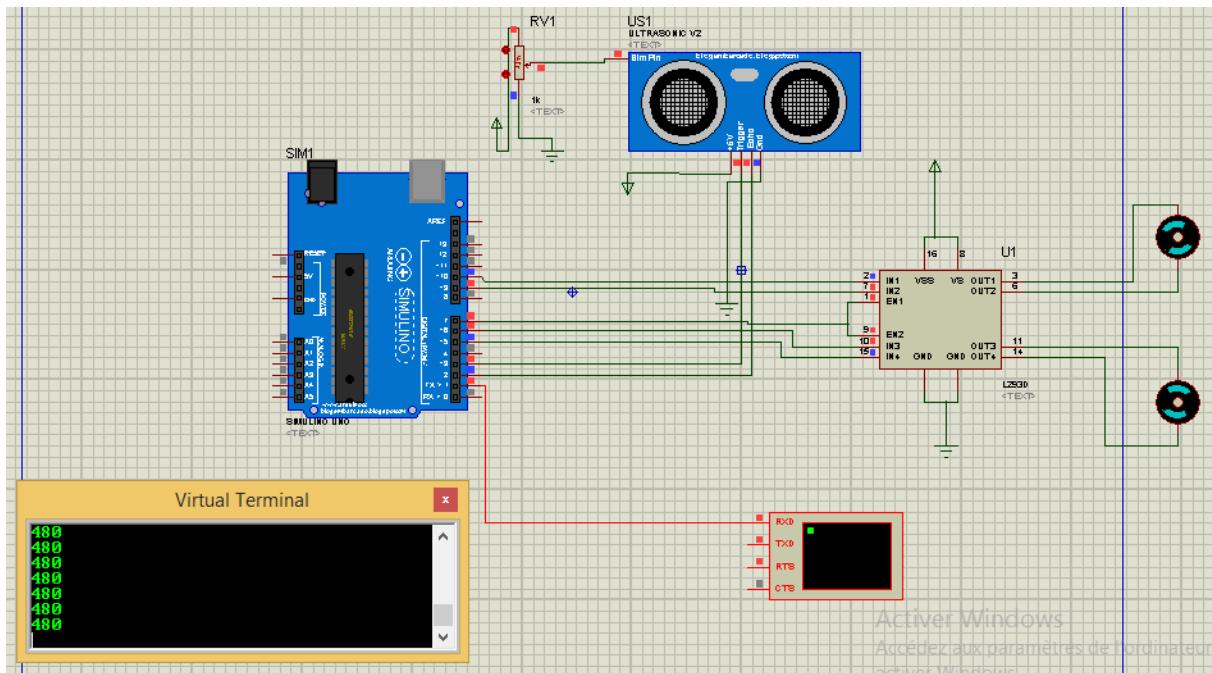


Fig.3.28 Simulation du robot intelligent par ISIS

Pour simuler la présence d'un obstacle on joue sur la résistance variable. On remarque que dans le cas où la distance mesurée est supérieure à 10 les deux moteurs tournent dans le sens opposé qui est censé donner au robot un mouvement tout droit (couples de sens opposés des moteurs installés sous le châssis). Quand le capteur détecte un obstacle proche c'est-à-dire une

distance inférieure ou égale à dix, l'un des moteurs change son sens et les deux commencent à tourner dans le même sens.

7.7. Montage du Robot intelligent : (Fig.3.29)

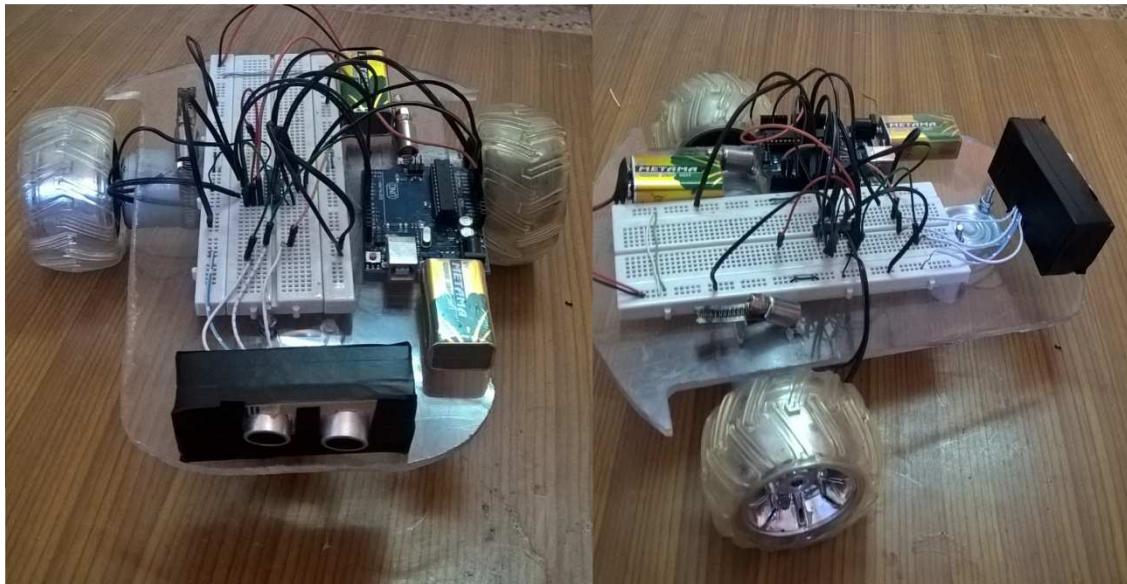


Fig.3.29 Montage final du robot intelligent

8. Conclusion :

Ce chapitre était le noyau de notre projet fin d'étude. On a commencé par la conception des circuits de chaque mode de notre robot, la simulation en 3D du suiveur de ligne par logiciel V-Rep pour voir son comportement et la performance de ses capteurs, les simulations des circuits avec ISIS après l'injection des fichiers « .hex » qui nous avait donné une observation et une confirmation de l'état de ces derniers, en finissant par la réalisation de ces trois modes.

Conclusion générale

Ce projet nous a permis d'élaborer un large spectre de connaissances acquise lors de notre formation comme les approches de l'ingénierie des systèmes en transformant les objectifs du robot et ses exigences en termes de critères de design lors de sa conception pour établir une architecture global qui répond et accomplit la mission du robot. Et comme la mécanique, l'électronique et l'automatique durant l'implémentation des composantes, la réalisation des circuits et la simulation des systèmes du robot.

Le gain majeur de ce travail est d'avoir fait l'étude d'un robot mobile à base différentielle qui est facile à monter et à démonter et peut être utilisé en trois (03) modes.

Le mode suiveur de ligne est généralement utilisé comme un AGV dans les systèmes de production car il est très flexible au changement des routages qui est fait par un simple traçage d'une bande colorée sur terre et engendre un coût minimale.

Pour le mode RC via bluetooth peut être contrôlé à distance. Il peut jouer le rôle d'un robot exploiteur dans un milieu dangereux.

Le dernier mode fait des mouvements en évitant les obstacles. Il peut être amélioré en ajoutant des modules comme « Arduino Digital Magnetic Compass » pour qu'il suive une trajectoire définie et un GPS comme une méthode de localisation absolue va lui donner une autonomie dans ses mouvements.

Références bibliographique

REFERENCES BIBLIOGRAPHIQUE

- [1] www.boutique.afnor.org Mécatronique vocabulaire
- [2] Ben-Zion Sandler – ROBOTICS Designing the Mechanisms for Automated Machinery – Second edition
- [3] Robotique Mobile – David FILLIAT, École Nationale Supérieure de Techniques Avancées ParisTech
- [4] Site Wikipédia – Da vinci medical robot
- [5] Cynthia L Breazeal – Designing Sociable Robots
- [6] www.futura-sciences.com Aéronautique et Drones
- [7] Julien Beaudry – Conception et contrôle d'un robot mobile à vitesses différentielles
- [8] bfrobot.com
- [9] fr.slideshare.net
- [10] arduino.cc/en/Main/ArduinoBoardUno
- [11] engineersgarage.com
- [12] coppeliarobotics.com
- [13] arduino.cc/en/Hacking/BuildProcess

ANNEXES

ANNEXES I

```
// Line Follower Code

// Normal State Definition:
int L=1;
int M=0;
int R=1;

// CNY70 sensors PINs definition:
int Leftsensor=13;
int Middlesensor=12;
int Rightsensor=11;

// L293D Control PINs definition:
int en=7;          // Enable of L293 to PIN7
int m11=6;         // Left motor forward signal
int m12=5;         // Left motor backward signal
int m21=4;         // Right motor forward signal
int m22=3;         // Right motor backward signal

void setup() {

    Serial.begin(9600); // Baud speed
    // Used pins types
    pinMode(en,OUTPUT);
    pinMode(m11,OUTPUT);
    pinMode(m12,OUTPUT);
    pinMode(m21,OUTPUT);
    pinMode(m22,OUTPUT);
    pinMode(Leftsensor, INPUT);
    pinMode(Rightsensor, INPUT);
    pinMode(Middlesensor, INPUT);

}
```

```

void loop() {
    L = digitalRead(Leftsensor);
    M = digitalRead(Middlesensor);
    R = digitalRead(Rightsensor);
    digitalWrite(en, HIGH);

    if ((L==1) && (M==0) && (R==1)) {
        // Normal State : Sensors reading 101
        digitalWrite(m11, HIGH);
        digitalWrite(m12, LOW);
        digitalWrite(m21, HIGH);
        digitalWrite(m22, LOW);
    }
    else if (L==0) {
        // Left turn :
        digitalWrite(m11, LOW);
        digitalWrite(m12, LOW);
        digitalWrite(m21, HIGH);
        digitalWrite(m22, LOW); }
    if (R==0) {
        // Right turn :
        digitalWrite(m11, HIGH);
        digitalWrite(m12, LOW);
        digitalWrite(m21, LOW);
        digitalWrite(m22, LOW); }
}

```

ANNEXES II

```

// RC via Bluetooth Code

#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3); // RX | TX

// Creating strings chain
char getstr;

// L293D Control PINs definition:
int en=7;
int m11=6;
int m12=5;
int en2=8;
int m21=9;
int m22=10;

```

```
// Creating moving forward function:  
void _forward(int m11, int m12, int m21,int m22)  
{  
    digitalWrite(m11,HIGH);  
    digitalWrite(m12,LOW);  
    digitalWrite(m21,HIGH);  
    digitalWrite(m22,LOW);  
}  
// Creating moving backward function:  
void _backward(int m11, int m12, int m21,int m22)  
{  
    digitalWrite(m11,LOW);  
    digitalWrite(m12,HIGH);  
    digitalWrite(m21,LOW);  
    digitalWrite(m22,HIGH);  
}  
  
// Creating turning right function:  
void _right(int m11, int m12, int m21,int m22)  
{  
    digitalWrite(m11,HIGH);  
    digitalWrite(m12,LOW);  
    digitalWrite(m21,LOW);  
    digitalWrite(m22,HIGH);  
}  
  
// Creating turning left function:  
void _left(int m11, int m12, int m21,int m22)  
{  
    digitalWrite(m11,LOW);  
    digitalWrite(m12,HIGH);  
    digitalWrite(m21,HIGH);  
    digitalWrite(m22,LOW);  
}  
// Creating stoping function:  
void _Stop(int m11, int m12, int m21,int m22)  
{  
    digitalWrite(m11,LOW);  
    digitalWrite(m12,LOW);  
    digitalWrite(m21,LOW);  
    digitalWrite(m22,LOW);  
}  
  
void setup() {  
  Serial.begin(38400);  
  pinMode(en,OUTPUT);  
  pinMode(m11,OUTPUT);  
  pinMode(m12,OUTPUT);  
  pinMode(m21,OUTPUT);  
  pinMode(m22,OUTPUT);  
}
```

```
void loop() {
    if (BTserial.available()>0) {
        getstr=Serial.read();
    }
    if(getstr=='f')
    {
        Serial.println(getstr) ;
        Serial.println("go forward!");
        _forward(m11,m12,m21,m22);
    }
    else if(getstr=='b'){
        Serial.println("go back!");
        _backward(m11,m12,m21,m22);
    }
    else if(getstr=='l'){

        Serial.println("go left!");
        _left(m11,m12,m21,m22);
    }
    else if(getstr=='r'){
        Serial.println("go right!");
        _forward(m11,m12,m21,m22);
    }

    else if(getstr=='s'){
        Serial.println("Stop!");
        _Stop(m11,m12,m21,m22);
    }
}
```

ANNEXE III

```

// Intelligent Robot Code
// Defining PINs Of Ultrasonic sensor
#define echoPin 2
#define trigPin 3

int en=7;          // Enable of L293 to PIN7
int m11=6;         // Left motor forward signal
int m12=5;         // Left motor backward signal
int m21=9;         // Right motor forward signal
int m22=10;        // Right motor backward signal
int DF=0;          // Decision Factor

void setup()
{
  Serial.begin(9600);
  // Ultrasonic PINs Types
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // L293D Controlling PINs Types:
  pinMode(en,OUTPUT);
  pinMode(m11,OUTPUT);
  pinMode(m12,OUTPUT);
  pinMode(m21,OUTPUT);
  pinMode(m22,OUTPUT);
}

void loop()
{
  // Calculating Obstacle Distance
  int distance,duration;
  digitalWrite(trigPin, HIGH);
  delay(200);
  digitalWrite(trigPin, LOW);
  duration=pulseIn(echoPin, HIGH);
  distance=(duration/2)/29.1;
  Serial.println(distance);

  digitalWrite(en, HIGH);

  if(distance >10) // For Going Forward
  {
    digitalWrite(m11, HIGH);
    digitalWrite(m12, LOW);
    digitalWrite(m21, HIGH);
    digitalWrite(m22, LOW);
  }

  else { DF=random(0,2);
  Serial.println(DF);
}

```

```
else { DF=random(0,2);
Serial.println(DF);

if (DF<=0.5) { // For Turning Right
  digitalWrite(m11, HIGH);
  digitalWrite(m12, LOW);
  digitalWrite(m21, LOW);
  digitalWrite(m22, HIGH);
  delay(2000);
}

else {          // For Turning Left
  digitalWrite(m11, LOW);
  digitalWrite(m12, HIGH);
  digitalWrite(m21, HIGH);
  digitalWrite(m22, LOW);
  delay(2000);
}
}
```

Résumé

Ce travail consiste à concevoir, réaliser et commander un robot mobile à base différentiel. Des logiciels de simulation et des résultats expérimentaux ont permis de valider notre approche de commande et la performance de notre conception.

Notre objectif était de développer un mode de suiveur de ligne et par la suite, une architecture de pilotage versatile et réactive, permettant à la fois d'intégrer la commande du robot par un Smartphone via Bluetooth, ainsi qu'un algorithme d'évitement d'obstacle réactif.

Mots Clés : Commande, base différentiel, simulation, versatile, réactive, suiveur de ligne, Bluetooth, évitement d'obstacle réactif

Abstract

This work, involves designing, constructing and controlling a differential drive mobile robot. Simulations and experimental results allowed validating our approach of command and our design performance.

Our goal was to develop to line following robot and later on a versatile and reactive control architecture, allowing the integration of a control mode using a Smartphone via Bluetooth and also a reactive obstacle avoidance algorithm.

Key words: Command, differential drive, simulation, versatile, reactive, line follower, Bluetooth, reactive obstacle avoidance.

ملخص

هذا العمل يتضمن تصميم إنشاء و التحكم في روبوت متحرك بقاعدة ذات عجلات تقاضالية. تم القيام بتجارب محاكاة و استخلاص نتائج لتأكيد إستراتيجية القيادة و أداء التصميم .

هدفنا تمثل في تطوير أسلوب تتبع الخط ، إضافة إلى تزويده ببنية متعددة الاستعمالات تمكن من جهة ، إدراج التحكم به بـهاتف ذكي عن طريق بلوتوث و من جهة أخرى إدماج خاصية تحاكي الحواجز.

كلمات مفتاحية: التحكم ، محاكاة ، متعدد الاستخدام ، تفاعلي ، متتبع خط ، بلوتوث ، تحاكي الحواجز.