

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 8 Mai 1945 – Guelma

Faculté des Sciences et de la Technologie

Département d'Electronique et Télécommunications



Mémoire de Fin d'Etudes

pour l'obtention du diplôme de Master Académique

Domaine : Sciences et Technologie

Filière : Electronique

Spécialité : Instrumentation

Robot mobile autonome

Présenté par :

AMIRI Ahmed Faris

ABDOURAHMAN Mahamat Djerou

Sous la direction de :

Dr. TABA Mohamed Tahar

Septembre 2020

Dédicaces

*J'adresse mes remerciements avec un grand respect et gratitude à
mes chers parents que je prie de trouver ici l'expression de ma
reconnaissance*

*et de mes sentiments les plus affectueux en espérant vous satisfaire
avec une réussite permanente.*

*Ainsi qu'à mes frères mes sœurs cousins et cousines
je leurs souhaite tous la réussite dans leurs vies.*

*Ainsi qu'à tous mes amis collègues de la spécialité « Ingénierie
en Instrumentation »*

*Et mon collègue et binôme. Ainsi qu'à tous ceux qui me
connaissent; qui m'ont aidé et qui sont toujours
Présents à mes côtés avec qui j'ai partagé le bon et le mauvais
Merci.*

Abdurahman Mahamat Djerou

Dédicaces

Grace à Dieu le tout puissant, j'ai achevé la réalisation de ce modeste travail que je tien très chaleureusement à le dédier à :

Ma mère chérie et mon père qui m'ont encouragé et soutenu tout au long de mes études et pour leur patience et sacrifice que Dieu les protègent et les garde pour moi.

Mes adorables sœurs à mon grand-père et grand-mère.

Mon collègue et binôme. Ainsi qu'à tous ceux qui me connaissent ; qui m'ont aidé et qui sont toujours

Présents à mes côtés avec qui j'ai partagé le bon et le mauvais

Et à toute la famille A.M.P.F.

Mes chères amis et camarades de loin ou de près pour leur encouragement permanent et leurs soutiens moraux.

Amir Ahmed Faris

Remerciement

Nous remercions ALLAH le tout puissant qui nous a permis de réaliser ce projet et de le présenter sous forme de ce mémoire.

*Nous tenons à exprimer notre profonde reconnaissance à notre encadreur **Dr TABA** Mohamed Tahar qui nous a aidé, poussé et encouragé dans le cadre de la réussite de ce projet et qui a consacré son précieux temps en mettant à notre profit ses inestimables compétences dans un cadre conviviale et humain.*

Un remerciement général à tous les professeurs du département de génie électrique et en particulier à toutes les parties prenantes du département d'Electronique et Télécommunication.

Sans omettre de remercier profondément nos différentes familles pour l'aide financière et morale sans oublier tous les étudiants, frère et amis

Résumé :

Ce travail concerne le montage et la réalisation d'un robot mobile autonome et à quatre roues à l'aide d'une carte électronique "Arduino" et ses différents shields, adaptés pour pouvoir la relier au robot. Après avoir développé le programme en logiciel Arduino, le robot a pour rôle de détecter un obstacle devant lui et de l'éviter en changeant de direction, et suivre la ligne noire à base d'un autre programme.

Mots clés : robot mobile autonome–Arduino–Shields

Abstract :

This work concerns the assembly and realization of an autonomous mobile robot with four wheels using an "Arduino" electronic board and its various shields, adapted to be able to connect it to the robot. After developing the program in Arduino software, the robot's role is to detect an obstacle in front of it and avoid it by changing direction, and follow the black line based on another program.

Keywords: autonomous mobile robot – Arduino – Shields

ملخص :

يتعلق هذا العمل بتجميع وانتاج روبوت متحرك مستقل بأربع عجلات باستخدام لوحة الكترونية " Arduino " و فروعته المختلفة و التي تم تكييفها لتتمكن من توصيلها بالروبوت . بعد تطوير البرنامج في برنامج " Arduino " يكون دور الروبوت هو اكتشاف عقبة امامه وتجنبها عن طريق تغيير الاتجاه ، واتباع الخط الاسود بناء على برنامج آخر.

الكلمات المفتاحية:روبوت متحرك ذاتي - اردوينو- دروع

Sommaire

Remerciement

Dédicace

Résumé

Introduction générale.....	1
Chapitre I : généralité sur les robots.....	2
I.1.Introduction.....	2
I.2. Evolution de la robotique.....	3
I.2.1. Les masques et statues animés de l'antiquité.....	3
I.2.2. L'horloge.....	4
I.2.3. Des automates de l'orient aux jacquemarts européens	5
I.2.4. Vers l'âge d'or des automates.....	5
I.2.5. La première machine programmable.....	6
I.2.6.Invention du premier ordinateur.....	7
I.3. Définition du mot robot.....	8
I.3.1. Les trois lois d'Asimov.....	8
I.3.2. Le robot.....	8
I.3.3. Les composants du Robot.....	9
I.3.3.1. Système de contrôle.....	9
I.3.3.2. Actionneurs.....	10
I.3.3.3. Capteurs.....	11
I.4. Les types de robot.....	11
I.4.1. Les robots manipulateurs.....	12
I.4.1.1. Les types des robots manipulateurs.....	12
I.5. Les robots mobiles	15
I.5.1. Différents types de navigation d'un robot mobile.....	16
I.5.1.1. Stratégies de navigation.....	16
I.5.1.2. Approche d'un objet.....	16
I.5.1.3. Guidage.....	16
I.5.1.4. Action associée à un lieu.....	16
I.5.2. Les architectures de contrôle.....	17
I.5.2.1. Contrôleurs Hiérarchiques.....	17

I.5.2.2. Contrôleurs réactifs.....	18
I.5.2.3. Contrôleurs hybrides.....	18
I.6. Les différents types des robots mobiles.....	19
I.6.1. Robots humanoïdes.....	19
I.6.2. Robots quadrupèdes.....	19
I.6.3. Les robots volants.....	20
I.6.4. Les robots à roues.....	20
I.7. Domaine d'utilisation des robots.....	21
I.7.1. Les robots industriels	21
I.7.2. Robots domestiques ou ménagers.....	21
I.7.3. Robots en médecine et chirurgie.....	21
I.8. Avantages et inconvénients des robots.....	23
I.8.1. Les avantages des robots.....	23
I.8.2. Les inconvénients des robots.....	24
I.9. Conclusion.....	25
Chapitre II : Réalisation du projet.....	26
II.1. Introduction	26
II.2. Microcontrôleur et ArduinoMega 2560.....	26
II.2.1. Microcontrôleur.....	26
II.2.2. ArduinoMega 2560	26
II.3. Caractéristiques et fonctionnement ArduinoMega 2560	27
II.3.1. Les caractéristiques.....	27
II.3.2. Fonctionnement.....	27
II.4. La programmation.....	28
II.4.1. Le langage de l'Arduino.....	28
II.4.2. Le programme.....	30
II.5. Les capteurs.....	31
II.5.1. Définition d'un capteur.....	31
II.5.2. Le Capteur à Ultrason HC-SR04.....	32
II.5.2.1. Les caractéristiques et principe de fonctionnement du capteur Ultrason HC-SR04.....	33
II.5.2.1.1. Les caractéristiques de capteur HC-SR04.....	33
II.5.2.1.2. Le principe de fonctionnement du capteur Ultrason HC-SR04..	33

II.5.2.2. Le montage du capteur sur Arduino.....	34
II.5.2.3. Le matériel utilisé pour tester notre capteur à ultraso.....	35
II.5.2.4. Etalonnage du capteur à ultrason.....	35
II.5.2.5. Interprétation des courbes mesure et erreurs.....	37
II.5.2.6. Avantage et Inconvénient.....	38
II.5.2.6.1. Avantage.....	38
II.5.2.6.2. Inconvénient.....	38
II.5.3. Capteur à infrarouge.....	38
II.5.3.1. Caractéristique principe de fonctionnement.....	39
II.5.3.1.1. Caractéristique.....	39
II.5.3.1.2.Principe de fonctionnement.....	39
II.5.3.2.Câblage.....	39
II.6. Les actionneurs.....	40
II.6.1. Les différents types d'actionneurs.....	40
II.6.1.1. Actionneur pneumatique.....	40
II.6.1.2. Actionneur hydraulique.....	40
II.6.1.3. Actionneur piézoélectrique.....	40
II.6.1.4. Actionneur électromécanique.....	41
II.6.2. Mode de fonctionnement des actionneurs.....	41
II.6.2.1. Mode tout ou rien.....	41
II.6.2.3. Mode réglé en vitesse.....	41
II.6.2.4. Mode servomoteur-réglé en position.....	42
II.6.2.5. Mode pas à pas.....	42
II.6.3. Les actionneurs électriques.....	42
II.6.3.1. Les moteurs à courant continu à aimant permanent.....	42
II.6.3.1.1. Le principe de fonctionnement.....	42
II.7. Les actionneurs du robot (Moteur DC, Servo-moteur et Relais L298N).....	43
II.7.1. Moteur à courant continu.....	43
II.7.1.1. Les caractéristiques du moteur.....	44
II.7.1.2. Avantages et inconvénients du moteur CC.....	45
II.7.2. Servo-moteur.....	46
II.7.2.1. Les caractéristiques du servo-moteur.....	47
II.7.2.2. Le principe de fonctionnement du servo-moteur.....	48

II.7.3. Le Relais L298N.....	49
II.7.3.1. Caractéristiques du relais L298N.....	51
II.7.3.2. Principe du fonctionnement du relais L298N.....	51
II.8. Alimentation du robot.....	53
II.8.1. Batterie.....	54
II.9. La conception du robot.....	54
II.10. Les étapes du montage mécanique.....	55
II.10.1. Châssis 4WD.....	55
II.10.2. La partie inférieure du robot.....	56
II.10.3. La partie supérieure du robot.....	57
II.10.4. Rassemblement des deux parties du robot.....	58
II.11. Câblage des toutes composantes du robot.....	58
II.11.1. Principe de fonctionnement du robot.....	59
II.11.2. Caractéristiques et performance du montage	60
II.12. Perspectives d'évolution et d'amélioration du montage.....	61
II.13. Conclusion.....	63
Conclusion générale.....	64

Sources et bibliographies

Annexe

Liste des tableaux :

Numéro du tableau	Titre du tableau	Page du tableau
II.1	Fonctionnement des icônes de la barre de boutons de l'environnement de développement	30
II.2	Domaine d'utilisation des quelques capteurs	32
II.3	Résultats des mesures sur une plage limitée de 1 à 50 centimètre	36
II.4	Tension de fonctionnement des composantes du robot	54

Liste des figures :

Numéro de la figure	Nom de la figure	Page
Chapitre I		
I.1	Poupées articulées	3
I.2	L'horloge de Ctésibios	4
I.3	L'automate d'Al Jazari	5
I.4	Le canard digérateur de Jacques Vaucanson	6
I.5	Machine à tisser	7
I.6	<u>ENIAC</u> Le premier ordinateur	7
I.7	Le logiciel qui commande le robot	10
I.8	Différents actionneurs d'un robot	10
I.9	Différents capteurs d'un robot	11
I.10	Un bras manipulateur	12
I.11	L'espace de travail de robot cylindrique	13
I.12	L'espace de travail de robot rectiligne	13
I.13	L'espace de travail de robot sphérique	14
I.14	L'espace de travail de robot articulé	14
I.15	Le robot SCARA	15
I.16	Exemple des robots mobiles	15
I.17	Exemple sur l'approche d'un objet	16
I.18	Illustration des architectures classiques des contrôleurs pour les robots mobiles : Hiérarchique (A), Réactive (B), et Hybride (C).	17
I.19	Robot humanoïde ATLAS	19
I.20	Robot quadrupède Big Dog	20
I.21	robot volant (drone)	21

I.22	Robot mobile à roues	21
I.23	Robot chirurgical Da Vinci	22
I.24	Robot infirmier RIBA	23
I.25	Robot patient Hanako Showa	23
Chapitre II		
II.1	Carte ArduinoMega 2560	27
II.2	Rôle du compilateur	28
II.3	Sélection du type de carte Arduino à utiliser	29
II.4	EDI Arduino et structure du programme	31
II.5	chaîne d'action d'un capteur	31
II.6	Capteur à ultrason vue de face et dos	33
II.7	Fonctionnement du capteur à Ultrason	34
II.8	câblage du capteur Ultrason avec Arduino	35
II.9	Les mesures effectuées par le capteur à ultrason	35
II.10	Graphique d'erreurs du capteur ultrason	37
II.11	Déviations de son d'un capteur à Ultrason	38
II.12	vue de face et de dos du capteur infrarouge TCRT5000	38
II.13	Règle de la main droite	42
II.14	Model électrique simple du moteur à courant continu	43
II.15	Moteur CC avec son réducteur + roue	44
II.16	Structure interne du moteur à courant continu	46
II.17	servomoteur MG90S	46
II.18	la structure interne du servo-moteur	47
II.19	connexion d'un servo-moteur à un Arduino	48
II.20	Chronogramme de commande d'un servo-moteur	48
II.21	programme pour faire varier l'angle avec un servo-moteur	49

II.22	Relais L298N	50
II.23	schéma du circuit de commande des moteurs	50
II.24	connexion du relais avec Arduino, les moteurs et l'alimentation	51
II.25	A et B programme de contrôle des moteurs	53
II.26	schéma fonctionnel du robot	55
II.27	châssis du robot et ses différentes pièces qui le composent	55
II.28	Soudage des moteurs et les placer sur la partie inférieure du châssis	56
II.29	Placé la partie supérieure du châssis et faire monter les roues	56
II.30	montage des capteurs infrarouge sur le châssis du robot	56
II.31	Placement du relais L298N sur le châssis et le câblage de quatre moteurs avec le relais	57
II.32	montage du capteur à ultrason, support des batteries sur le châssis	57
II.33	assemblage de la partie inférieure et supérieure du robot	58
II.34	Schéma structurel du robot	59
II.35	Le montage complet du robot	61

Liste des acronymes :

Shield : Carte d'interface pour Arduino

PC : Personal computer

EDI : environnement de Développement Intégré

Moteur CC : moteur à courant continu

PWM : Pulse Width Modulation (modulation de largeur d'impulsion)

GND : ground (masse)

cm :centimetre

% :Poucentage

USB: Universal Serial Bus (bus universelensérie)

V : volt

4WD : Four-Wheel Drive

TTL: Transistor-Transistor Logic

UART: Universal Asymchronous Receiver Transmitte(emetteur –
recepteurasynchroneuniversel)

ISO: Organisation international de normalisation

R U P : Rossum's Universal Robots

RPM : Revolutions Per Minute (Le nombre de tours par minute)

Introduction générale

Introduction générale

Introduction générale :

Le monde dans le quel nous vivons actuellement hérite des aspects des civilisations les plus anciennes ; c'est celle qui dépend de plus en plus de l'usage des machines : il s'agit avant tout d'une civilisation industrielle. La présence de machines et de technologies est devenue indispensable dans le monde d'aujourd'hui et en plus devenu tellement familières qu'on les considère comme omniscience et qu'on y prend plus garde.

On peut prétendre que toutes les autres sciences on existé pour permettre à la robotique de se développer afin d'aider l'Homme à crée des alliages mécanique, électronique et informatique accomplissant automatiquement soit des taches qui sont généralement dangereuses, pénibles, répétitives ou impossibles pour les humains, soit des tache plus simples mais en les réalisant mieux que ce que ferait un être humain.

En rendant de grands services aux industries et aux laboratoires, Les manipulateurs sériels ou parallèles rendent de grands services mais ils sont limités par une contrainte intrinsèque ; leur base est fixe dans un espace de travail bien précis. Dans le cas d'un robot mobile on cherche plutôt la position de travail optimale, cette mobilité a ouvert de nouveaux horizons d'application de la robotique, comme ces robots : ménagers, d'exploration ou exécutant des taches interactives. Si leur avantage est frappant ce n'est pas pour autant courant dans l'industrie, car leur développement demande de relever de nombreux défis, requièrent eux-mêmes des compétences variées d'ingénierie. Avec le temps il y a eu d'énorme progrès sur les robots mobiles qui se sont vite succédé avec des performances plus évoluées en matière de perception, d'autonomie et de pouvoir décisionnel.

Le principal objectif de notre travail consiste à réaliser un robot mobile autonome à l'aide d'une carte Arduino et ses différents shields, et qui a pour tache de suivre une ligne noire et de détecter un obstacle et de l'éviter.

Ce travail est composé de deux chapitres :

Le premier chapitre se penche sur une description générale sur la robotique et certains différents types de robots fixes et mobile

Dans le deuxième chapitre nous présentons les étapes de réalisation de notre robot mobile et les différents composants du robot à savoir les capteurs, les actionneurs et la partie commande.

Nous terminons notre travaille par une conclusion et des perspectives.

Chapitre I :

Généralité sur

les robots

I.1. Introduction :

Depuis des années, un effort particulier a été fait dans les domaines de la recherche et de l'industrie pour construire des robots ; qui pouvant évoluer avec un peu d'intervention humain. Ces robots ont un impact considérable sur la vie moderne d'aujourd'hui : fabrication industrielle, le transport et l'exploration de l'espace. La première génération des robots sont des machines capables d'évoluer dans des environnements parfaitement connus qui a été planifiés à partir d'une modélisation complète de l'environnement ou ils se contentent de suivre une trajectoire. L'humanité a rêvé depuis longtemps de créer des machines qui sont qualifiées et intelligentes et c'est ce qui est entraine de devenir une réalité dans le monde d'aujourd'hui.

Quand les chercheurs ont commencé à réfléchir sur la conception de robots, on appelait « robotique » la science des robots et/ou l'art de concevoir et fabriquer des robots. Les succès des robots industriels, qui travaillaient initialement en poste isolé et qui ont été intégrés à des chaînes de production comme des machines parmi d'autres, ont conduit le public à élargir le sens du terme « robotique » et à le rendre désormais presque synonyme d'automatisation. Le correspondant de robotique devrait naturellement être « automatique » mais ce n'est pas le cas dans le langage courant. Pour compenser ce décalage se sont créés le mot « robotisation » et le verbe « robotiser » pour désigner la plupart des automatisations et leur mise en œuvre, même en l'absence de véritables robots. Dans certains champs d'applications, des mots calqués sur robotique sont apparus. Il en est ainsi de « domotique » qui désigne tout ce qui concerne l'automatisation dans l'habitat ou « productique » pour tout ce qui a trait aux moyens de production. [1]

La robotique est une science ou l'ensemble de techniques permettant la conception et la construction des machines automatiques ou plus simplement de robots. Sa pratique réunie des savoir-faire techniques et des connaissances scientifiques des domaines de l'électronique, de l'informatique et de la mécanique.[2]

En effet, on constate a posteriori que la structure d'une machine ou d'un système n'a pas besoin d'adopter la forme physique de ce qu'on appelle généralement robot pour que son contrôle adopte les mêmes composants que ces machines, à savoir : des capteurs internes pour la régulation, des capteurs externes pour connaître l'environnement, des moyens d'action motorisés, ainsi qu'une informatique de commande pouvant faire appel à la programmation et aux techniques de l'intelligence artificielle. Tous ces composants et leurs actions peuvent être distribués dans l'espace au lieu d'être rassemblés ou de

provenir d'une seule structure. Par ailleurs, une collaboration de divers ensembles est possible aboutissant donc à un système robotisé.

I.2. Evolution de la robotique :

Durant plusieurs millénaires, les automates ont reposé sur des ressorts, engrenages et autres mécanismes, ce qui ne les a pas empêchés d'atteindre des performances qui force l'admiration. Et puis l'électronique et l'informatique sont venues changer la donne en permettant de stocker de très larges quantités d'informations et de séquences d'actions dans une petite puce. Le robot est ainsi arrivé à une sophistication telle qu'il peut désormais tenter de trouver par lui-même la solution de certains problèmes. [1]

I.2.1. Les masques et statues animés de l'antiquité :

L'idée des automates date de la haute Antiquité. On trouve dès l'Egypte pharaonique, un masque Anubis avec une mâchoire mobile, et bien plus tard au XVIIe siècle au Japon, des poupées articulées. Etymologiquement, automate signifie « se mouvoir par soi-même », si bien qu'il ne peut y avoir d'automates autonomes ! D'ailleurs, il leur faut toujours une énergie extérieure, mécanique (impulsion, ressort) ou énergétique (électricité, induction, combustion, chaleur...). En -380, Archytas de Tarente, ami de Platon, se construisit un pigeon mécanique en bois : Suspendu à l'extrémité d'une barre, il tournait sur lui-même grâce à un jet d'air comprimé.[3]



Figure I.1 : Poupées articulées.[3]

I.2.2. L'horloge :

L'horloge hydraulique antique est une horloge hydraulique qui, par définition, est tributaire de « l'invention des heures » datant du V^e siècle avant notre ère. Précédemment, quelques essais pour mesurer le temps par l'eau ont vu le jour en Chine, en Mésopotamie, en Égypte. Dès l'apparition des heures, dans l'antiquité gréco-romaine, ce sont surtout les Grecs qui ont développé ces horloges, qu'elles soient monumentales ou individualisées. Un dénommé Ctésibios qui habite la ville d'Alexandrie. Ctésibios est parvenu à créer une horloge si précise que son cadran fait exactement un tour par année solaire. On les retrouvera plus tard, à l'époque médiévale, sous une forme complexe en Chine, mais les héritiers de l'Antiquité gréco-romaine seront les peuples de la civilisation arabe. Elles y seront pérennisées, avant d'être introduites dans le monde occidental, où bientôt, l'horloge mécanique les remplacera avantageusement. Cette horloge hydraulique, trouvée dans la vallée du Main en Allemagne, est conservée au musée archéologique de Francfort. D'origine romaine, elle est datée du III^e siècle.[4]

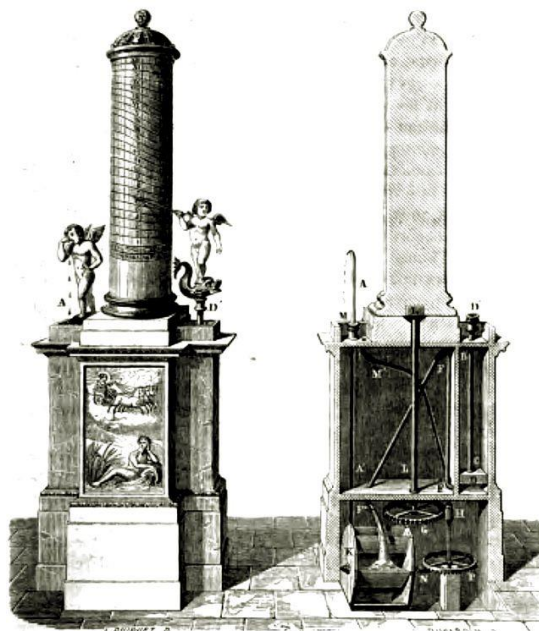


Figure I.2 : L'horloge de Ctésibios [4]

I.2.3. Des automates de l'orient aux jacquemarts européens

Les arabes sont les premiers à mettre en pratique à une grande échelle les techniques décrites par le mathématicien et mécanicien grec Héron d'Alexandrie (et aussi par Phylon de Byzance). Dès 809, Charlemagne reçoit de la part du sultan Haroun Al Rachid un automate mécanique. Puis, lors des huit expéditions en Orient menées à l'occasion des Croisades - de 1096 à 1291 - les européens découvrent de visu l'étonnant raffinement des horloges à eau réalisées par Al Jazari pour le compte de ce même Haroun Al Rachid. Pour obtenir un écoulement constant de l'eau, Al Jazari a développé un système d'une rare ingéniosité, inspiré d'un système inventé par Archimède. La plus grande de ses horloges mesure 3,3 mètres de hauteur et 1,35 mètre de largeur. [1]

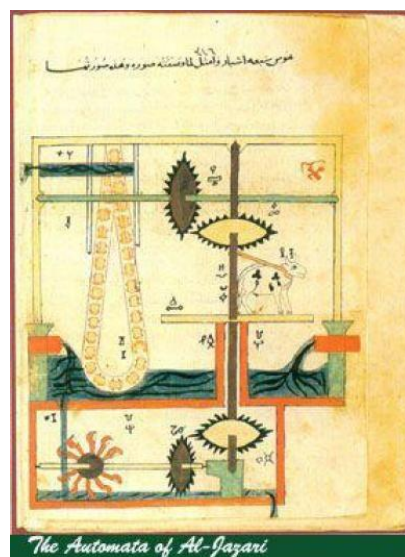


Figure I.3 : L'automate d'Al Jazari

I.2.4. Vers l'âge d'or des automates

Nous entrons dans l'ère de la connaissance, du tout savoir ; époque à laquelle on voit notamment apparaître l'art de l'horlogerie et de la mécanique. Grâce à ces nouvelles techniques vont pouvoir se développer les automates comme objet de curiosité scientifique, capable de reproduire des phénomènes biologiques : c'est notamment le cas du canard digérateur de Jacques Vaucanson. Ce qui est particulièrement ingénieux dans ce canard c'est la façon avec laquelle l'animal avale sa nourriture. Cela se fait grâce à un petit soufflet aménagé tout en bas et qui, par des tuyaux, monte jusqu'au cou. Quand, par ce soufflet, l'air est aspiré dans le cou et que la nourriture remplit le bac, l'air extérieur doit l'y pousser. (...) Là, dans un petit espace, il a fallu construire un petit laboratoire chimique, pour en

décomposer les principales parties intégrantes, et le faire sortir à volonté, par des circonvolutions de tuyaux, à une extrémité de son corps tout opposée. Le Canard digérateur est exposé en 1744 au palais Royal. Il fut malheureusement détruit dans un incendie au milieu du XIX^e siècle. (Figure 1).[5]

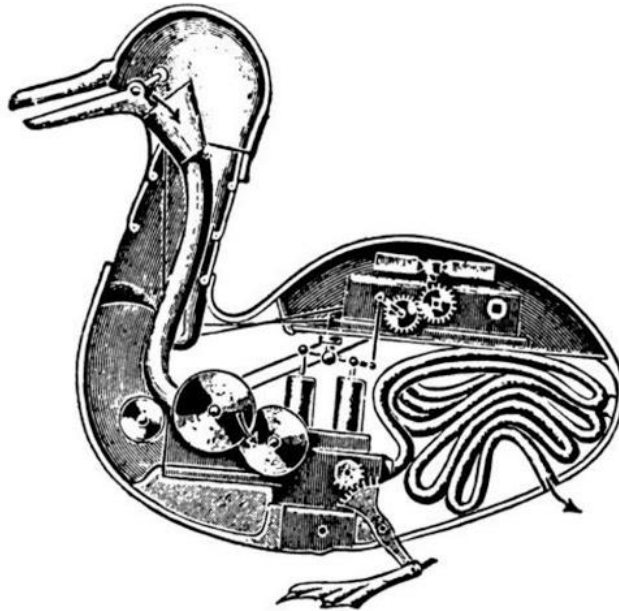


Figure I.4 : Le canard digérateur de Jacques Vaucanson.[5]

I.2.5. La première machine programmable

Au XIX^e siècle, la machine de Vaucanson (un métier à tisser) fut reprise par le tisserand lyonnais Joseph-Marie Jacquard, qui remplaça les cylindres à picots par des chapelets de cartons perforés qui sélectionnaient les aiguilles du métier sur un motif géométrique précis. Mais la résurgence des mythes créateurs porta des préjugés sur l'utilité des automates, perçus comme nuisibles à l'environnement et à l'évolution de l'Humanité. Jacquard en fit la triste expérience, jeté dans le Rhône par ses propres ouvriers qui l'accusaient de sacrifier leurs emplois (déjà) à ces machines qui prenaient une place croissante dans son entreprise.[6]



Figure I.5 : Machine à tisser

I.2.6. Invention du premier ordinateur :

En 1949, les ingénieurs américains John P. Eckert et John W. Mauchly créent le premier ordinateur numérique électronique programmable qu'ils appellent l'ENIAC; acronyme anglais d'Electronic Numerical Integrator Analyser and Computer. L'appareil est inventé durant la seconde guerre mondiale dans le but d'effectuer des calculs qui servent à l'armée américaine. Ou, dit plus simplement, **ENIAC**, le premier ordinateur de l'histoire. L'appareil de 30 tonnes, construit par l'armée américaine au cours des années 1940 afin de calculer la trajectoire des tirs d'artillerie, est un véritable monstre [7]

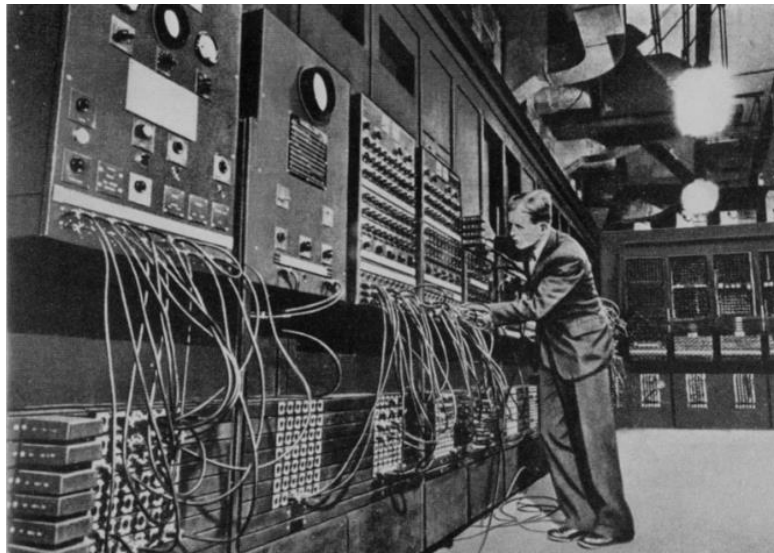


Figure I.6 : ENIAC Le premier ordinateur

I.3. Définition du mot robot :

Un robot est un dispositif mécatronique qui est un assemblage mécanique, électronique et informatique conçu pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines. La conception de ces systèmes est l'objet d'une discipline scientifique, branche de l'automatisme nommé robotique. Le terme robot apparaît pour la première fois dans la pièce de théâtre (science-fiction) R. U. R. (Rossum's Universal Robots), écrite en 1920 par l'auteur Karel Čapek¹. Le mot a été créé par son frère Josef à partir du mot tchèque « robota » qui signifie « travail, besogne, corvée ». Les premiers robots industriels apparaissent, malgré leur coût élevé, au début des années 1970. Ils sont destinés à exécuter certaines tâches répétitives, éprouvantes ou toxiques pour un opérateur humain : peinture ou soudage des carrosseries automobiles. Aujourd'hui, l'évolution de l'électronique et de l'informatique permet de développer des robots plus précis, plus rapides ou avec une meilleure autonomie. Industriels, militaires ou spécialistes chirurgicaux rivalisent d'inventivité pour mettre au point des robots assistants les aidant dans la réalisation de tâches délicates ou dangereuses. Dans le même temps apparaissent des robots à usages domestiques : aspirateur, tondeuses, etc. [8]

I.3.1. Les trois lois d'Asimov : [9]

Les trois lois de la robotique furent le fruit de discussions entre Isaac Asimov et John Campbell autour du thème des robots. Elles furent citées explicitement pour la première fois en 1942 dans la nouvelle Cycle fermé (Runaround). En voici la formulation originale

- Un robot ne peut blesser un humain ni par son action, ni de permettre qu'un humain soit blessé.
- Un robot doit obéir aux ordres donnés par les humains, sauf si de tels ordres se trouvent en contradiction avec la première loi.
- Un robot doit protéger sa propre existence aussi longtemps qu'une telle protection n'est pas en contradiction ni avec la première et/ou ni avec la deuxième loi.

I.3.2. Le robot :

C'est une machine polyvalente pouvant manipuler des objets ou d'exécuter des opérations en réalisant des mouvements variés selon un programme fixe, adaptable ou aisément modifiable. Programmer un robot consiste, dans un premier temps, à lui spécifier la séquence des mouvements qu'il devra réaliser. Certains robots sont dotés de "sens" ; c'est-à-dire d'un ensemble plus ou moins important d'instruments de mesure et d'appréciation (caméra, thermomètre, télémètre, ...) permettant au programme du robot de décider du mouvement le mieux adapté aux conditions extérieures. Par

exemple: si un robot mobile muni d'une caméra peu être amené à se déplacer dans un local inconnu, on peut le programmer pour qu'il contourne tout obstacle qui entraverait sa route. Des robots doter d'un dispositif d'intelligence artificielle afin qu'ils puissent faire face a des situations imprévues et nouvelles. [1]

I.3.3. Les composants du Robot :

I.3.3.1 Système de contrôle tels que :

Contrôleur: le contrôleur est plutôt proche de notre cerveau; même si elle n'a pas la puissance du cerveau; il contrôle toujours les mouvements. Le contrôleur reçoit les données des capteurs et commande les mouvements des actionneurs.

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

Logiciel: trois groupes de logiciels sont utilisés dans un robot. L'un est le système d'exploitation qui exploite le processeur. Le second est le logiciel robotique qui calcule la motion nécessaire de chaque joint du robot basée sur des équations cinématiques. Ces informations sont envoyées au dispositif de commande. Ce logiciel peut être à différents niveaux, de la langue de la machine aux langues sophistiqués utilisés par les robots modernes. Le troisième groupe est la collection d'application - orientée les routines et les programmes développés pour utiliser le robot ou ses périphériques pour des tâches spécifiques telles que l'assemblage, le chargement de machines, la manutention et les routines de vision. [1]

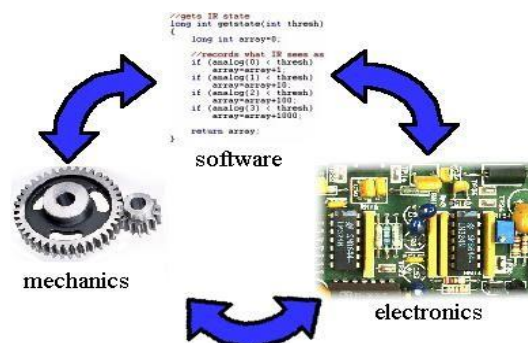


Figure I.7 : Le logiciel qui commande le robot.[1]

I.3.3.2. Actionneurs :

Un actionneur est un élément capable de créer un phénomène physique ou transformer l'énergie qu'il reçoit en chaleur, champs magnétique, lumière, mouvement, position, pression, son etc... Les actionneurs sont les «muscles» de manipulateurs. Le contrôleur envoie des signaux aux actionneurs, qui, à leurs tours déplacent les articulations du robot et des jonctions, les types communs des actionneurs sont les servomoteurs, les moteurs pas à pas et à courant continu, les actionneurs pneumatiques, les vérins hydrauliques et les relais. Les actionneurs sont sous le contrôle du contrôleur.



Figure I.8 : Différents actionneurs d'un robot [10]

I.3.3.3. Capteurs :

Les capteurs sont utilisés pour recueillir des informations sur l'état interne du robot ou pour communiquer avec l'environnement extérieur. Comme chez l'humain, le dispositif de commande de robot doit connaître l'emplacement de chaque lien du robot afin de connaître la configuration du robot. Toujours comme vos principaux sens de la vue, le toucher, l'ouïe, le goût et la parole, les robots sont équipés de dispositifs sensoriels externes comme un système de vision, le toucher et les capteurs tactiles, synthétiseur de parole, et grâce à eux le robot peut communiquer avec le monde extérieur. [1]

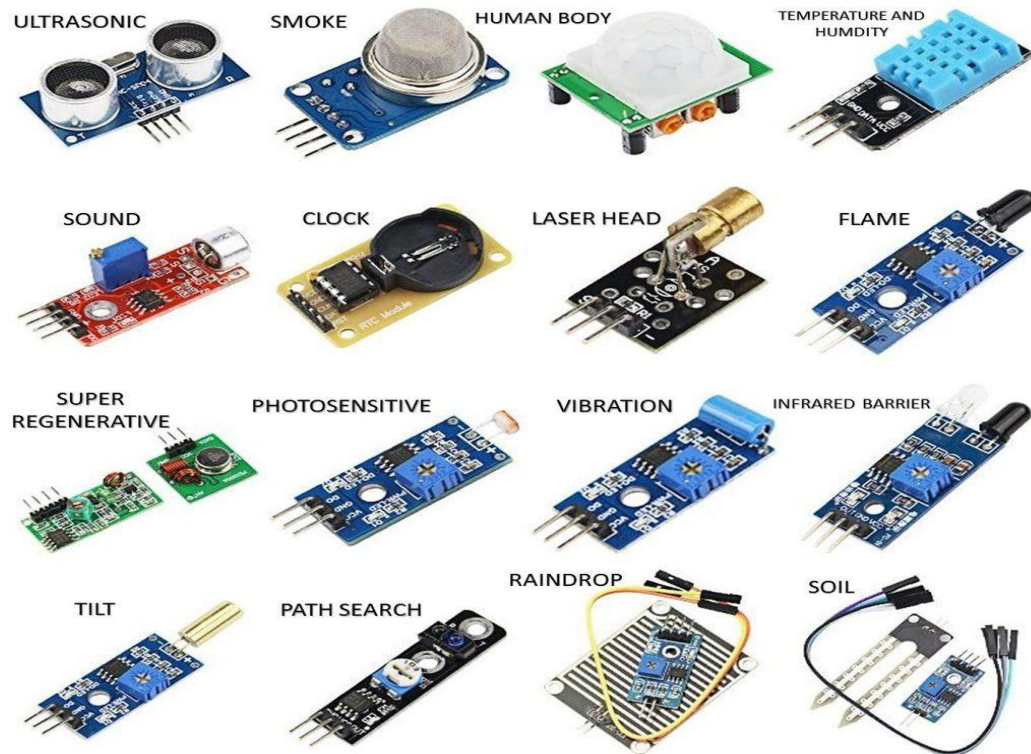


Figure I.9 : Différents capteurs d'un robot.[11]

I.4. Les types de robot : [1]

Il existe deux grandes familles de robots qui sont :

- Les robots manipulateurs.
- Les robots mobiles.

I.4.1. Les robots manipulateurs :

Ce sont des robots plus complexes, tels que les bras manipulateur industriel. Ceci dit le niveau de complexité varie fortement en fonction du nombre de degré de liberté du robot. Un robot manipulateur est en forme d'un bras et se compose d'un certain nombre de segments qui sont conçus pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement en toute sécurité. Les robots manipulateurs sont utilisés dans des applications industrielles pour s'effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage.[1]



Figure I.10 : Un bras manipulateur

I.4.1.1. Les types des robots manipulateurs : [1]

Les robots manipulateurs viennent sous plusieurs formes. Les formes se répartissent en en cinq grandes catégories :

- Robots cylindriques
- Robots rectilignes
- Robots sphériques
- Robots articulés
- Robots SCARA

Robots cylindriques : Le robot cylindrique a deux axes de mouvement, un pour le mouvement en haut et bas. La rotation se fait par la jonction à la base. De plus, le bras horizontal peut se déplacer à l'intérieur et à l'extérieur, ce qui donne un troisième axe de mouvement limité.

robot à structure cylindrique

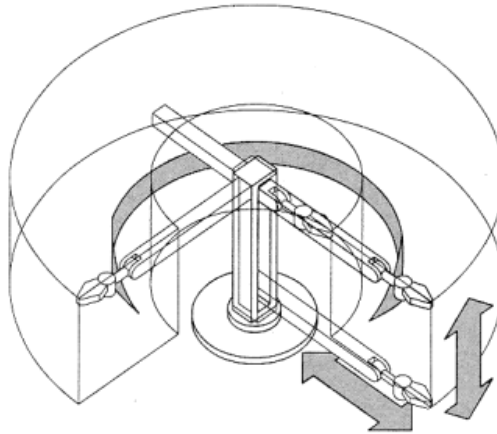


Figure I.11 : L'espace de travail de robot cylindrique [12]

Robots rectilignes : Les robots rectilignes a trois axes de mouvement (x , y , z). Pour cette raison, le robot rectiligne est parfois appelé Robot cartésien. Ces robots sont exploités par vérin pneumatique. [1]

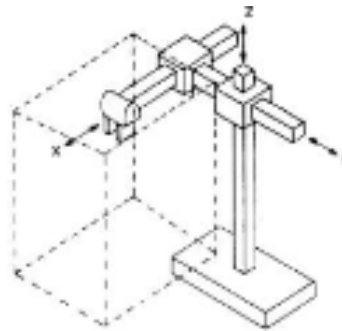


Figure I.12 : L'espace de travail de robot rectiligne

Robots sphériques : Robot manipulateur dont le porteur de robot possède deux articulations rotoïdes consécutives suivies d'une articulation prismatique, les axes de mouvement des deux articulations rotoïdes étant concourants et l'ensemble des articulations permettant d'effectuer des rotations et des translations selon des axes de mouvement définis par un système de coordonnées sphériques, donnant ainsi à l'enveloppe de travail la forme d'une sphère

robot à structure sphérique

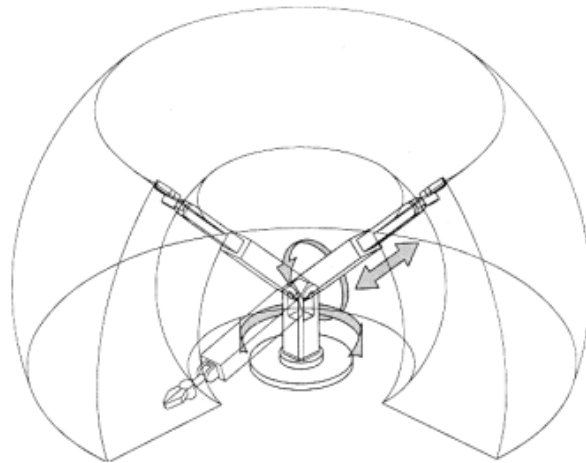


Figure I.13 : L'espace de travaille de robot sphérique [12]

Robots articulés : Un robot articulé est un robot avec des axes rotatifs. Il est alimenté par une variété de moyens, y compris les moteurs électriques. Certains types de robots, comme les bras, peuvent être articulés ou non-articulés. Les robots articulés sont très utilisés dans le domaine de l'industrie. Ils permettent de réaliser du soudage, de l'assemblage et de la peinture.

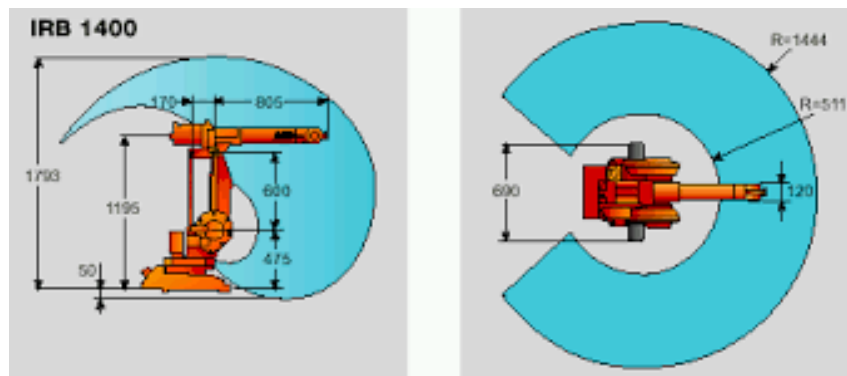


Figure I.14 : L'espace de travaille de robot articulé [13]

Robots SCARA : Un robot SCARA est défini dans la norme ISO 8373: 1994, No.3.15.6, comme un «robot qui comporte deux liaisons pivots parallèles pour fournir conformément à un plan sélectionné».et peut être considérée comme un cas particulier d'un robot cylindrique. Le terme «SCARA» signifie «Selective Compliance Arm for Robot Assembly». (Autre interprétation comprennent «Selective Compliance Articulated for Robot Assembly»). [1]

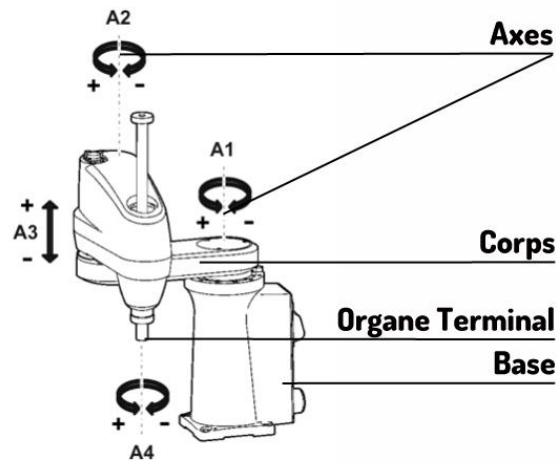


Figure I.15 : Le robot SCARA

I.5. Les robots mobiles :

Ce sont des robots capables de se déplacer dans leur environnement. Celui-ci reste à déterminer soit: aquatique, aérien, sur sol plat, accidenté, sable, pelouse,... Les robots mobiles sur terrain plat en intérieur sont les plus faciles à créer. Il y a un moyen de réaliser de petites machines très simples jusqu'à des robots d'une grande complexité capables de se déplacer sans risque dans des environnements complexes. [2]



Figure I.16 : Exemple des robots mobiles

I.5.1. Différents types de navigation d'un robot mobile :

I.5.1.1. Stratégies de navigation :

Les stratégies de navigation permettant à un robot mobile de se déplacer pour rejoindre un but sont extrêmement diverses, de même que les classifications qui peuvent en être faites. Afin de situer ce type de navigation dans son contexte général, cette classification présente l'avantage de distinguer les stratégies sans modèles internes et les stratégies avec modèle interne. [14]

I.5.1.2. Approche d'un objet :

Cette capacité de base permet de se diriger vers un environnement inconnu depuis une position et de percevoir un objet visible depuis sa courante du robot. 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. [14]

I.5.1.3 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, qui l'entourent ou qui en sont plus proches. [14]

I.5.1.4. Action associée à un lieu :

C'est une capacité qui permet de rejoindre un but depuis des positions pour les quelles ce but qui caractérise son emplacement est invisible. Elle 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. [14]

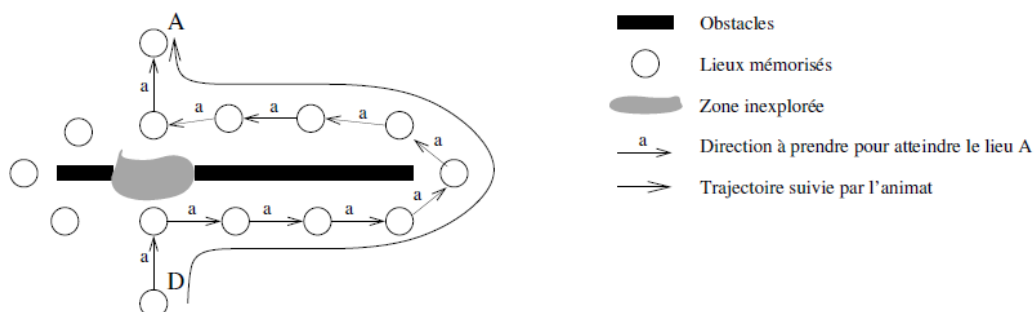


Figure I.17 : Exemple sur l'approche d'un objet. [14]

I.5.2. Les architectures de contrôle :

Un robot est un système complexe qui doit satisfaire à des exigences variées et parfois contradictoires. Un exemple typique pour un robot mobile est l'arbitrage qui doit être fait entre l'exécution la plus précise possible d'un plan préétabli pour atteindre un but et la prise en compte d'éléments imprévus, tels que les obstacles mobiles. Ces arbitrages, que ce soit au niveau du choix de stratégie, ou au niveau de l'utilisation des capteurs, des effecteurs ou des ressources de calcul, sont réglés par un ensemble logiciel appelé architecture de contrôle du robot. Cette architecture permet donc d'organiser les relations entre les trois grandes fonctions que sont la perception, la décision et l'action. [14]

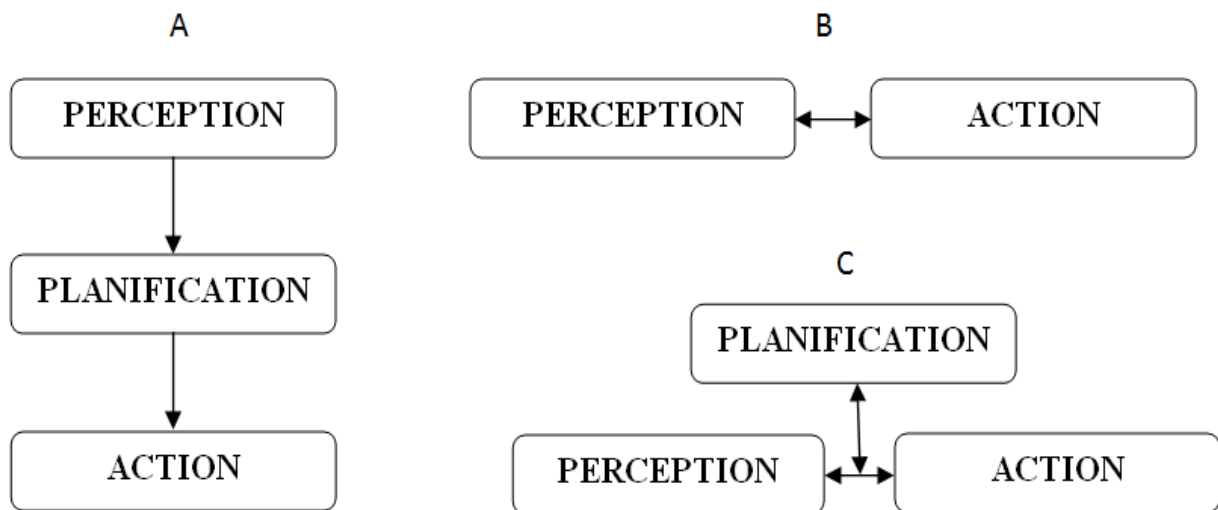


Figure I.18 : Illustration des architectures classiques des contrôleurs pour les robots mobiles : Hiérarchique (A), Réactive (B), et Hybride (C).

I.5.2.1. Contrôleurs Hiérarchiques :

Un robot doit commencer par traiter toutes les données recueillies par ses capteurs concernant son environnement. Ce n'est qu'une fois ce traitement fait qu'il peut identifier les objets qui sont dans son environnement proche. Il lui faut ensuite bâtir une sorte de structure interne des données analysées, pour se représenter la scène dans son ensemble, puis utiliser cette structure pour faire une planification. Il doit calculer après au mieux une séquence de commandes vers les effecteurs pour exécuter le plan

prévu. Dans une telle architecture, on essaie de construire un modèle de l'environnement le plus complet possible et ensuite de raisonner sur la (les) représentation(s) de celui-ci. On privilégie l'aspect cognitif dans le but de reproduire au mieux l'intelligence humaine. [14]

I.5.2.2. Contrôleurs réactifs :

Dans cette architecture, un ensemble de comportements réactifs, fonctionnant en parallèle, contrôle le robot sans utiliser de modèle du monde. Cette architecture supprime évidemment les problèmes dûs aux différences entre la réalité, d'une part, et le modèle de l'environnement du robot, d'autre part, mais limite clairement les tâches que peut effectuer le robot. En effet, sans représentation interne de l'état de l'environnement, il est très difficile de planifier une suite d'actions en fonction d'un but à atteindre. Les robots utilisant cette architecture sont donc en général efficaces pour la tâche précise pour laquelle ils ont été conçus, dans l'environnement pour lequel ils ont été prévus, mais sont souvent difficiles à adapter à une tâche différente.

Les réussites de ces architectures sont liées au couplage direct entre la perception et l'action qui permet une prise en compte très rapide des phénomènes dynamiques de l'environnement. On a donc une bonne robustesse dans des environnements complexes. Ces architectures sont en général basées sur plusieurs comportements : évitement d'obstacles, déplacement aléatoire, déplacement vers un but, fuite d'un point pour guider le robot, il faut donc choisir à chaque instant lequel de ces comportements activer. [14]

I.5.2.3. Contrôleurs hybrides :

La plupart des contrôleurs actuellement utilisés choisissent une solution intermédiaire entre deux approches sous la forme d'une architecture hybride. Cette architecture se compose de deux niveaux. Le premier est chargé des tâches de navigation de haut niveau, telles que la localisation, la cartographie et la planification. Pour cela, il s'appuie sur un second niveau réactif qui est chargé d'exécuter les commandes avec le plus de précision possible et de gérer les éléments non modélisés de l'environnement tels que les obstacles inconnus ou dynamiques. L'action conjointe de ces deux niveaux permet de réagir rapidement face aux variations imprévues de l'environnement, tout en permettant la réalisation d'actions planifiées à plus long terme. Le bas niveau de ces architectures peut être réalisé sous forme de comportements, tels que ceux utilisés dans les architectures réactives. Ces comportements sont des boucles sensorimotrices qui relient les actions aux perceptions avec une phase de décision très courte, qui assure la réactivité. Dans le même temps, les informations sensorielles sont utilisées par le haut niveau dans une boucle sensorimotrice à une échelle de temps beaucoup plus

longue. C'est la mise en parallèle de ces deux échelles de temps qui fait la force de ces architectures.[14]

I.6. Les différents types des robots mobiles :

I.6.1. Robots humanoïdes :

Ce sont des robots qui marchent sur deux jambes et qui ont la morphologie d'un humain. La dernière génération d'un robot humanoïde nommé ATLAS présentée par la société américaine Boston Dynamics. Ce Robot à été fabriqué pour le compte de la DARPA Defense Advanced Research Projects Agency (Agence pour les projets de recherche avancée de défense). Dont l'objectif sera de sauver des vies dans les zones sinistrées mais aussi d'accomplir des missions pour le compte de l'armée. Il dispose de nombreuses particularités dont celle d'être en mesure de marcher sur ses deux jambes même sur un terrain accidenté, son équilibre est extrêmement évolué.



Figure I.20 : Robot humanoïde ATLAS.[15]

I.6.2. Robots quadrupèdes

Quadrupède désigne les espèces d'animaux terrestre qui marchent à quatre pattes et la société américaine Boston Dynamics, spécialisée dans la robotique à usages militaires s'est inspiré des caractéristique de ces animaux pour créer un robot. Ce robot est un robot quadrupède conçu selon la

morphologie du chien en 2005 ; d'où vient elle le son nom, "gros chien" Big Dog. Sa mission principale est d'accompagner les soldats sur des terrains trop irréguliers pour des véhicules et accessibles seulement à pied en transportant à leur place tous leurs matériels tels que : armes, médicament, campement...etc.



Figure I.21: Robot quadrupède Big Dog.[18]

I.6.3. Les robots volants :

Les robots volants peuvent prêter main forte quand il s'agit de réaliser des tâches risquées et répétitives par exemple dans l'industrie, pour inspecter des parois ou des conduites. Ils ont aussi révolutionné le monde du cinéma et de la production vidéo en général, puisque utilisé en tant que drones les robots volants équipés de caméras peuvent capturer des plans à couper le souffle quasiment impossible à réaliser par un caméraman tenant sa caméra. Grâce à la robotique la caméra peut pénétrer dans des endroits lointains difficiles d'accès, d'aller tout en haut puis de redescendre, et surtout en faisant des rotations dans tout les sens possibles, permettant même des prises et des conceptions en 3D dans le montage. [2]



Figure I.22 : Robot volant (drone)

I.6.4. Les robots à roues [1]

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



Figure I.23 : Robot mobile à roues.

I.7. Domaine d'utilisation des robots :

I.7.1. Les robots industriels :

Robots industriels sont des robots utilisés dans un environnement de fabrication industrielle. Ils sont utilisés dans la fabrication des automobiles, des composants et des pièces électroniques, des médicaments et de nombreux produits

I.7.2. Robots domestiques ou ménagers :

Ce type de robots comprend de nombreux appareils très différents, tels que les aspirateurs robotiques, robots nettoyeurs de piscines, balayeuses, nettoyeurs gouttières et autres robots qui peuvent faire différentes tâches. En outre, certains robots de surveillance et de téléprésence pouvaient être considérées comme des robots ménagers peuvent être utilisé dans cet environnement.

I.7.3. Robots en médecine et chirurgie :

La révolution chirurgicale est en route depuis plusieurs années avec comme point de rupture l'avènement de la chirurgie laparoscopique. Les robots semblent avoir de l'avenir à l'hôpital en mai 1998, le robot chirurgical Da Vinci a exécuté le premier pontage coronarien en Allemagne. Près de 20 ans après, il est la référence mondiale. Le système chirurgical Da Vinci : Le robot chirurgien permet d'opérer à distance, soit dans la même pièce avec une machine comme intermédiaire, soit d'un endroit très éloigné, ce qui peut être très utile souvent

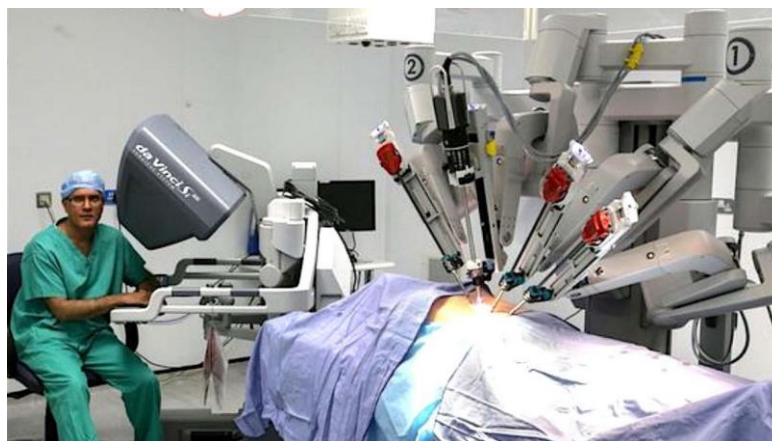


Figure I.24 : Robot chirurgical Da Vinci [19]

Les infirmiers du futur : Les infirmiers qui portent et déplacent les malades seront des robots. En fait, ils ne sont pas prévus pour un avenir si lointain: ils fonctionnent déjà! Le robot infirmier peut prendre un patient dans ses bras, le porter et le déposer dans un fauteuil.

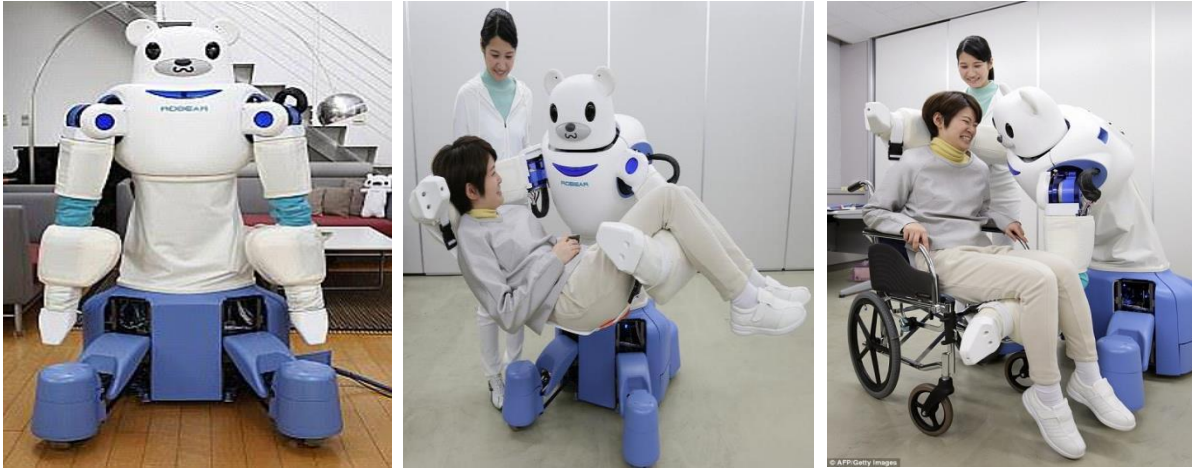


Figure I.25 : Robot infirmier RIBA[20]

Le robot patient : Et le robot patient permet aux futurs chirurgiens dentistes d'apprendre à soigner sans faire de dégâts... L'actroïde Simroid assez réaliste réagit quand l'opérateur le touche à un endroit sensible. Il permet ainsi de réaliser un apprentissage du métier sans frais. Hanako Showa est une initiative similaire. Grâce à des capteurs implémentés dans ses dents artificielle, le robot peut réagir aux actes du praticien novice, émettre des gémissements ou bouger les bras quand on lui "fait mal". Il peut même communiquer grâce à un procédé de synthèse vocal.[1]



Figure I.26: Robot patient Hanako Showa [25]

I.8 Avantages et inconvénients des robots

Un système robotique consiste non seulement des robots mais aussi d'autres dispositifs et systèmes qui sont utilisés avec le robot pour effectuer la tâche nécessaire.

I.8.1. Les avantages des robots sont: [1]

- Accroître la productivité, la sécurité, l'efficacité, la qualité et la cohérence des produits.
- Peuvent travailler dans un environnement dangereux, sans le besoin de soutien de la vie, ou les préoccupations concernant la sécurité.
- N'ont pas besoin de l'éclairage, la climatisation, de ventilation et de protection contre le bruit.
- Travaillent continuellement, sans ressentir une fatigue ou l'ennui, et ne nécessitent pas une assurance médicale ou de vacances.
- Font des tâches de précision répétable à tous les moments, sauf si quelque chose arrive à eux ou ils s'usent.
- Peuvent être beaucoup plus précis que les humains. Précision linéaire d'un robot typiquement est de 20 à 10 microns.

I.8.2. Les inconvénients des robots sont: [1]

L'inconvénient des robots est qu'ils manquent de capacité de réagir en cas d'urgence, à moins que les situations comprises et les réponses sont inclus dans le système. Les mesures de sécurité nécessaires pour s'assurer qu'ils ne lèsent pas les opérateurs et n'endommagent les machines qui travaillent avec eux. Inconvénients des robots comprennent:

- Réponse inadéquate ou mal,
- Le manque de pouvoir prendre une décision,
- Consommation de l'énergie
- Ils peuvent causer des dommages à des autres appareils, et la blessure de l'homme

Bien que les robots ont de bonnes caractéristiques mais aussi ces caractéristiques limités comme la capacité à degré de liberté, capteurs, système de vision et la réponse en temps réel. Les robots sont coûteux en raison du: coût initial de l'équipement, le coût d'installation, le besoin de périphériques, le besoin de formation et la nécessité de la programmation.

I.9. Conclusion :

Nous avons présenté les différentes notions issues d'une vision générale de la robotique et en particulier sur les robots mobiles qui sont essentielles pour la réalisation de notre projet. En suite nous avons défini la manière dont un robot gère ses différents éléments à travers son architecture de contrôle, qui va faire appel la plupart du temps à un modèle interne de l'environnement qui lui permettra de planifier ses actions. Et travers ce chapitre on peut dire que la robotique mobile joue un rôle très importante dans la vie moderne.

Chapitre II: Réalisation du projet

II.1. Introduction :

Notre travail consiste à utiliser la carte Arduino méga qui est la pièce maitresse de notre projet, à fin de concevoir un robot mobile autonome à quatre roues. A travers les capteurs, le microcontrôleur et les actionneurs nous allons réaliser ce robot, et par la suite le programme qui nous permettra de gérer toutes ces applications du robot.

II.2. Microcontrôleur et Arduino Mega 2560 :

Définition : un microcontrôleur est l'équivalent d'un mini ordinateur tel qu'un PC par exemple, correspond au cerveau du robot. C'est lui qui va traiter les informations provenant des capteurs et qui va donner la réponse voulue aux actionneurs.

II.2.1. Microcontrôleur:

Le microcontrôleur est composé de quatre parties, contenu dans un seul boîtier de circuit intégré à plus ou moins grand nombre de pattes. Il contient ainsi une unité centrale, l'équivalent du microprocesseur qui équipe un PC, de la mémoire vive, de la mémoire morte, des interfaces diverses pour communiquer avec le monde extérieur et toute la circuiterie électronique et logique nécessaire pour faire fonctionner tout cela ensemble.

II.2.2. Arduino Mega 2560 :

Définition Arduino : est une carte de circuit imprimé supportant un microcontrôleur et la circuiterie nécessaire pour lui permettre de fonctionner, associé à une interface USB permettant de le programmer. Il y a plusieurs types de carte d'arduino : UNO, ATMEGA, JUNINO ...etc. mais ce qui nous intéresse ce l'Arduino Mega 2560. Nous avons utilisé cette carte puis que nous avons besoin de plusieurs broches de connections et aussi de sa capacité de mémoire interne ce qui n'est pas le cas avec la carte UNO ; et nous aussi possédons cette carte.

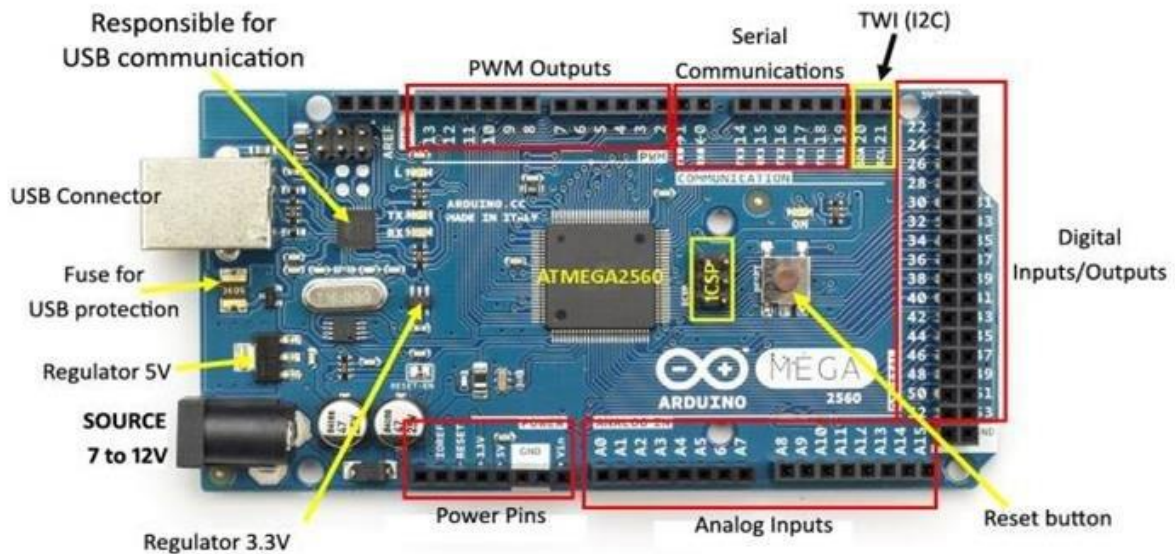


Figure II.1 : Carte Arduino Mega 2560. [16]

II.3. Caractéristiques et fonctionnement Arduino Mega 2560 :

II.3.1. Les caractéristiques :

- Microcontrôleur : ATmega2560
- Tension de fonctionnement : 5 V
- Tension d'entrée (recommandée) : 7 à 12 V
- Tension d'entrée (limites) : 6 à 20 V
- Broches E/S numériques : 54 (dont 14 fournissent la sortie PWM)
- Broches d'entrée analogiques : 16
- courant continu pour la broche de 5 V : 40 mA
- Mémoire Flash : 256 Ko (dont 8 Ko utilisés par le chargeur initial de programme)
- SRAM : 8 Ko
- EEPROM : 4 Ko
- Vitesse de l'horloge : 16 MHz

II.3.2. Fonctionnement :

La carte Arduino Mega 2560 est basée sur le microcontrôleur qui a pris la relève de la carte Mega 1280. Il est doté de 54 broches d'entrée/sortie numériques repérer de 0 à 53 (dont 14 peuvent être utilisées comme sorties PWM), 4 entrées/sorties Rx0 à Rx3 et Tx0 à Tx3, 16

entrées analogiques, de 4 émetteurs-récepteurs universels asynchrones (UART, ports de série de matériel), d'un oscillateur en cristal de 16 MHz, d'une connexion USB, d'une prise de courant, d'une embase ICSP et d'un bouton de réinitialisation. Il contient tout ce qui est nécessaire pour prendre en charge le microcontrôleur.

Chacune des 54 broches numériques sur le Mega peuvent être utilisées en tant qu'entrée ou sortie, en utilisant les fonctions `pinMode()`, `digitalWrite()`, et `digitalRead()` Il dispose également de 16 entrées analogiques, chacune d'elles disposant de 10 bits de résolution (c'est à dire 1 024 valeurs différentes). L'Arduino Mega2560 dispose d'un certain nombre de moyens pour communiquer avec un ordinateur, un autre Arduino, ou d'autres microcontrôleurs. L'ATMEGA2560 fournit quatre UART physiques pour des communications sérielles de type TTL (5 V). [29]

II.4. La programmation :

Un programme est une liste d'instructions qui est exécutée par un système et nous allons utiliser le logiciel Arduino (compilateur) EDI (environnement de Développement Intégrer) qui est une open source pour programmer notre robot. Un compilateur en informatique, ce terme désigne un logiciel qui est capable de traduire un langage informatique, ou plutôt un programme utilisant un langage informatique, vers un langage plus approprié afin que la machine qui va le lire puisse le comprendre. C'est le rôle du traducteur. Le compilateur va donc traduire les instructions du programme précédent, écrites en langage texte, vers un langage dit "machine". Ce langage utilise uniquement des 0 et des 1.

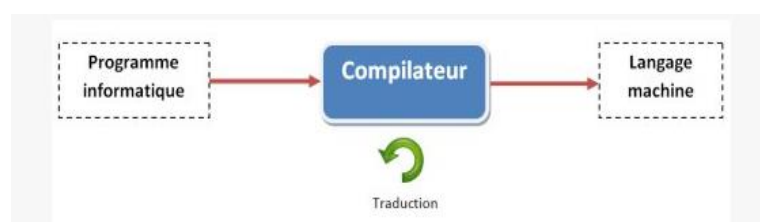


Figure II.2 : Rôle du compilateur

II.4.1. Le langage de l'Arduino :

L'Arduino se programme en un langage qui est un mélange de C et de C++, restreint et adapté aux possibilités de la carte. Aujourd'hui le langage C et C++ sont les plus enseignés

et les plus utilisés en environnement professionnel. Pour cela nous avons installé le logiciel Arduino qui est « open source » c'est-à-dire n'importe qui peut le télécharger sur le site officiel Arduino (arduino.cc), l'utiliser et aussi apporter son contribution s'il le souhaite. Après avoir installé le programme on doit choisir le type de carte Arduino utilisé. La figure ci-dessous illustre bien comment choisir. [29]

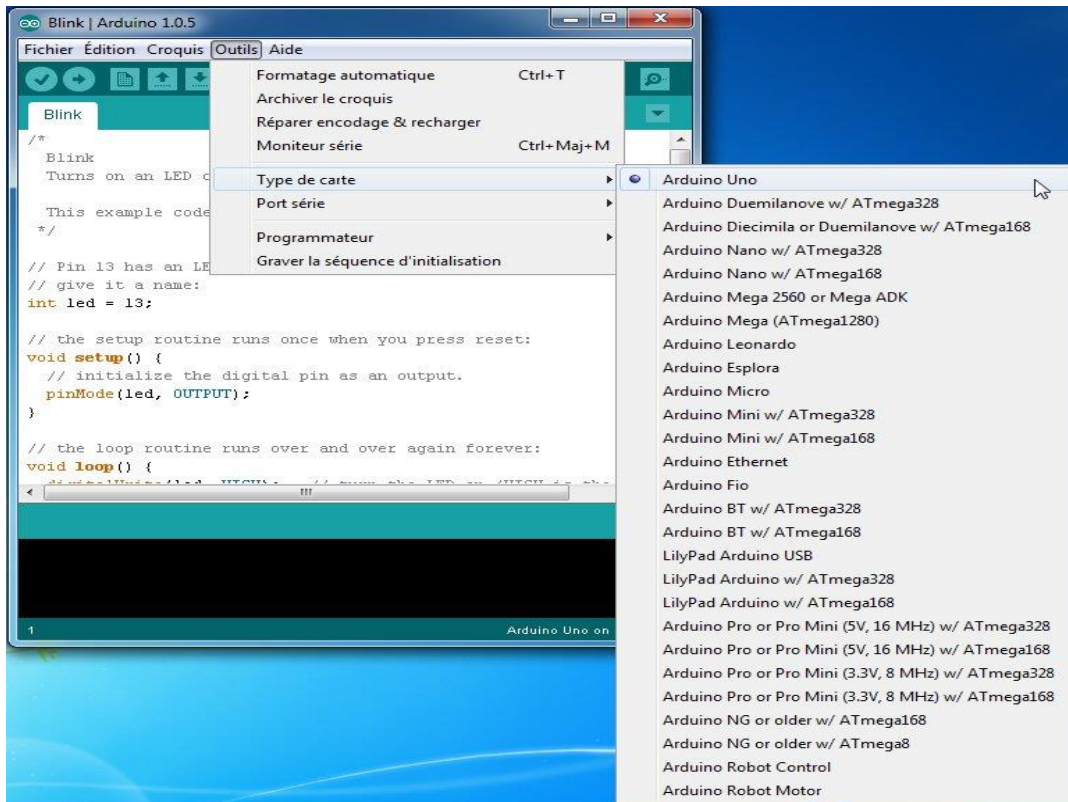








Figure II.3 : Sélection du type de carte Arduino à utiliser.

Pour prendre un peu de base sur ce langage et ses règles de syntaxe nous avons utilisé le Livre [29]. Comment l'utiliser au moyen des exemples de sketches qui y sont fournis. Pour maîtriser le logiciel Arduino on a besoin de connaître certaines icônes et leur utilisation de l'environnement de développement d'après le tableau ci-dessous.

Tableau II.1: Fonctionnement des icônes de la barre de boutons de l'environnement de développement

Bouton	Fonction
	Vérifier : compile le sketch (programme) en cours d'édition et, de ce fait, vérifie sa syntaxe
	Nouveau : ouvre une fenêtre vierge pour écrire un nouveau sketch (programme)
	Ouvrir : ouvre la liste de tous les sketches contenus dans votre sketch book. Le fait de cliquer sur un d'entre eux l'ouvre dans la fenêtre d'édition courante
	Enregistrer : sauvegarde le sketch en cours d'édition
	Téléverser : compile le programme en cours d'édition et le télécharge en mémoire de l'Arduino
	Moniteur série : ouvre l'écran du moniteur de port série intégré à l'environnement

II.4.2. Le programme :

Certaines règles doivent être respectées pour écrire un programme avec le langage Arduino. En effet, l'exécution d'un programme Arduino s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres, le compilateur vérifie l'existence de deux structures obligatoires à tout programme Arduino qui sont:

- La partie initialisation et configuration des entrées/sorties - la fonction `setup ()`
- La partie principale qui s'exécute en boucle - la fonction `loop ()`

Par contre, la partie déclaration des variables est optionnelle. La figure ci-dessous montre l'interface graphique de l'EDI ainsi que la structure d'un programme réalisé avec le langage Arduino.

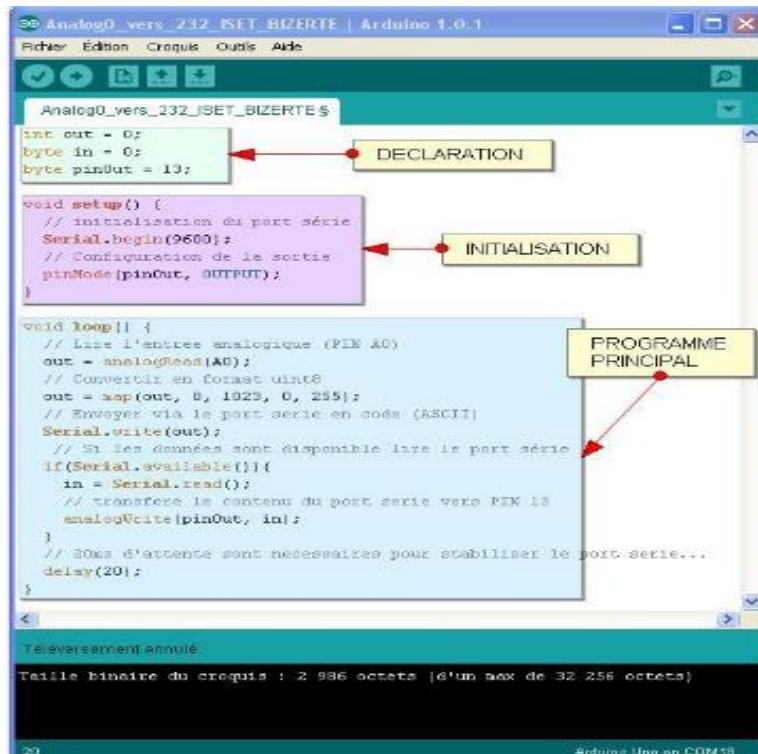


Figure II.4 : EDI Arduino et structure du programme

II.5. Les capteurs :

II.5.1. Définition d'un capteur :

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique (information entrante) une autre grandeur physique de nature différente (la plus part du temps, électrique). Cette grandeur, représentative de la grandeur prélevée, est utilisable à des fins de mesure ou de commande. En quelque sorte le capteur est la partie d'une chaine de mesure qui se trouve au contact direct du mesurande ; la plus part des cas le signale de sortie est électrique en raison de la facilité de transmission de l'information. [17]

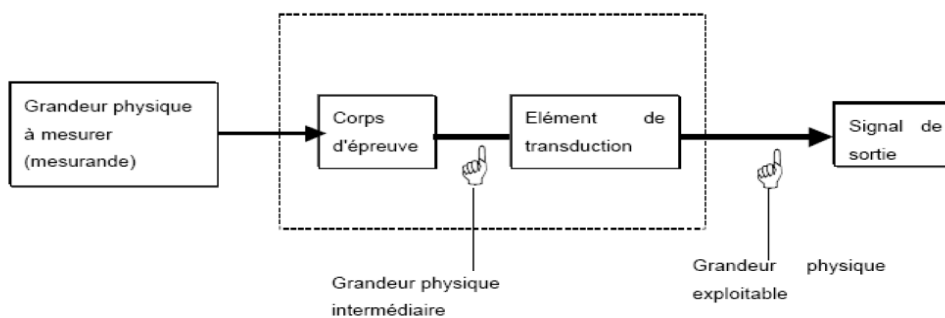


Figure II.5 : chaine d'action d'un capteur.[24]

Tableau II.2 : Domaine d'utilisation des quelques capteurs

Processus	Les capteurs utilisés
Avions	déplacement, vitesse, accélération, l'altitude, la position, force, pression, température, débit de fluide, la tension, le courant, le système de positionnement mondial (GPS)
Automobile	déplacement, vitesse, force, pression, température, débit de fluide, niveau de liquide, la tension, le courant
Système de chauffage domestique	pression, température, débit de fluide
Robot	image optique, déplacement, vitesse, force, la tension, le courant, couple motrice
Four de séchage du bois	La température, l'humidité relative, taux d'humidité et le débit d'air

II.5.2. Le Capteur à Ultrason HC-SR04 :

Un capteur à ultrasons émet à des intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho. La distance étant déterminée par le temps de propagation des ultrasons et non par leur intensité. Pratiquement tous les matériaux qui reflètent le son peuvent être détectés et ce, quelle que soit leur couleur. Même les matériaux transparents ou les feuilles minces ne représentent aucun problème pour un capteur à ultrasons. Les capteurs à ultrasons microsonic sont disponibles pour des portées de 20 mm à 10 m et, du fait même de leur principe, donnent la valeur mesurée au millimètre près. Certains capteurs peuvent même atteindre une précision de 0,025 mm.[30]

Ils y a plusieurs types de capteurs à ultrason : mode fenêtre, mode barrière à réflexion, à sortie analogique etc... mais ce qui nous intéresse ici c'est le capteur à ultrason « le mode réflexion ». Il représente le mode d'utilisation classique. Il exploite le principe de suppression

d'arrière-plan qui est supérieur à tout autre procédé de détection. Ici, la sortie est activée dès que la cible est localisée entre le capteur et le seuil de commutation réglé. Le point de commutation est doté d'une hystérésis réglable. Ce mode de fonctionnement est indiqué, par exemple, pour le comptage de pièces sur un convoyeur ou pour une détection de présence.[30]



Figure II.6 : Capteur à ultrason vue de face et dos.[21]

II.5.2.1. Les caractéristiques et principe de fonctionnement du capteur Ultrason HC-SR04 :

II.5.2.2 Les caractéristiques de capteur HC-SR04 :

- La tension de fonctionnement : 5V
- Le courant de fonctionnement : 15mA
- Fréquence de fonctionnement : 40kHz
- La distance de détection : de 2cm à 450cm
- L'intervalle angulaire de capture : de $-7,5^\circ$ à $+7,5^\circ$
- Signale de détection : impulsion de 10us TTL
- Signale d'écho : sortie TTL

II.5.2.3. Le principe de fonctionnement du capteur Ultrason HC-SR04 :

Le principe de fonctionnement du capteur est entièrement basé sur la vitesse du son.

Voilà comment se déroule une prise de mesure :

- On envoie une impulsion HIGH de 10 μ s sur la broche TRIGGER du capteur.
- Le capteur envoie alors une série de 8 impulsions ultrasoniques à 40KHz (inaudible pour l'être humain, c'est quand plus agréable qu'un biiiiip).

- Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retournent dans l'autre sens vers le capteur.
- Le capteur détecte l'écho et clôture la prise de mesure.

Le signal sur la broche ECHO du capteur reste à HIGH durant les étapes 3 et 4, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance.[31]

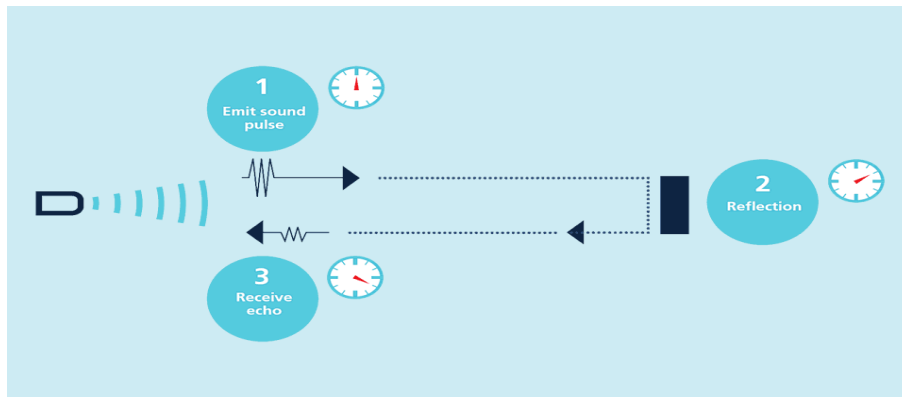


Figure II.7 : Fonctionnement du capteur à Ultrason.

II.5.2.4. Le montage du capteur sur Arduino :

- L'alimentation 5V de la carte Arduino va sur la broche VCC du capteur.
- La broche GND de la carte Arduino va sur la broche GND du capteur.
- La broche D2 mais il n'est pas obligatoire de brancher la broche Trigger du capteur à la broche D2 de l'Arduino, ça dépend du programme. On peut la placer à la broche D4, D4, ou D6 à condition de le déclarer dans le programme de la carte Arduino va sur la broche TRIGGER du capteur.
- La broche D3 de la carte Arduino va sur la broche ECHO du capteur.

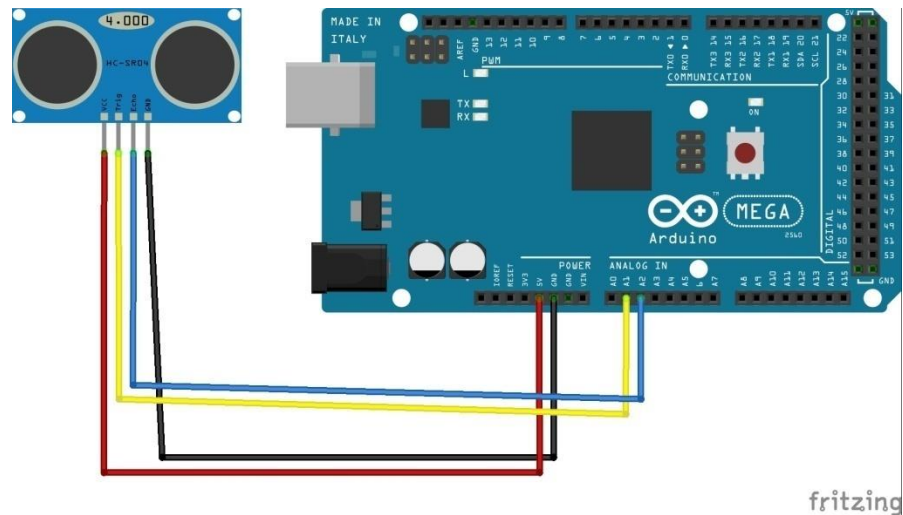


Figure II.8 : câblage du capteur Ultrason avec Arduino

II.5.2.5. Le matériel utilisé pour tester notre capteur à ultrason :

- Une carte Arduino Mega2560
- Un capteur à ultrason HC-SR04
- Une plaque d'essai et des fils pour câbler le montage
- Un câble USB pour connecter l'Arduino à un ordinateur

II.5.2.6. Etalonnage du capteur à ultrason :

Les images ci-dessous montrent comment on a effectué les mesures avec le capteur à ultrason

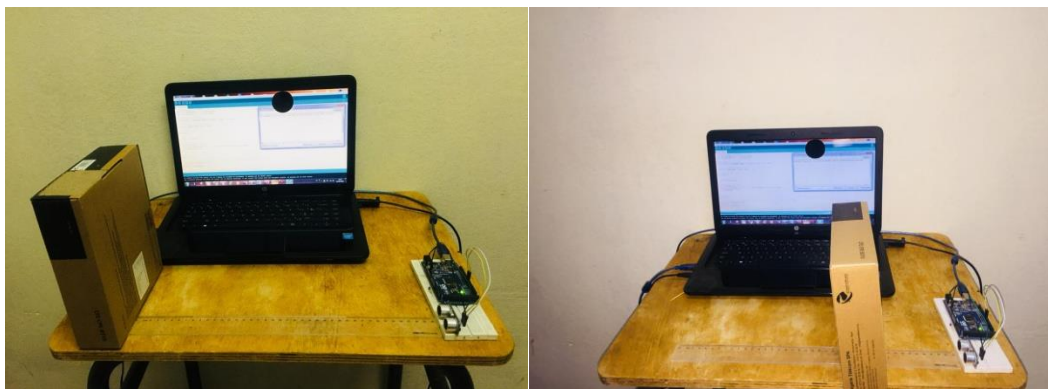


Figure II.9 : Les mesures effectuées par le capteur à ultrason

Le tableau ci-dessous montre les résultats des mesures que nous avons effectuées avec le capteur ultrason HC-SR04.

Tableau II.3 : Résultats des mesures sur une plage limitée de 1 à 50 centimètre.

Distance (cm)	Mesure	Erreur (cm)	Erreur (%)
1	1,18	0,18	18
2	1,868	0,132	6,60
3	2,346	0,65	21,80
4	3,264	0,73	18,5
5	4,25	0,75	15,00
6	5,2	0,80	13,33
7	6,63	0,370	5,29
8	7,640	0,360	4,50
9	8,534	0,466	5,18
10	9,553	0,447	4,47
15	14,450	0,550	3,67
20	18,870	1,130	5,65
25	23,766	1,234	4,94
30	28,662	1,338	4,46
35	33,456	1,544	4,41
40	38,486	1,514	3,79
45	43,350	1,650	3,67
50	48,246	1,754	3,51

Nous avons calculé les erreurs comme suit :

$$\text{Erreur (en cm)} = |\text{valeur approximative} - \text{valeur exacte}|$$

$$\text{Erreur (\%)} = \frac{|\text{valeur approximative} - \text{valeur exacte}|}{\text{valeur exacte}} \times 100$$

En traçant les courbes correspondantes aux résultats obtenus dans le tableau de mesure on obtient les courbes ci-dessous.

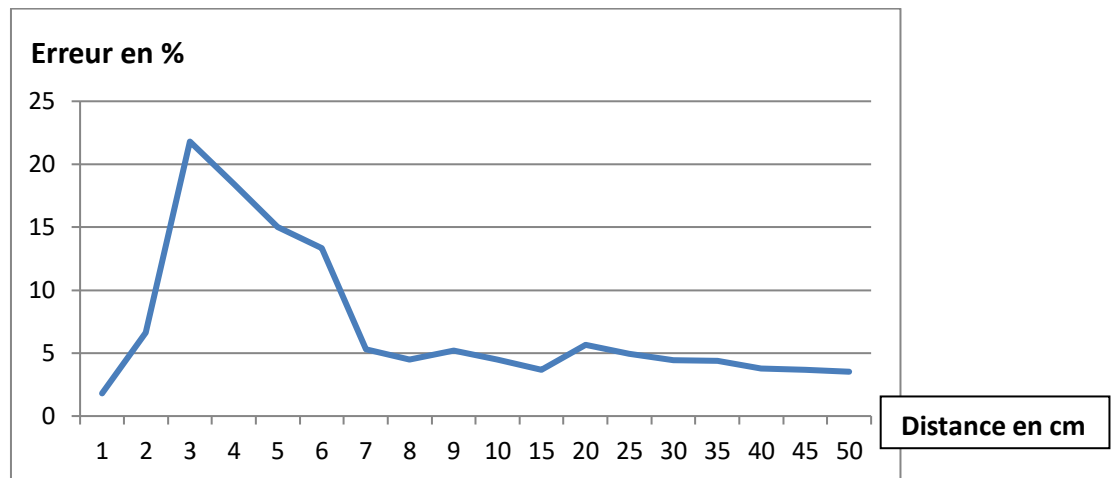


Figure II.10- Graphique d'erreurs en % du capteur ultrason.

II.5.2.7. Interprétation des courbes mesure et erreurs :

La remarque que nous avons faite après les mesures effectuées ; on peut dire que la théorie et la pratique ne sont pas tout à fait identiques. Il y a une légère dérive avec une marge d'erreur de 5% en moyenne au-dessus de 7 centimètres. Concernant la précision, en dessous de 1 centimètre, les données obtenues ne sont pas très convaincantes. Si on veut des mesures à peu près correctes, il faut oublier les premiers centimètres. Même dans ces conditions non optimales, le capteur reste tout à fait utilisable. On obtient une précision de +/- 2cm, ce qui n'est pas bien précis, mais pour un petit robot comme le notre, cela pourrait tout à fait suffire.

II.5.2.8. Avantage et Inconvénient :

Avantage :

- Les capteurs à ultrasons peuvent voir à travers l'air chargé en poussières, les brouillards d'encre et même s'il y a des minces dépôts sur la membrane du capteur ne nuisent pas à son fonctionnement d'où leur efficacité.
- ils sont utilisés dans les laboratoires de chimie pour les mesures du niveau de remplissage des orifices de plaques d'analyses ou de tubes à essai,
- dans l'industrie de l'emballage ils sont utilisés pour la détection des petites bouteilles. Ils peuvent être mis en œuvre avec facilité. Même les fils minces sont détectés de manière fiable.

Inconvénient :

Déviation de son le rayon acoustique peut être dévié sans pertes significatives par une surface réfléchissante et lisse. Il existe à cet effet des renvois d'angle à 90° renvois d'angle à 90° comme accessoires. Ces derniers peuvent être utilisés de manière avantageuse dans des espaces de montage restreints. Ce pour cela qu'il existe plusieurs variétés ou gamme de capteurs à Ultrason selon le besoin.



Figure II.11 : Déviation de son d'un capteur à Ultrason.

II.5.3 Capteur à infrarouge :

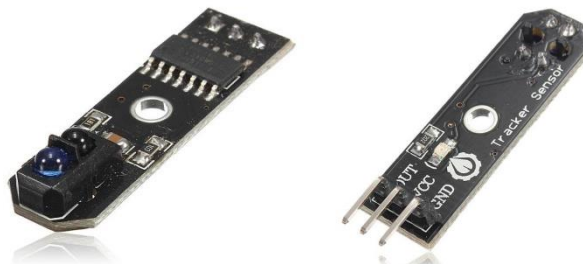


Figure II.12 : vue de face et de dos du capteur infrarouge TCRT5000

II.5.3.1. Caractéristique principe de fonctionnement :

II.5.3.1.1. Caractéristique :

- utilisant le capteur de réflexion infrarouge TCRT5000
- Distance de réflexion de détection: 1mm ~ 25mm
- tension de fonctionnement: 5 V
- la forme de sortie: signal numérique (0 et 1)
- avec des trous de boulon fixes pour une installation facile
- petite taille de carte PCB: 3.5 cm x 1 cm
- poids unique: 4.5g
- Capteur intégré TCRT5000 NSOR
- Applicable à une variété de plates-formes, y compris pour Arduino/AVR/ARM/PIC

II.5.3.1.2. Principe de fonctionnement :

La diode électroluminescente infrarouge du capteur TCRT5000 émet en continu des rayons infrarouges. Lorsque les rayons infrarouges émis ne sont pas réfléchis ou réfléchis mais l'intensité n'est pas assez grande, le tube récepteur infrarouge est toujours à l'état off, et l'extrémité de sortie du module est de bas niveau, indiquant La diode est toujours éteinte; lorsque l'objet détecté apparaît dans la plage de détection, la lumière infrarouge est réfléchié et l'intensité est assez grande, et le tube récepteur infrarouge est saturé. À ce moment, l'extrémité de sortie du module est élevée, ce qui indique que la diode est éclairée, et il y a aussi une autre diode de signalisation qui s'allume en rouge pour indiquer la détection d'un objet.

II.5.3.2 Câblage:

1. VCC: connectez le pôle positif de l'alimentation (5 V)
2. GND: connecté à l'alimentation négative
3. OUT: signal de commutation de haut/bas niveau (signal numérique 0 et 1)

II.6. Les actionneurs :

Un actionneur est un élément capable de créer un phénomène physique ou transformer l'énergie qu'il reçoit en chaleur, champs magnétique, lumière, mouvement, position, pression, son etc...

II.6.1. Les différents types d'actionneurs :

Il existe plusieurs types et variétés d'actionneurs qu'on ne peut pas les citer tous ici mais on peut énumérer quelques-uns.

II.6.1.1. Actionneur pneumatique :

Les actionneurs pneumatiques sont utilisés principalement pour des mouvements séquentiels simples ils utilisent de l'air comprimé à 6 bar et permettent de réaliser des vérins dont la force peut atteindre 50000N. L'air est fourni par un compresseur, complété de filtres, d'un séparateur d'eau et d'un déshuileur. Il est souvent produit pour tous les ateliers, et distribué à toutes les machines. Les actionneurs sont généralement des vérins linéaires, mais aussi des moteurs rotatifs. On utilise également des aspirateurs suceurs à vide pour saisir des objets. Ils sont commandés en tout ou rien par des distributeurs, actionnés mécaniquement ou électriquement.[1]

II.6.1.2. Actionneur hydraulique :

Un actionneur hydraulique est composé de deux parties : une partie vérin et une partie valve. Dans un système à commande proportionnelle, on place dans l'étage de pilotage (valve) une bobine (solénoïde) qui régule la pression du fluide proportionnellement à la consigne électrique (courant tension). [26]

II.6.1.3. Actionneur piézoélectrique :

C'est un convertisseur électrique-mécanique non résonant qui est généralement utilisé pour engendrer des microdéplacements. On les rencontre principalement en microélectronique pour le positionnement des circuits intégrés et en microscopie à effet tunnel pour cartographier les surfaces des matériaux. En fait, physiquement, tous les systèmes ont des fréquences de résonance.[27]

II.6.1.4. Actionneur électromécanique :

Est un transducteur assurant une conversion d'énergie et d'information de la forme électrique à la forme mécanique. Un actionneur électromécanique est aussi appelé moteur. Selon les applications, l'aspect énergétique ou la qualité du signal prime. Le rendement énergétique est parfois déterminant. Dans d'autres cas, c'est le caractère linéaire de la transformation qui importe. Dans d'autre cas encore, c'est le temps de réaction ou la fréquence limite qui sont déterminants. [28]

II.6.2. Mode de fonctionnement des actionneurs :

II.6.2.1. Mode tout ou rien :

L'actionneur ou le moteur est connecté à une alimentation hydraulique ou pneumatique de pression constante ou à une alimentation électrique de tension et fréquence constantes. Sa vitesse de déplacement ou de rotation dépend de son principe de fonctionnement et des caractéristiques de son alimentation, mais aussi de la charge (frottement, couple d'usinage, etc.). Non alimenté, il ne produit plus aucune force ou couple et se laisse entraine par la charge, généralement, il s'arrête après un temps plus ou moins long sous l'effet des frottements, ou soudainement sous l'action d'un frein mécanique. La commande est alors particulièrement simple à réaliser, à l'aide d'un distributeur pour les entraînements pneumatiques ou hydraulique et d'un interrupteur pour les entraînements électriques. [1]

II.6.2.2. Mode contrôlé en vitesse :

En ajustant la pression hydraulique ou pneumatique, la tension électrique ou la fréquence, il est possible de modifier la vitesse d'un actionneur ou d'un moteur de manière continue, au moins dans une certaine plage (par exemple de 20% à 100% de la vitesse nominale). Toutefois la vitesse reste plus ou moins dépendante de la charge.

II.6.2.3. Mode réglé en vitesse :

Le principe de la régulation est illustré dans la figure ci-dessous : l'opérateur ou le programme d'automate choisit une valeur de consigne en fonction de la vitesse qu'il souhaite obtenir. La valeur réelle est mesurée et fournit le signal de contre- réaction, qui est comparé à la valeur de consigne. La différence entre ces deux valeur est appelée écart de régulation s'efforce de le minimiser en ajustant la grandeur de réglage et, par l'intermédiaire de l'amplificateur de puissance, l'alimentation du moteur. [1]

II.6.2.4. Mode servomoteur-réglé en position :

En ajustant un capteur de position sur l'arbre du moteur ou sur la charge en mouvement, il est possible de réaliser des déplacements point à point et d'arrêter le moteur à des positions très précises. C'est cet actionneur que nous allons utiliser dans notre projet.

II.6.2.5. Mode pas à pas :

Il est combiné le mode tout ou rien et le mode servomoteur. L'actionneur travaille bien en mode tout ou rien, mais il est alimenté par une succession d'impulsions. A chaque impulsion, il avance d'une petite distance appelée pas ou incrément. La distance parcourue dépend directement du nombre d'impulsions reçues. La vitesse dépend de la fréquence des impulsions. De plus, lorsqu'il ne reçoit plus d'impulsions, un tel actionneur est tenu en place avec une certaine force de maintien. [1]

II.6.3. Les actionneurs électriques :

II.6.3.1. Les moteurs à courant continu à aimant permanent

Dans l'organisation d'une machine à courant continu on peut distinguer trois principaux éléments :

- Les pôles inducteurs avec les enroulements ou leurs aimants placés généralement sur le stator.
- L'induit, dont les différentes voies d'enroulements sont connectées au collecteur.
- Les organes mécanique, permettant la rotation du rotor et maintient des différents sous ensemble.

II.6.3.1.1. Le principe de fonctionnement :

Selon la loi de Laplace : un conducteur traversé par un courant I et placé dans un champ magnétique B est soumis à une force F dont le sens est déterminé par la règle des trois doigts de la main droite. $F = B \times I \times L$ avec : (L : longueur du conducteur en mètre).

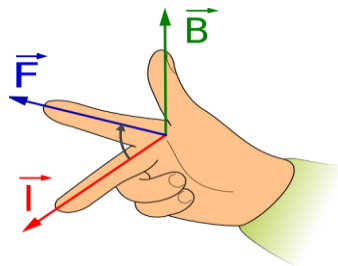


Figure II.13 : Règle de la main droite [16]

Lorsque l'on place une spire parcourue un courant dans le champ magnétique il apparaît une couple de force ; ce couple crée un couple de rotation qui fait dévier la spire de plus ou moins 90° par rapport au plan initial, le sens du courant restant inchangé dans la spire au cours de ce déplacement.

Pour inverse le sens de rotation, il suffit d'inverser le sens du courant dans la spire donc d'inverser la polarité de la tension d'alimentation.

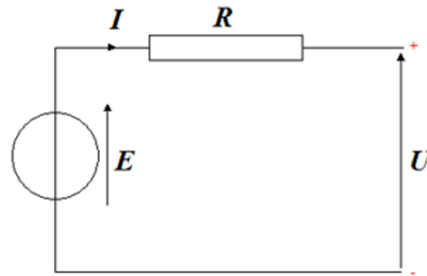


Figure II.14 : Model électrique simple du moteur à courant continu

R : Résistance des enroulements de l'induit (Ω).

E : Force électromotrice induite (**V**).

I : Courant traversant les spires de l'induit (**A**).

U : Tension d'alimentation de l'induit(**V**).

Equation électrique du moteur : $U = R \times I + E$

Moteur synchrone sans balais et servomoteur

II.7. Les actionneurs du robot (Moteur DC, Servo-moteur et Relais

L298N) :

Sans rentre dans les détails concernant les actionneurs, puis que nous avons déjà donné de détails de ce dernier dans ci haut. Ici ce que nous allons mentionner les actionneurs que avons utilise pour ce projet à savoir : les moteurs à courant continu, le servo-moteur et le relais L298N. Dans le cas des appareils, objets, machines qui ont un fonctionnement automatique, l'actionneur va faire son action seulement si un ordre lui parvient de la partie commande.

II.7.1. Moteur à courant continu :

Afin que le robot soit capable de se déplacer il est nécessaire d'utiliser les moteurs, qu'on a décidé de réaliser un robot à quatre roues, d'ou l'utilisation de moteur CC pour commander les quatre roues. L'utilisation des moteurs CC c'est un choix pour mener à bien

l'objectif envisagé, il n'y a pas beaucoup de vibrations, haute consommation d'énergie et plus rapide par rapport à un moteur pas à pas.



Figure II.15 : Moteur CC avec son réducteur + roue

II.7.1.1. Les caractéristiques du moteur :

- Tension de fonctionnement du moteur CC : 3V - 6V
- vitesse à vide: 125 tr / min à 3V, 208 tr / min à 6V
- courant à vide: 150mA à 3V, 175mA à 6V
- taux de réduction : 48 : 1
- RPM (avec pneus) : 100-240
- diamètre du pneu : 65mm
- poids du moteur (g) : 29 / chaque
- la taille du moteur : 70mm x 22mm x 18mm

Avec une boîte de vitesses qui est idéale pour la voiture robotisée ou un robot suiveur de ligne D'après les caractéristiques des moteurs que nous avons utilisé, on ne peut pas les alimenter par une tension supérieur ou égal a 9 V, ceci nous oblige à utiliser un étage entre le contrôleur et le moteur qui joue le rôle de circuit de puissance ou de commande, et en plus on a besoin de contrôler les moteurs pour qu'ils tournent vers deux sens. C'est pour cela qu'on a choisi le circuit L298N puisqu'il est disponible et il peut commander quatre moteurs.

II.7.1.2. Avantages et inconvénients du moteur CC :

L'avantage principal des machines à courant continu réside dans leur adaptation simple aux moyens permettant de régler ou de faire varier leur vitesse, leur couple et leur sens de rotation : les variateurs de vitesse, voire leur raccordement direct à la source d'énergie : batteries d'accumulateur, piles, etc.

Le principal problème de ces machines vient de la liaison entre les balais, ou « charbons » et le collecteur rotatif. Ainsi que le collecteur lui-même et la complexité de sa réalisation. De plus il faut signaler que :

- plus la vitesse de rotation est élevée, plus la pression des balais doit augmenter pour rester en contact avec le collecteur donc plus le frottement est important ;
- aux vitesses élevées les balais doivent donc être remplacés très régulièrement ;
- le collecteur imposant des ruptures de contact provoque des arcs, qui usent rapidement le commutateur et génèrent des parasites dans le circuit d'alimentation, ainsi que par rayonnement électromagnétique.

Un autre problème limite les vitesses d'utilisation élevées de ces moteurs lorsque le rotor est bobiné, c'est le phénomène de « défretage », la force centrifuge finissant par casser les liens assurant la tenue des ensembles de spires (le fretage). La température est limitée au niveau du collecteur par l'alliage utilisé pour braser les conducteurs du rotor aux lames du collecteur. Un alliage à base d'argent doit être utilisé lorsque la température de fonctionnement dépasse la température de fusion de l'alliage classique à base d'étain.

Un certain nombre de ces inconvénients ont partiellement été résolus par des réalisations de moteurs sans fer au rotor, comme les moteurs « disques » ou les moteurs « cloches », qui néanmoins possèdent toujours des balais.

Les inconvénients ci-dessus ont été radicalement éliminés grâce à la technologie du moteur sans balais (aussi dénommé moteur *brushless*, ou improprement « moteur à courant continu sans balais »), qui est une machine synchrone auto-pilotée.[32]

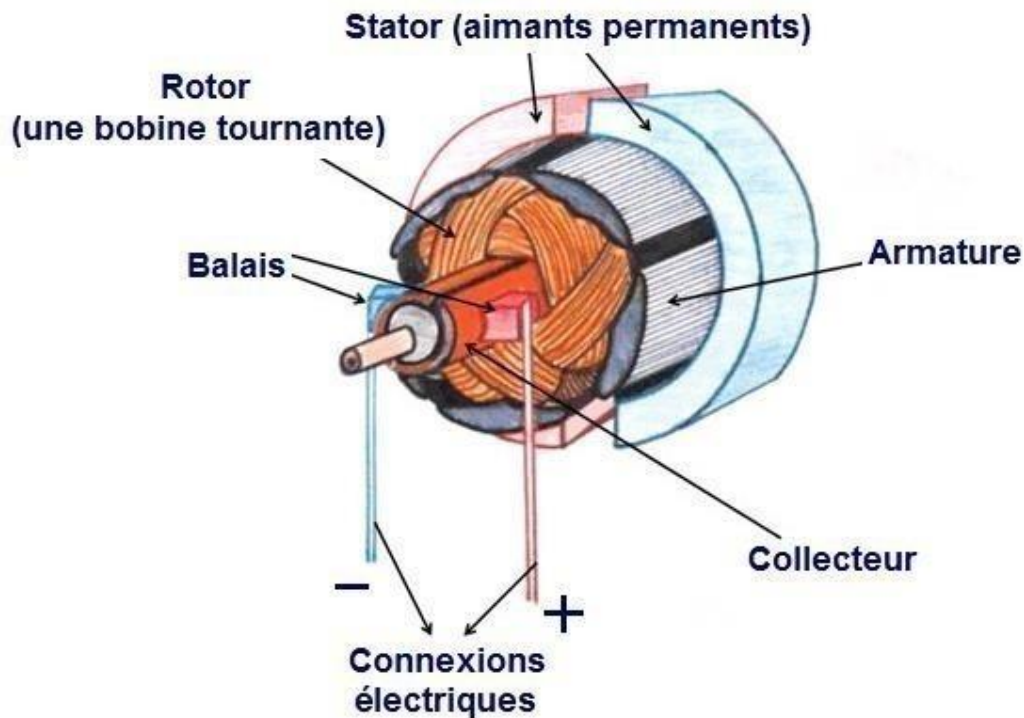


Figure II.16 : Structure interne du moteur à courant continu.[33]

II.7.2. Servo-moteur :

Définition : Un servomoteur est un système qui a pour but de produire un mouvement précis en réponse à une commande externe, il est composé d'un moteur à courant continu. C'est un actionneur qui mélange l'électronique, la mécanique et l'automatique.



Figure II.17 : Servomoteur MG90S

On a utilisé un servomoteur pour que le robot en plus de détecter un obstacle devant lui il cherche des obstacles à gauche et à droite à fin de déterminer la direction à prendre. Premièrement on a utilisé un servomoteur SG90 malheureusement la pièce était défectueuse donc on a dû changer avec MG90S qui est conçu pour supporter un certain poids.

II.7.2.1. Les caractéristiques du servo-moteur

- Nom : MG90S servo-moteur
- Poids: environ 13.4 g
- Dimension: 22.8 x 12.2 x 28.5 mm
- Couple de décrochage: 1.8 kg / cm (4.8v), 2.2 kg / cm (6v)
- Vitesse de fonctionnement: 0.1sec / 60degree (4.8v), 0.08sec / 60degree (6v)
- Tension de fonctionnement: 4.8-6.0v
- Type de moteur: sans noyau moteur
- Longueur de câble de connecteur: 175mm

La connexion du servomoteur avec Arduino est facile près que tous les fils d'un servomoteur ont le même ordre de couleur, le rouge avec +5V, le noir avec GND et le jaune avec le pin qu'on devra le préciser dans le programme de l'Arduino. Il y a deux types de connexions possibles pour pouvoir alimenter le servo-moteur, direct ou avec Arduino.

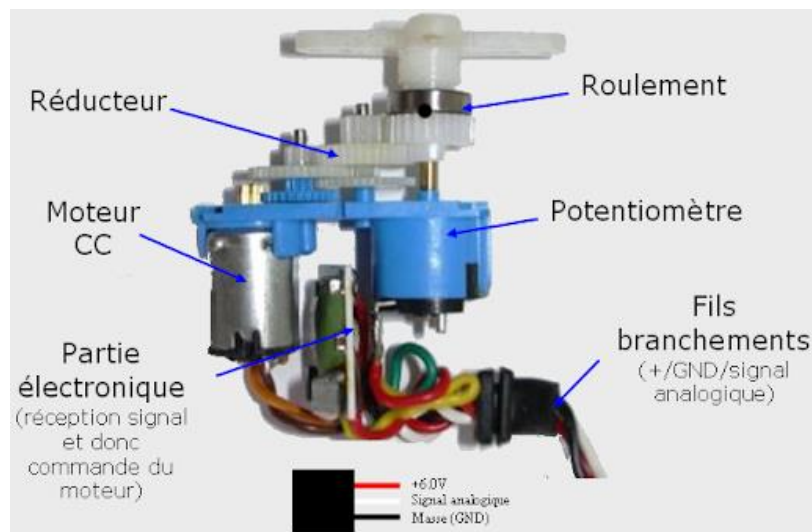


Figure II.18 : La structure interne du servo-moteur.[24]

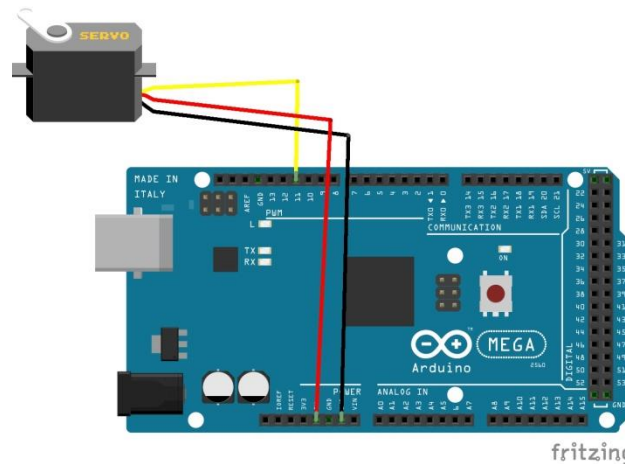


Figure II.19 : Connexion d'un servo-moteur à un Arduino.

II.7.2.2 Le principe de fonctionnement du servo-moteur

Pour commander un servo-moteur, il faut lui envoyer ce qu'on appelle un train d'impulsions électriques, qui peut se traduire par envois de courant électrique ou qui se suivent à intervalle et durée précis. Le servo-moteur va interpréter ces impulsions de la manière suivante :

Si on envoie une impulsion de 1ms le servo-moteur reste à 0° , une impulsion de 1,5ms il reste à 90° et par la suite à 2ms il se positionne à 180° . Grâce à cette formule, on peut obtenir toutes les valeurs d'impulsions pour des positions entre 0° et 180° , on peut donc faire un chronogramme des valeurs pour obtenir la durée de l'impulsion en fonction de l'angle :

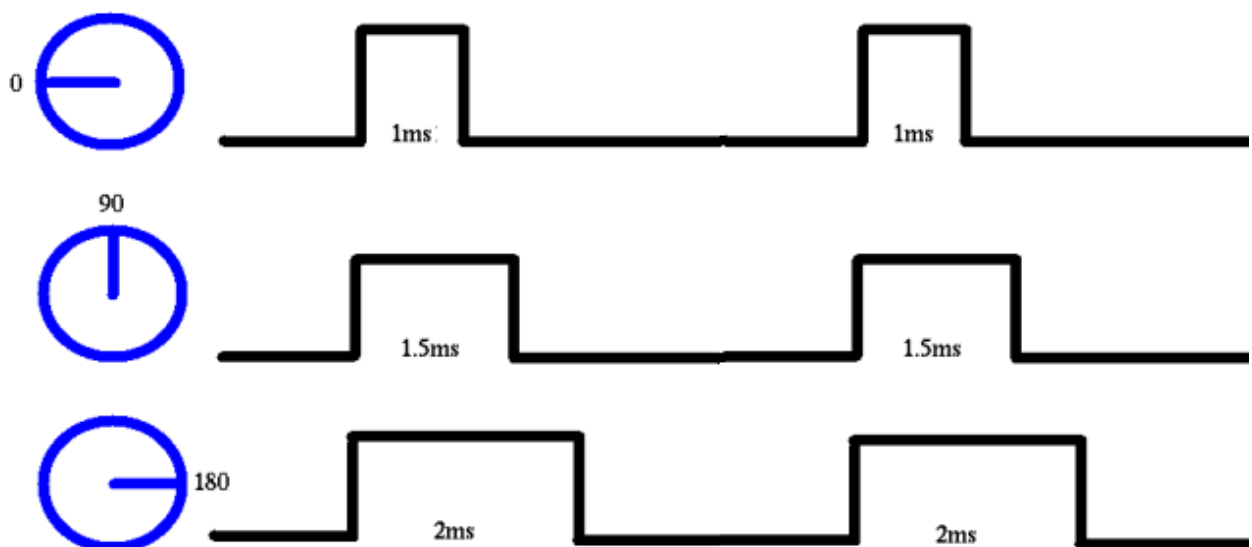


Figure II.20 : Chronogramme de commande d'un servo-moteur

```

Servo.ino
Fichier Édition Croquis Outils Aide
Servo.ino $
#include <Servo.h>
Servo myservo; // création de l'objet myservo
int pin_servo = 6; // Pin 6 sur lequel est branché le servo sur l'Arduino si vous utilisez un ESP32 remplacez le 6 par 4 et si vous utilisez un ESP8266 remplacez le 6 par 2
int pos = 0; // variable permettant de conserver la position du servo
int angle_initial = 0; //angle initial
int angle_final = 180; //angle final
int increment = 1; //incrément entre chaque position
bool angle_actuel = false; //Envoi sur le port série la position courante du servomoteur
void setup() {
  Serial.begin(9600);
  while(!Serial){};
  myservo.attach(pin_servo); // attache le servo au pin spécifié sur l'objet myservo
}
void loop() {
  for (pos = angle_initial; pos <= angle_final; pos += increment) { // Déplace le servo de 0 à 180 degrés par pas de 1 degré
    myservo.write(pos); // Demande au servo de se déplacer à cette position angulaire
    delay(30); // Attend 30ms entre chaque changement de position
    if (angle_actuel) {
      Serial.println(myservo.read());
    }
  }
  for (pos = angle_final; pos >= angle_initial; pos -= increment) { // Fait le chemin inverse
    myservo.write(pos);
    delay(30);
    if (angle_actuel) {
      Serial.println(myservo.read());
    }
  }
}
}
Enregistrement terminé.
19 Arduino Duemilanove or Diecimila, ATmega328P sur COM8

```

Figure II.21 : Programme pour faire varier l'angle avec un servo-moteur.

II.7.3. Le Relais L298N :

Définition : le relais est un composant électrique réalisant la fonction d'interfaçage entre un circuit de commande, généralement bas niveau, et un circuit de puissance alternatif ou continu. Module L298N H-bridge, pour le contrôle de 2 moteurs courant continu ou 1 moteur pas à pas bipolaire avec un microcontrôleur Arduino. - Il pilote directement la vitesse d'un ou deux moteurs à courant continu CC dans les deux sens de rotation, avec une interface de sortie sous 5V, et fournit une alimentation de 5V. Avec ce module, on peut faire facilement varier la vitesse et la direction de rotation du ou des moteurs. Le module L298N est un double pont-H, ou H-Bridge et peut contrôler des moteurs dans une plage de tension entre 5 et 35 VDC.

Il est très utile dans un tas de projets Arduino et/ou de robotique. Il peut piloter des charges inductives (pensez bobines) avec une commande TTL parfaitement adaptée aux microcontrôleurs Arduino.

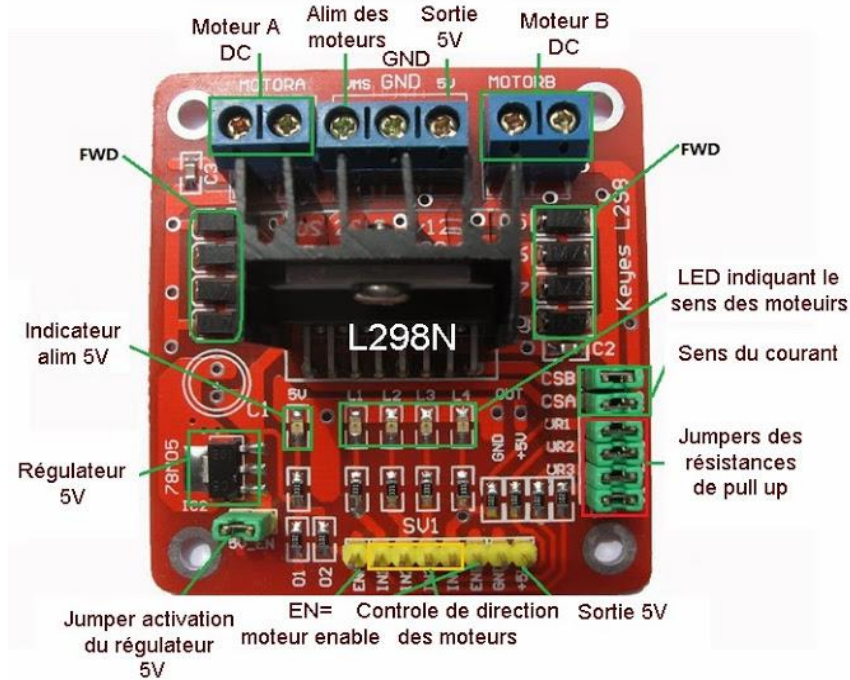


Figure II.22 : Relais L298N. [22]

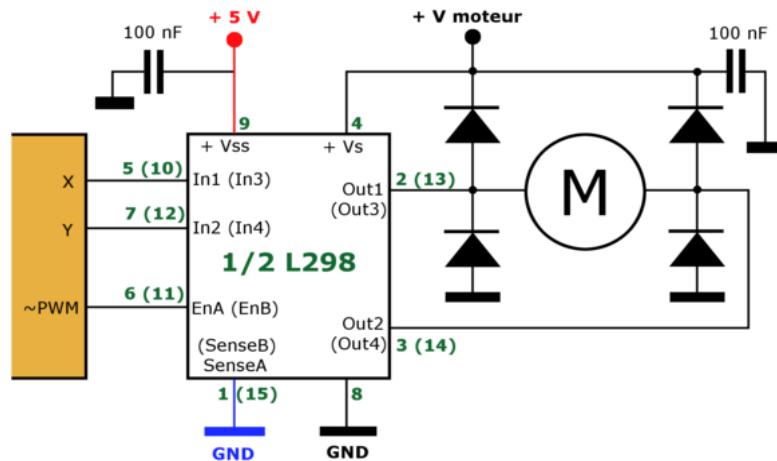


Figure II.23 : Schéma du circuit de commande des moteurs

Les signaux appliqués aux entrées In1 (5) et In2 (7) déterminent le sens de rotation du moteur ou son arrêt rapide. Le signal appliqué à l'entrée EnA (6) commande la vitesse de rotation s'il est du type PWM, ou on peut ne pas les utiliser. Une fois de plus, si on veut commander

deux moteurs CC, le schéma ci-dessus montre comment interfacier la moitié d'un circuit L298 pour commander un moteur ; le montage doit être fait en double à partir du même circuit L298N pour commander deux moteurs.

II.7.3.1. Caractéristiques du relais L298N :

- Circuit intégré L298N, double pont en H de pilotage.
- Borne d'alimentation (de la partie puissance) VMS : + 5 V à + 35 V.
- Courant crête en sortie : 2 A par pont.
- Tension d'alimentation (de la partie logique): Vss de 4,5 à 5,5 V
- Intensité de courant dans la partie logique 0 à 36 mA.
- Signal de commande : niveau haut = 4.5 à 5.5 V, niveau bas = 0 V.
- Puissance consommée maximale : 20 W.
- Température de stockage: -25 °C ~ + 130 °C
- Taille de la plaque : 55 mm x 60 mm x 30 mm
- Poids : 33 g

II.7.3.2. Principe du fonctionnement du relais L298N :

L'Arduino seul est incapable d'alimenter deux moteurs à courant continu. Ce pour cela que nous avons donc passer par une carte de puissance qui est le relais L298N, qui permettra de commander les quatre moteurs dans les deux sens de rotation et à des vitesses pouvant varier entre 0 tr/min et la vitesse maximale du moteur selon la programmation. En principe le relais L289N est fait pour deux moteurs selon le schéma ci-dessous, mais dans notre cas nous avons relié quatre moteurs à notre relais L298N puis que notre châssis est composé de quatre roues motrices donc on a connecté deux moteurs de plus.

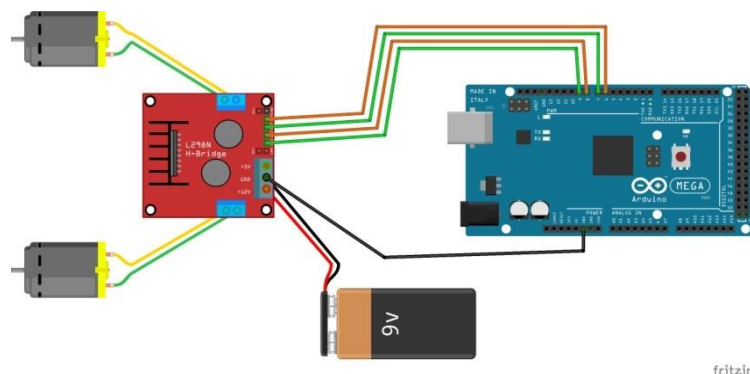
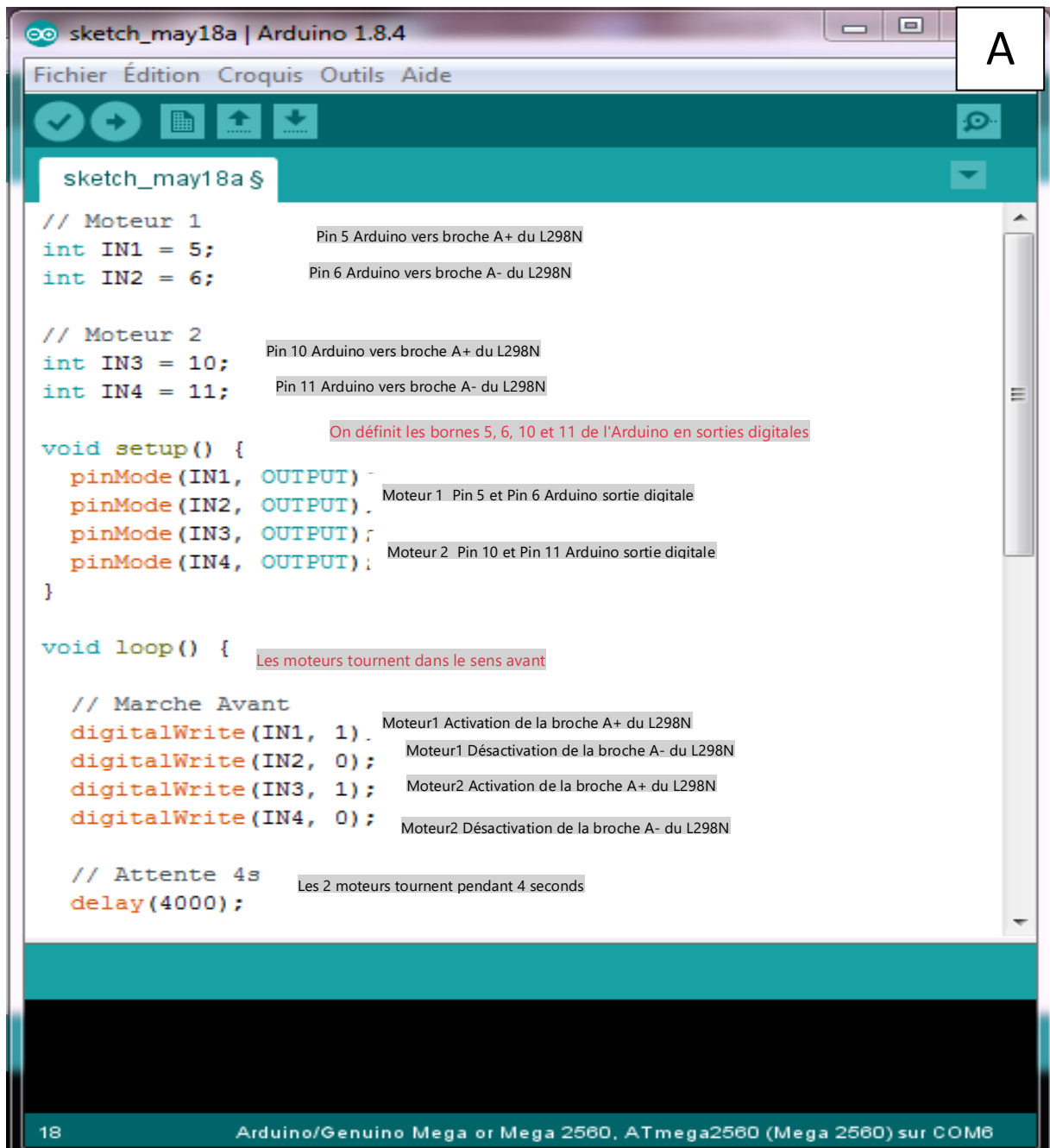


Figure II.24 : Connexion du relais avec Arduino, les moteurs et l'alimentation

Ici la connexion des pins les traits roses et verts connectés aux pins EN1, EN2, EN3, EN4 du relais contrôlent la direction des moteurs et du côté de l'Arduino la connexion se fait selon le programme, on déclare le numéro des pins dans le programme.

Voici un exemple ci-dessous du programme qui fait tourner les moteurs dans un sens pendant 4 secondes et arrête les moteurs pendant 2 secondes, puis recommence le même processus dans le sens inverse.



```

sketch_may18a | Arduino 1.8.4
Fichier Édition Croquis Outils Aide

sketch_may18a $
// Moteur 1                                     Pin 5 Arduino vers broche A+ du L298N
int IN1 = 5;
// Moteur 2                                     Pin 6 Arduino vers broche A- du L298N
int IN2 = 6;

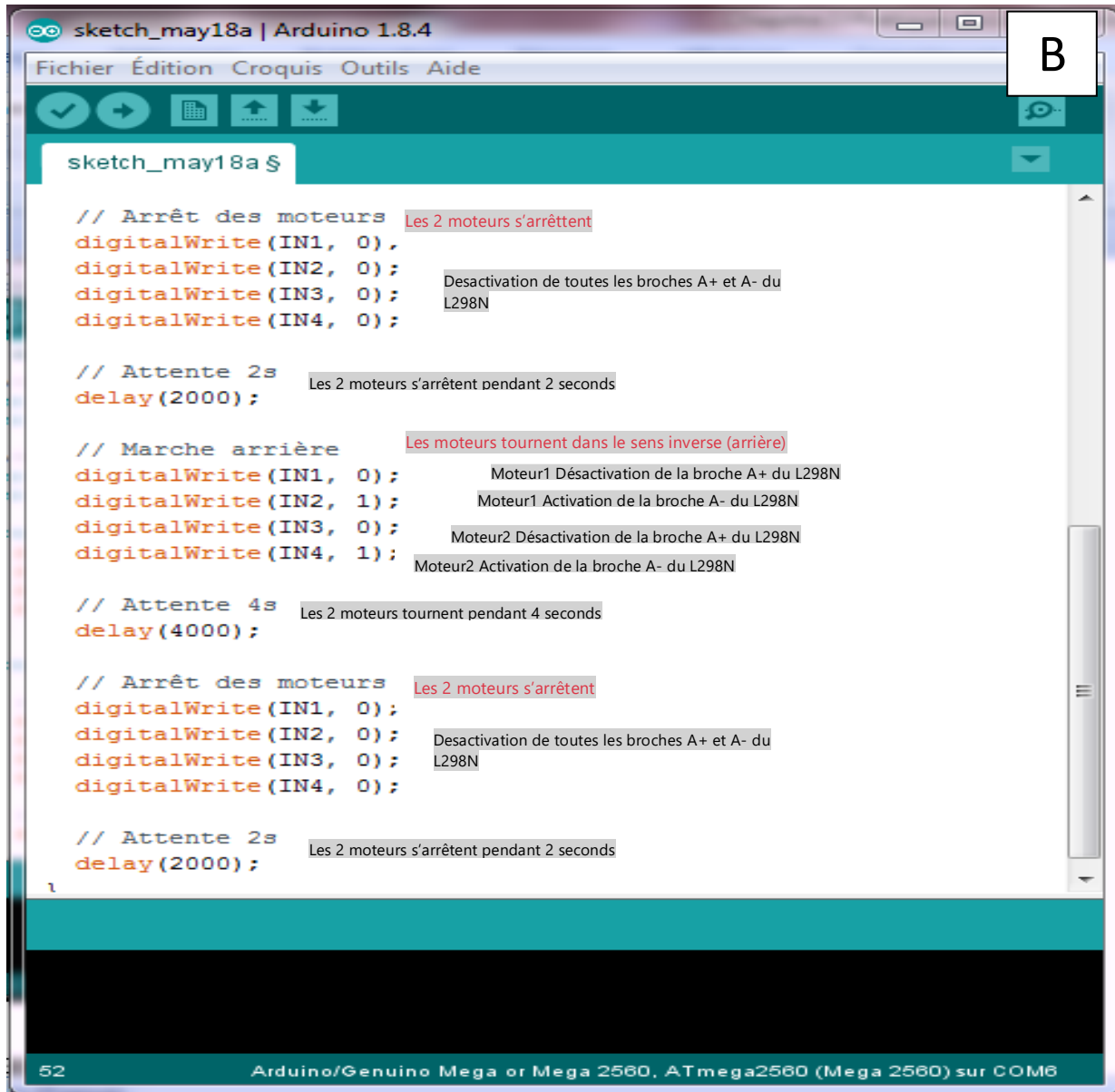
// Moteur 2                                     Pin 10 Arduino vers broche A+ du L298N
int IN3 = 10;
// Moteur 2                                     Pin 11 Arduino vers broche A- du L298N
int IN4 = 11;

// On définit les bornes 5, 6, 10 et 11 de l'Arduino en sorties digitales
void setup() {
  pinMode(IN1, OUTPUT); // Moteur 1 Pin 5 et Pin 6 Arduino sortie digitale
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT); // Moteur 2 Pin 10 et Pin 11 Arduino sortie digitale
  pinMode(IN4, OUTPUT);
}

void loop() { // Les moteurs tournent dans le sens avant
  // Marche Avant
  digitalWrite(IN1, 1); // Moteur1 Activation de la broche A+ du L298N
  digitalWrite(IN2, 0); // Moteur1 Désactivation de la broche A- du L298N
  digitalWrite(IN3, 1); // Moteur2 Activation de la broche A+ du L298N
  digitalWrite(IN4, 0); // Moteur2 Désactivation de la broche A- du L298N

  // Attente 4s // Les 2 moteurs tournent pendant 4 seconds
  delay(4000);
}
18 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM6

```



```

sketch_may18a | Arduino 1.8.4
Fichier Édition Croquis Outils Aide

sketch_may18a $

// Arrêt des moteurs Les 2 moteurs s'arrêtent
digitalWrite(IN1, 0);
digitalWrite(IN2, 0);
digitalWrite(IN3, 0); Desactivation de toutes les broches A+ et A- du
digitalWrite(IN4, 0); L298N

// Attente 2s Les 2 moteurs s'arrêtent pendant 2 seconds
delay(2000);

// Marche arrière Les moteurs tournent dans le sens inverse (arrière)
digitalWrite(IN1, 0); Moteur1 Désactivation de la broche A+ du L298N
digitalWrite(IN2, 1); Moteur1 Activation de la broche A- du L298N
digitalWrite(IN3, 0); Moteur2 Désactivation de la broche A+ du L298N
digitalWrite(IN4, 1); Moteur2 Activation de la broche A- du L298N

// Attente 4s Les 2 moteurs tournent pendant 4 seconds
delay(4000);

// Arrêt des moteurs Les 2 moteurs s'arrêtent
digitalWrite(IN1, 0);
digitalWrite(IN2, 0); Desactivation de toutes les broches A+ et A- du
digitalWrite(IN3, 0); L298N
digitalWrite(IN4, 0);

// Attente 2s Les 2 moteurs s'arrêtent pendant 2 seconds
delay(2000);

```

52 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM8

Figure II.25 : A et B programme de contrôle des moteurs.

II.8. Alimentation du robot:

Comme tout appareil électronique notre robot a besoin d'être alimenté par une source d'énergie pour fonctionner. La source d'alimentation utilisée pour ce projet est des batteries de 9V, au départ nous avons utilisé une seule batterie mais il y a eu des complications au niveau de consommation d'énergie vu le nombre des composantes. Donc on est obligé d'ajouter une autre batterie de 9V.

Tableau II.4 : Tension et courant de fonctionnement des composantes du robot

Composantes	Tension de fonctionnement	Courant de fonctionnement
Arduino	7 à 12V	500mA
Relais L298N	5 à 35V	2A
Moteurs x4	3 à 6V par moteur	4x150mA
Capteur à Ultrason HC-SR04	5V	15mA
Capteur à infrarouge	5V	15mA
Servo-moteur MG90S	4,8 à 6V	40Ma

II.8.1. Batterie :

Type des batteries utilisées sont : deux batteries rechargeables ; marque Energizer 9V, 175mAh qui nous ont été fournies par le laboratoire de l'université comme les autres composantes et une batterie rechargeable ; marque Camelion 9V, 200mAh qu'on a dû acheter. Pour le bon fonctionnement du projet il faut une batterie supplémentaire.

II.9. La conception du robot :

Le robot doit suivre le schéma correspondant à un modèle au sein d'une architecture de contrôle. L'activité d'un tel robot se ramène aux tâches énoncées ci-après architecture de contrôle. L'activité de notre robot se ramène aux tâches suivantes :

- **Percevoir:** Le robot doit acquérir des informations sur l'environnement dans lequel il évolue par l'intermédiaire de capteurs ; Ces informations permettent de mettre à jour un modèle de l'environnement (architecture hiérarchique), ou peuvent être directement utilisées comme entrées de comportements de bas niveau (architectures purement réactives).

- **Décider:** Le robot doit définir des séquences d'actions résultant d'un raisonnement appliqué sur un modèle de l'environnement (architecture hiérarchique) ou répondant de manière réflexe à des stimuli étroitement liés aux capteurs (architectures purement réactives).

- **Agir:** Le robot doit enfin exécuter les séquences d'actions en envoyant des consignes aux actionneurs par l'intermédiaire des asservissements. Au sein de la partie décision, il est nécessaire de contrôler l'exécution de ces actions afin que le robot s'adapte rapidement à des événements imprévus.

La représentation simplifiée des éléments qui composent le robot. Nous avons commencé par comprendre les caractéristiques et le fonctionnement chaque pièce que nous allons utiliser

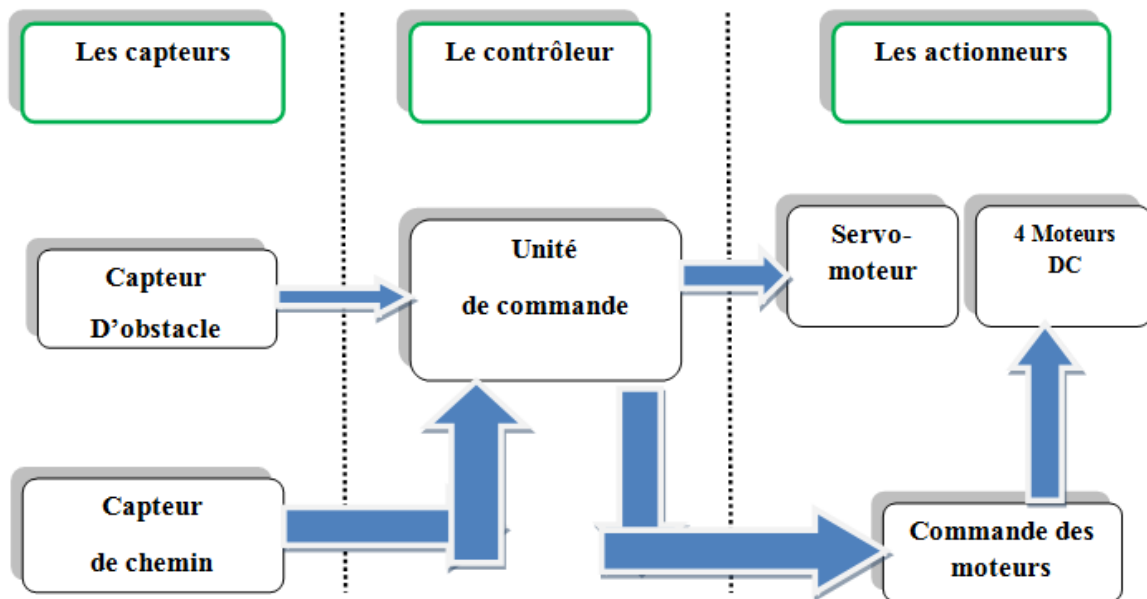


Figure II.26 : Schéma fonctionnel du robot.

II.10. Les étapes du montage mécanique :

II.10.1. Châssis 4WD:

Ce châssis aussi nous a été fourni par le laboratoire LAIG (Laboratoire d'Automatique et d'Informatique de Guelma) sous la direction de notre encadreur comme pour toutes les autres composantes du robot.

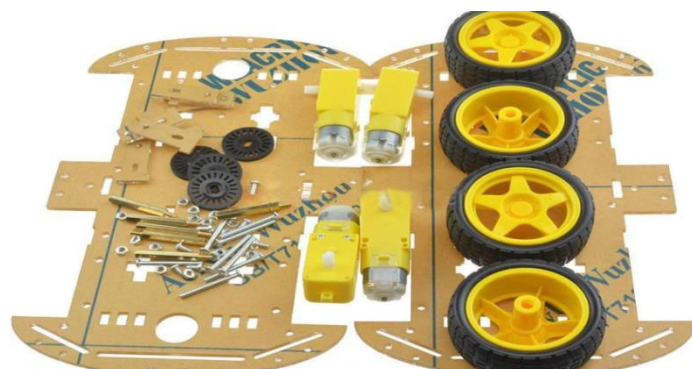


Figure II.27 : Châssis du robot et ses différentes pièces qui le composent

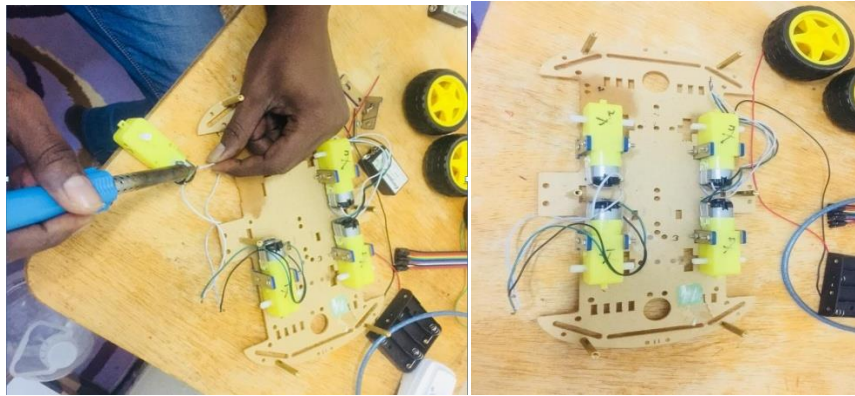


Figure II.28 : Soudage du fil des moteurs et les placer sur la partie inférieure du châssis

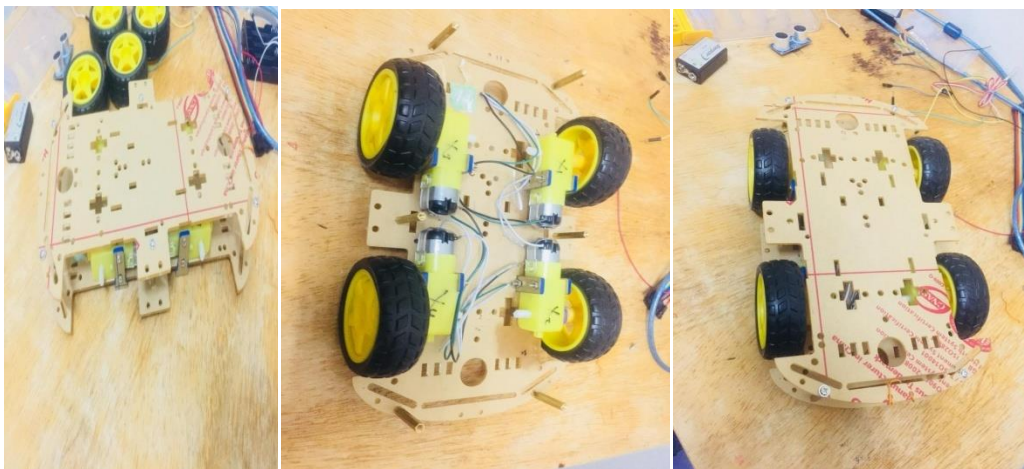


Figure II.29 : Placé la partie supérieure du châssis et faire monter les roues

II.10.2. La partie inférieure du robot :

Tout d'abord nous avons commencé par placer les capteurs à infrarouge TCRT5000

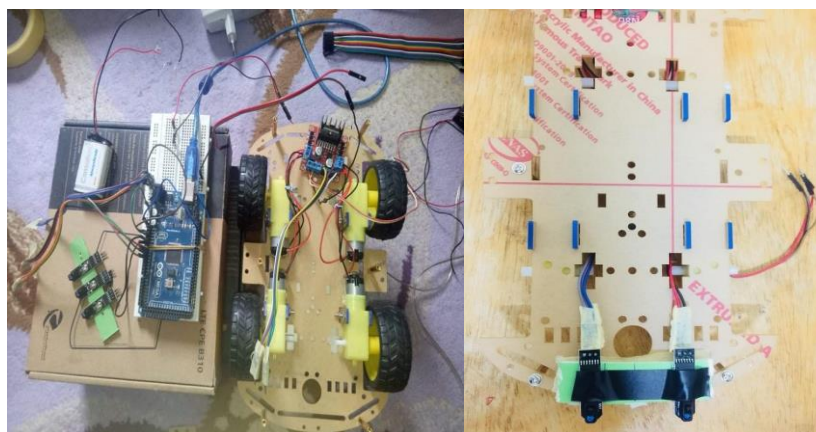


Figure II.30 : Montage des capteurs infrarouge sur le châssis du robot

Nous avons placé trois capteurs à infrarouge au début (photo à gauche) et on a eu un peu de soucis à faire le programme, l'aide de trois capteurs c'est pour qu'il y ait plus fluidité de déplacement pour le robot. Par la suite on a pu réaliser avec deux capteurs et les capteurs sont placés en dessous de la partie inférieure du robot de manière à ce qu'il soit à 20mm du sol (photo à droite).

Toujours nous sommes sur la partie inférieure du robot.

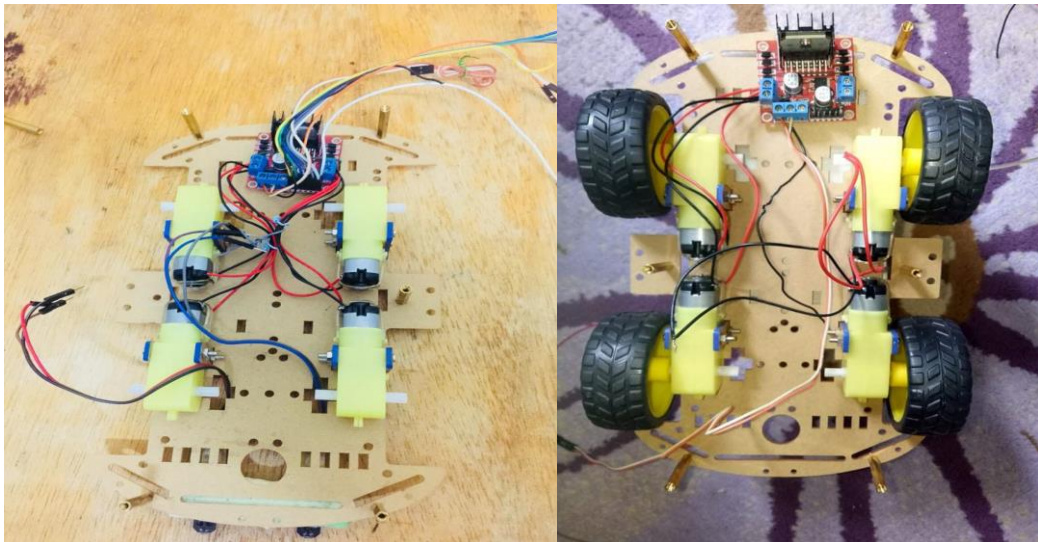


Figure II.31 : Placement du relais L298N sur le châssis et le câblage de quatre moteurs avec le relais

II.10.3. La partie supérieure du robot :

Ces quatre photos ci-dessous montrent comment nous avons placé le servo-moteur, le capteur à ultrason et deux supports pour l'alimentation (batterie 9V rechargeable).

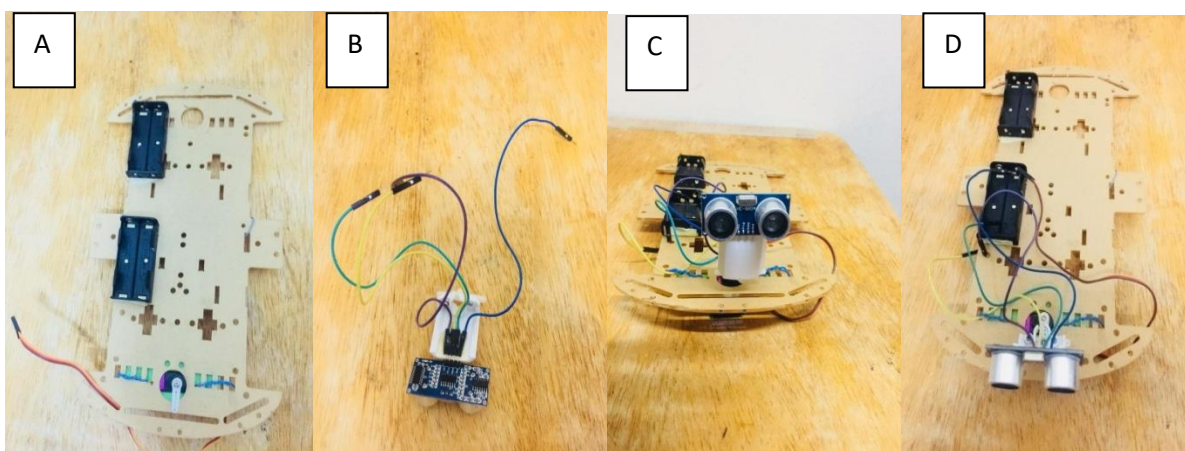


Figure II.32 : Montage du capteur à ultrason, support des batteries sur le châssis

Sur la photo A montre comment on a placé le servo-moteur et les deux supports de batteries. Sur la seconde photo montre comment on a adapté un support pour le capteur à ultrason pour pouvoir bien le lié avec le servo-moteur et c'est ce qui est faite sur la troisième et quatrième photo.

II.10.4. Rassemblement des deux parties du robot :

Sur les photos ci-dessous montrent comment nous avons fixé les parties (partie inférieure et supérieure) du robot, placé notre Arduino et une plaque d'essai pour nous facilité la connexion des différents composants du robot.

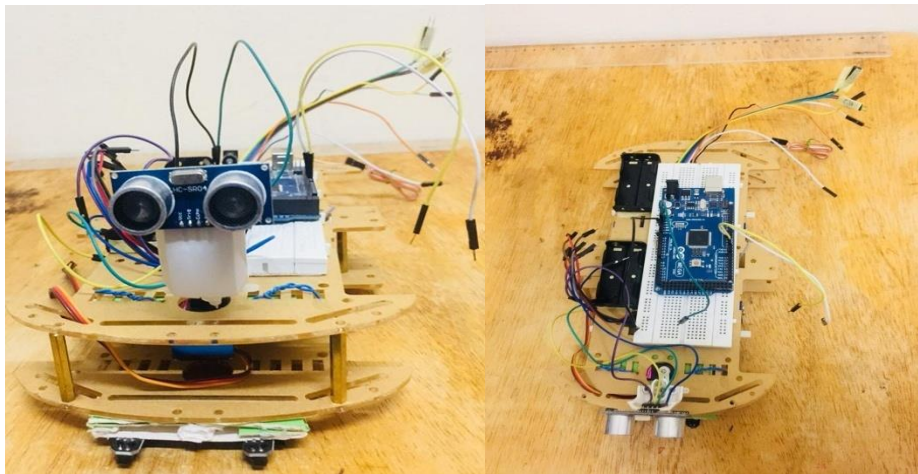


Figure II.33 : Assemblage de deux parties inferieur et supérieur du robot

II.11 Câblage des toutes composantes du robot :

Il s'agit de la représentation tous les composants et toutes les liaisons du circuit électronique de notre robot. Selon le schéma ci-dessous les composants sont représentés par leur photo tandis que les liaisons sont schématisées par des traits de couleur qui lient les composants entre eux. Sur ce schéma on tien compte de l'alimentation de l'Arduino.

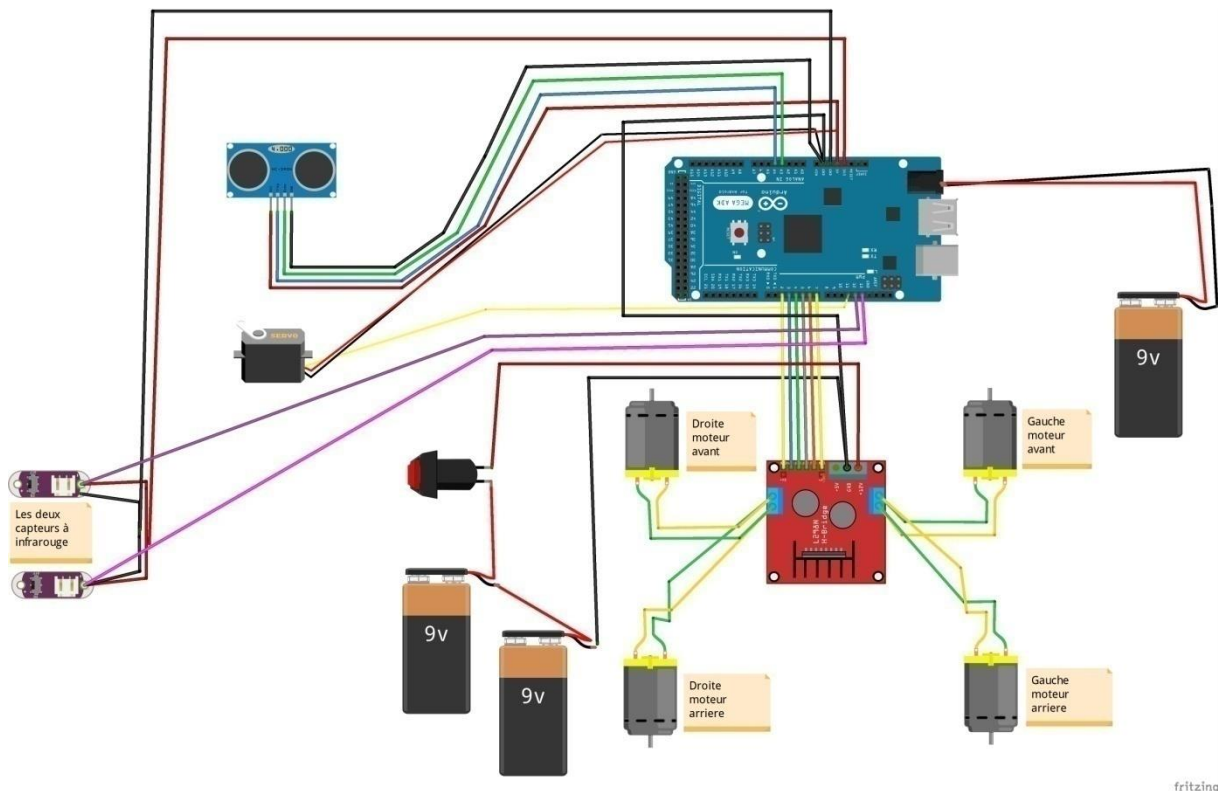


Figure II.34 : Schéma structurel du robot

Comme montré sur la figure précédente c'est le même câblage que nous avons utilisé sur le robot. Ici il est question de connaître les entrées/sorties de chaque pin numérique et analogique de l'Arduino. Chacun peut faire le câblage à sa manière mais le plus important est de déclarer les pins dans le programme. Les photos ci-dessous montrent comment nous avons fait les connexions des fils et placé les composants de notre robot mais sans le schéma structurel ce serait difficile de comprendre les connexions de composants.

II.11.1. Principe de fonctionnement du robot :

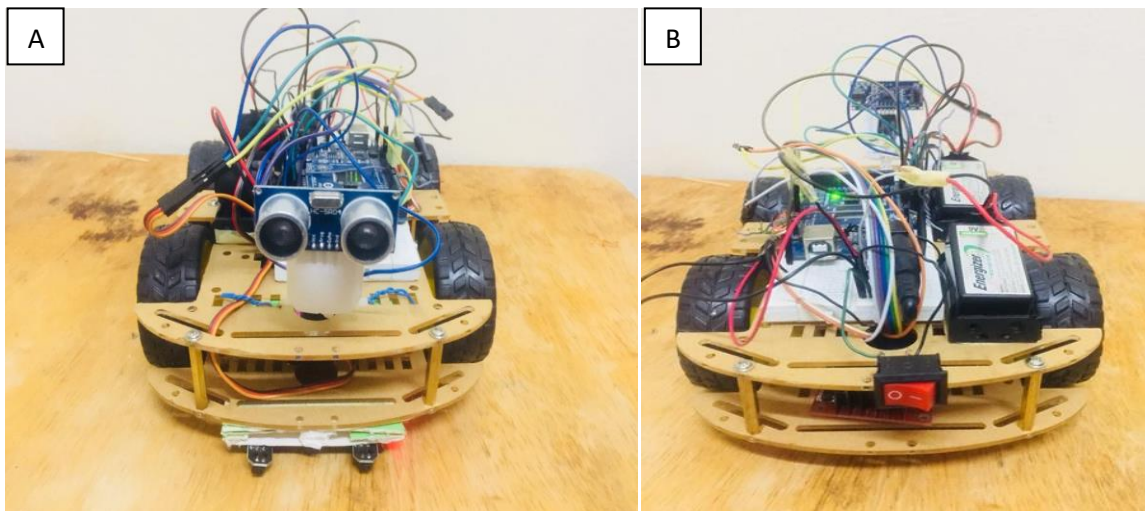
Détecteur d'obstacle : À l'état initial quand on appuie sur le bouton marche les moteurs sont actionnés et le servo-moteur est à 90° (angle) et le capteur à ultrason envoie des impulsions ultrasoniques avec une fréquence de 40 kHz via la sortie trigger qui est en état haut. Ces impulsions se propagent dans l'air à la vitesse du son, lorsqu'elles rencontrent un objet ou un obstacle devant lui à 20 cm elles se réfléchissent et reviennent sous forme d'écho au capteur et le robot s'arrête pendant quelques microsecondes et recule de quelques centimètres (selon le programme) en suite le servo-moteur fait un balayage de gauche (angle 0°) et à droite (angle 180°). Si il y a un obstacle à gauche il tourne à droite si non il tourne gauche et avance. Tant qu'il n'y a pas d'obstacle devant lui il continue d'avancer.

Suiveur de ligne : Deux capteurs à infrarouge sont placés sur la partie inférieure du robot à 20mm du sol. On place le robot sur une ligne noire de manière à ce que la ligne soit entre les capteurs et à l'état initial les moteurs sont actionnés le robot avance entre la ligne noire. Si le robot sort de la ligne du côté gauche le capteur de côté droit détecte la ligne noire les moteurs du côté droit s'arrêtent et ceux du côté gauche continuent de tourner jusqu'à ce que l'autre capteur sort de la ligne noire par la suite les moteurs de deux côtés continuent de tourner ainsi le robot avance en suivant cette ligne et inversement.

II.11.2. Caractéristiques et performance du montage :

Le montage est réussi le robot fonctionne correctement mais il y a trop de fils de connexion du coup ça pose un peu du souci, en plus nous avons constaté que le capteur à ultrason ne peut pas mesurer une distance avec exactitude sur un plan incliné. Pour remédier à ce problème il lui faut (robot) d'autres capteurs à ultrason chaque côté (droite et gauche) qui soit capable de mesurer sur une surface inclinée.

Sur la performance du robot ; l'alimentation qu'on utilise est insuffisante pour lui faire fonctionner correctement pendant longtemps, le robot peut fonctionner environ 10 minutes. Après ça les batteries se déchargent. Nous avons besoin des batteries qui ont une bonne autonomie. Sur les photos ci-dessous nous avons montré les vues de face (A), derrière (B), gauche (C) et droite (D) du robot.



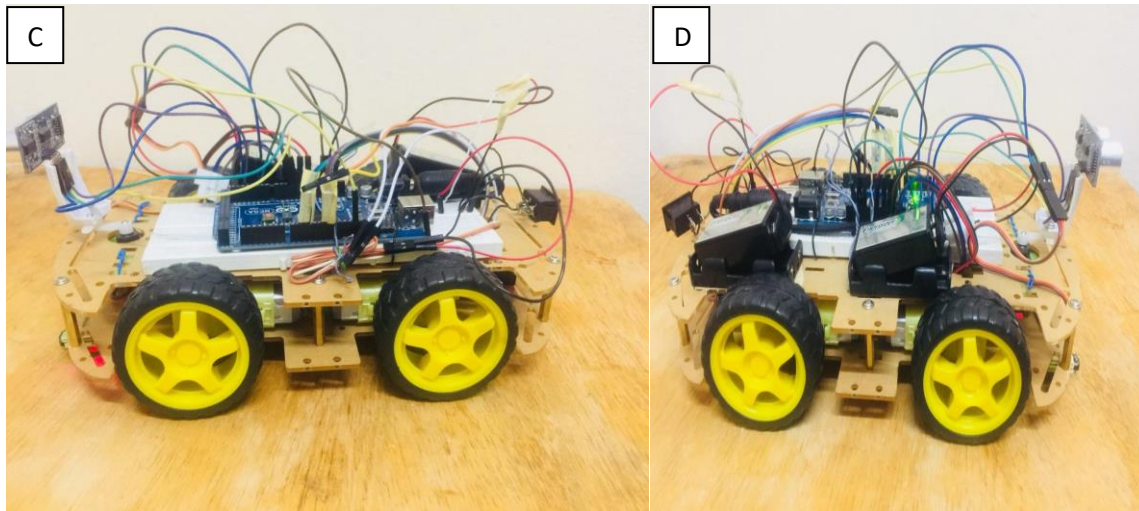


Figure II.35 : Le montage complet du robot

II.12. Perspectives d'évolution et d'amélioration du montage :

Dans notre étude il y a plusieurs perspectives qui sont envisageables pour le robot mobile détecteur d'obstacle. Il nous semble envisageable de prolonger ce projet en remplaçant l'Arduino par raspberry pi et intégrer une camera pour pouvoir suivre le robot. Mais malheureusement nous n'avons pu effectuer cette amélioration pour cause la covid-19. A l'avenir il est envisageable de réaliser une étude innovante sur le robot mobile suiveur d'un être humain.

Beaucoup de problèmes encore à résoudre pour atteindre une navigation robuste dans des espaces non structurés, en particulier en intérieur augmenter les connaissances en termes de localisation et de navigation de systèmes autonomes.

Capteur TCRT5000 : Compte tenu de l'interruption du projet pour cause la pandémie, on a quand même pas mal apporté des améliorations sur notre projet à savoir sur la programmation, l'ajout des diodes leds qui s'allument et indique s'il y a un obstacle ou pas. Mais avant cela on a rencontré pas mal de problèmes. Au début on a commencé par le robot suiveur de ligne les capteurs qu'on a utilisé premièrement on n'arrive pas à les faire fonctionner c'est qui à un peu trainer les choses, mais ça c'est amélioré en changeant les capteurs TCRT5000 avec potentiomètre ajustable aux capteurs TCRT5000 simple sans potentiomètre nouvelle model.

Détecteur d'obstacle : En suite le robot détecteur d'obstacle ce qui n'a pas été facile pour nous car on a demandé beaucoup d'aide au près des informaticiens concernant la programmation, au bout d'une semaine de recherche et de réflexion on y est arrivé à faire fonctionner correctement le robot.

Servo-moteur : Le fonctionnement du robot avec le servo-moteur MG90S. En réalité on a commencé par le servo-moteur SG90 mais quand on connecte le avec le robot, il n'arrive pas tourner d'un angle précis (angle précisé par le programme), le servo-moteur ne fait que tourner comme un simple moteur à courant continu quel que soit la programmation. Ça nous a tellement causé de difficulté ; nous avons demandé à un professeur qui nous a expliqué que le servo-moteur a été modifier, un servo-moteur est équipé d'un potentiomètre si on l'enlève ce dernier devient un simple moteur à courant continu. On a changé le servo-moteur est on a obtenu le résultat souhaité.

L'alimentation : Concernant l'alimentation premièrement on utilisait qu'une batterie de 9V, après quelques testes toutes les composantes du robot ne fonctionnaient pas correctement, on s'est rendu compte que c'est tellement insuffisant. Nous avons donc addition la tension nécessaire pour le bon fonctionnement notre robot et nous avons ajouté deux batteries supplémentaires de 9V.

II.13. Conclusion :

Dans ce dernier chapitre, nous avons expliqué clairement les différentes étapes qui nous ont permis de concevoir ce robot avec deux types de fonctionnalité : robot suiveur de ligne et détection d'obstacles (robot autonome). Nous avons présenté les différents composants utilisés et leurs tâches et présenté aussi la description générale de chaque pièce qui est connecté sur carte Arduino; dont le tout forme notre robot. Pour la réalisation de ce projet nous avons rencontré pas mal de difficultés avec les capteurs à infrarouge, le servo-moteur, l'alimentation et beaucoup plus le covid19 virus(confinement) qui nous a empêché de travailler ensemble et surtout au laboratoire, mais au final nous avons pu arranger, améliorer et réaliser ce projet.

Conclusion générale

Conclusion générale

Conclusion générale :

Nous avons réussi à réaliser l'objectif de notre projet qui est la réalisation d'un robot mobile autonome à l'aide de la carte électronique Arduino avec ses différents shields et son logiciel ; notre robot qui est un robot mobile a quatre roues, il détecte un obstacle ou présence et change de direction, aussi nous avons pu faire un robot suiveur de ligne. Tout d'abord nous avons étudié les principales caractéristiques d'un robot et en particulier le robot mobile.

Ainsi grâce aux enseignements que nous avons reçus durant notre cursus qui nous a permis de réaliser ce projet qui n'a pas été facile pour nous car nous avons rencontré beaucoup de difficulté. Tout en étant guidé mais à la fois libre, ce projet nous a permis de prendre des initiatives, d'effectuer un travail éducatif et des réfléchir par nous même.

La technologie de détection d'obstacle est de nos jours une idée qui se développe de jour en jour au niveau des usines de fabrication des voitures autonomes.

Sources et bibliographies

- [1] Takhi H, Attach R. conception et réalisation d'un robot mobile à base d'Arduino. [**en ligne**]. Electronique : Instrumentation. Université Amar Telidji-Laghout, 2014-2015. 88 p Disponible sur : <https://www.memoireonline.com/> (consulté le : 03/02/2020)
- [2] mémoire1 :Dahmani Fatima Zohra DamouAhlemthème : Commande et contrôle du bras manipulateur du ROB3. Mémoire de fin d'étude Master électronique instrumentation. Université de SAIDA Dr Moulay Tahar. 2018/2019. Pages 67
- [3] technolangevin12c3ABg3 ; Histoire des robots et des automates <https://sites.google.com/site/technolangevin12c3abg3/capteur> consulté le 8/9/2020
- [4] Jérôme Bonnin, La mesure du temps dans l'Antiquité, Paris, [Les Belles Lettres](#), 2015, 444 p. ISBN 978-2-251-44509-0.
- [5] Jacques de Vaucanson, Le mécanisme du fluteur automate présenté à messieurs de l'Académie royale des sciences, Paris, J. Guérin, 1738
- [6] technolangevin12c3ABg3 ; Histoire des robots et des automates <https://sites.google.com/site/technolangevin12c3abg3/capteur> consulté le 8/9/2020
- [7] Jean Marguin, Histoire des instruments et machines à calculer, trois siècles de mécanique pensante 1642-1942, [Hermann](#), 1994, 206 p
- [8] Karel Čapek (trad. Jan Rubes), *R.U.R.* : Rossum's Universal Robots, Éditions de La Différence, 2011, 220 p. ISBN 2729119221.
- [9] Futura tech ; les trois lois de la robotique <https://www.futura-sciences.com/tech/dossiers/robotique-trois-lois-robotique> consulté le 02/05/2020
- [10] LARAJTEKNO <https://larajtekno.info/aides-a-la-creation-des-programmes-arduino-augmente> consulté le 19/09/2020
- [11] LA Fabrique DIY <http://www.lafabriquedi.com/tutoriel/liste-des-capteurs-229/> , consulté le 20/09/2020
- [12] <http://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id> ; robot à structure sphérique Domaine : cybernétique Auteur : Office québécois de la langue française, 1998 Consulté le 23/09/2020
- [13] Mecatron ; rapport de projet de fin d'étude ; <http://mecatron.rma.ac.be/PDF/Students/islam.pdf> . Consulté le 23/09/2020
- [14] DavidFilliat. Robotique Mobile. Engineering school.Robotique Mobile, ENSTA ParisTech, 2011, pp.175. cel-00655005 HAL archives-ouvert.fr

- [15] Robot humanoïde ATLAS ; <https://sites.google.com/site/tpeia9/big-dog> . consulté le 02/08/2020
- [16] Saidur Rahman ; researchgate.
https://www.researchgate.net/publication/329686583_IoT_Based_Smart_Parking_System consulté le 17/06/2020
- [17] S.Poujouly IUT CACHAN / Module2-Capteur : Capteurs – présentation
- [18] Une machine autonome sur les champs de bataille ;
<https://sites.google.com/site/tpeia9/big-dog> . consulté le 02/08/2020
- [19] L'est éclair abonné ; <https://abonne.lest-eclair.fr/> . consulté le 20/09/2020
- [20] Des vidéos et images à partager ; <http://olalabuzz.blogspot.com/> ; consulté le 22/09/2020
- [21] Mercadolivre https://produto.mercadolivre.com.br/MLB-724218445-3-x-hc-sr04-sensor-ultrasonico-arduino-0127-_JM consulté le 17/06/2020
- [22] Tutoriel pour Arduino consulté le 15/05/2020
http://tiptopboards.free.fr/arduino_forum/viewtopic.php
- [23] Openclassroom, programmez vos premier montage avec Arduino,
<https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino/3285287-donnez-du-mouvement-a-vos-montages-avec-un-servo-moteur-et-la-fonction-switch> consulté le 17/05/2020
- [24] Guillaume Martin. académie-Grenoble : Algorithmique et Arduino Programmation d'une carte Arduino ; 17p consulté le 13/09/2020 Site : www.technobm.clg-gdm.fr
- [25] <https://techcrunch.com/2010/03/25/meet-robot-hanako-the-dental-patient-simulator> : techcrunch; hanako. Consulté le 23/09/2020
- [26] Pierre Lopez, Ahmed Saïd Nouri ; Théorie élémentaire et pratique de la commande par les régimes glissants, ISBN103-540-31003-7
- [27] Michel Brissaud ; Matériaux piézoélectriques: caractérisation, modélisation et vibration : ISBN978-2-88074-692-6,1er Edition,2007
- [28] Marcel Jufer ; Electromécanique,volume IX ; ISBN2-88074-285-4 à 1995, 1998, 2004
- [29] CHRISTIAN, Tavernier. Arduino : Maitrisez sa programmation et ses cartes d'interface (shields). EEA : Dunod, 2^e édition, 2014, 213p
- [30] Microsonic ; <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe>. Consulté le 10/03/2020.

[31] TamiaLab ; mesurer une distance avec un capteur à ultrason HC-SR04 et une carte Arduino/Genuino. publier le 21/02/2017

[32] Robert Chauprade, Commande électronique des moteurs à courant continu – À l'usage de l'enseignement supérieur, écoles d'ingénieurs, maîtrise, IUT, Paris, éd. [Eyrolles](#), coll. « EEA », 1975, 243 p.,

[33] Astuces pratiques ; moteur à courant continu : <https://www.astuces-pratiques.fr/electronique/le-moteur-a-courant-continu-principe> . consule le 6/09/2020

Programme de détecteur d'obstacle :

```
#include <Servo.h>      //Servo moteur bibliotheque
#include <NewPing.h>    // bibliotheque du capteur Ultrason .

// L298N pins de controle
// int S_A = 10; //speed motor a
const int LeftMotorForward = 3;
const int LeftMotorBackward = 2;
const int RightMotorForward = 6;
const int RightMotorBackward = 7;
// int S_B = 9; //speed motor b

int Ledrouge = 8;// le numero du pin de la Led est 2
int Bleu = 9;// le numero du pin de la Led est 4

//sensor pins
#define trig_pin A3 //entree analogique 1
#define echo_pin A4 //entree analogique 2

#define maximum_distance 500
boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance); // fonction capteur a Ultrason
Servo servo_motor; //on nomme notre servo moteur
```

```
void setup(){

    pinMode(RightMotorForward, OUTPUT);
    pinMode(LeftMotorForward, OUTPUT);
    pinMode(LeftMotorBackward, OUTPUT);
    pinMode(RightMotorBackward, OUTPUT);

    pinMode(Ledrouge, OUTPUT);//effectue la sortie digitale du pin:
    pinMode(Bleu, OUTPUT);//effectue la sortie digitale du pin:

    servo_motor.attach(11); // pin du servo moteur

    servo_motor.write(115);
    delay(2000);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    //analogWrite(S_A, 150);
    //analogWrite(S_B, 150);
    //delay(200);
}

void loop(){
```

```
int distanceRight = 0;

int distanceLeft = 0;

delay(50);

if (distance <= 20){
  digitalWrite(Ledrouge, HIGH);// allumage de IED Rouge
  digitalWrite(Bleu, LOW);
  moveStop();
  delay(300);
  moveBackward();
  delay(400);
  moveStop();
  delay(300);
  distanceRight = lookRight();
  delay(300);
  distanceLeft = lookLeft();
  delay(300);

  if (distance >= distanceLeft){
    turnRight();
    moveStop();
  }
  else{
    turnLeft();
    moveStop();
  }
}

else{
  moveForward();
```



```
}  
  
    distance = readPing();  
  
    digitalWrite(Bleu, HIGH);  
  
    digitalWrite(Ledrouge, LOW);  
  
}
```

```
int lookRight(){  
  
    servo_motor.write(50);  
  
    delay(500);  
  
    int distance = readPing();  
  
    delay(100);  
  
    servo_motor.write(115);  
  
    return distance;  
  
}
```

```
int lookLeft(){  
  
    servo_motor.write(170);  
  
    delay(500);  
  
    int distance = readPing();  
  
    delay(100);  
  
    servo_motor.write(115);  
  
    return distance;  
  
    delay(100);  
  
}
```

```
int readPing(){  
  
    delay(70);  
  
    int cm = sonar.ping_cm();
```

```
if (cm==0){  
    cm=350;  
}  
return cm;  
}
```

```
void moveStop(){
```

```
    digitalWrite(RightMotorForward, LOW);  
    digitalWrite(LeftMotorForward, LOW);  
    digitalWrite(RightMotorBackward, LOW);  
    digitalWrite(LeftMotorBackward, LOW);  
}
```

```
void moveForward(){
```

```
    if(!goesForward){  
  
        goesForward=true;  
  
        digitalWrite(LeftMotorForward, HIGH);  
        digitalWrite(RightMotorForward, HIGH);  
  
        digitalWrite(LeftMotorBackward, LOW);  
        digitalWrite(RightMotorBackward, LOW);  
    }  
}
```

```
void moveBackward(){
```

```
goesForward=false;
```

```
digitalWrite(LeftMotorBackward, HIGH);
```

```
digitalWrite(RightMotorBackward, HIGH);
```

```
digitalWrite(LeftMotorForward, LOW);
```

```
digitalWrite(RightMotorForward, LOW);
```

```
}
```

```
void turnRight(){
```

```
digitalWrite(LeftMotorForward, HIGH);
```

```
digitalWrite(RightMotorBackward, HIGH);
```

```
digitalWrite(LeftMotorBackward, LOW);
```

```
digitalWrite(RightMotorForward, LOW);
```

```
delay(500);
```

```
digitalWrite(LeftMotorForward, HIGH);
```

```
digitalWrite(RightMotorForward, HIGH);
```

```
digitalWrite(LeftMotorBackward, LOW);
```

```
digitalWrite(RightMotorBackward, LOW);
```

```
}
```

```
void turnLeft(){
```

```
    digitalWrite(LeftMotorBackward, HIGH);
```

```
    digitalWrite(RightMotorForward, HIGH);
```

```
    digitalWrite(LeftMotorForward, LOW);
```

```
    digitalWrite(RightMotorBackward, LOW);
```

```
    delay(500);
```

```
    digitalWrite(LeftMotorForward, HIGH);
```

```
    digitalWrite(RightMotorForward, HIGH);
```

```
    digitalWrite(LeftMotorBackward, LOW);
```

```
    digitalWrite(RightMotorBackward, LOW);
```

```
}
```

Programme de suiveur de ligne :

```
int S_A = 10; //speed motor a
```

```
int M_A1 = 2; //motor a = +
```

```
int M_A2 = 3; //motor a = -
```

```
int M_B1 = 6; //motor b = -
```

```
int M_B2 = 7; //motor b = +
```

```
int S_B = 9; //speed motor b
```

```
int R_S = A0; //sincer R
```

```
int L_S = A2; //sincer L
```

```
void setup()
```

```
{
```

```
pinMode(M_B1, OUTPUT);
```

```
pinMode(M_B2, OUTPUT);
```

```
pinMode(M_A1, OUTPUT);
```

```
pinMode(M_A2, OUTPUT);
```

```
pinMode(S_B, OUTPUT);
```

```
pinMode(S_A, OUTPUT);
```

```
pinMode(L_S, INPUT);
```

```
pinMode(R_S, INPUT);
```

```
analogWrite(S_A, 150);
```

```
analogWrite(S_B, 150);
```

```
delay(200);
```

```
}
```

```
void loop()
```

```
{  
if ((digitalRead(L_S) == 1)&&(digitalRead(R_S) == 1)){forward();}  
  
if ((digitalRead(L_S) == 0)&&(digitalRead(R_S) == 1)){turnLeft();}  
if ((digitalRead(L_S) == 0)&&(digitalRead(R_S) == 1)) {turnLeft();}  
  
if ((digitalRead(L_S) == 1)&&(digitalRead(R_S) == 0)){turnRight();}  
if ((digitalRead(L_S) == 1)&&(digitalRead(R_S) == 0)){turnRight();}  
  
if ((digitalRead(L_S) == 0)&&(digitalRead(R_S) == 0)){Stop();}  
}
```

```
void forward(){  
digitalWrite(M_A1, LOW);  
digitalWrite(M_A2, HIGH);  
digitalWrite(M_B1, HIGH);  
digitalWrite(M_B2, LOW);  
}
```

```
void turnRight(){  
digitalWrite(M_A1, LOW);  
digitalWrite(M_A2, LOW);  
digitalWrite(M_B1, HIGH);  
digitalWrite(M_B2, LOW);  
}
```

```
void turnLeft(){  
digitalWrite(M_A1, LOW);
```

```
digitalWrite(M_A2, HIGH);  
digitalWrite(M_B1, LOW);  
digitalWrite(M_B2, LOW);  
}
```

```
void Stop(){  
digitalWrite(M_A1, LOW);  
digitalWrite(M_A2, LOW);  
digitalWrite(M_B1, LOW);  
digitalWrite(M_B2, LOW);  
}
```