

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

Université Badji Mokhtar Annaba
Badji Mokhtar Annaba University



جامعة باجي مختار - عنابة

Faculté : Sciences de l'Ingénierat
Département : Electronique
Domaine : Sciences et Techniques
Filière : Electronique
Spécialité : automatique et informatique
industrielle

Mémoire

Présenté en vue de l'obtention du diplôme de : **MASTER**

Thème :

Commande d'un fauteuil roulant pour handicapé

Présenté par : LOUKIL Akram

Encadrant : REDJATI Abdelghani

Maitre de conférences B

UBM Annaba

Jury de Soutenance :

FEZARI M.	Professeur	UBM Annaba	Examineur
BENMOUSSA S.	Maître de conférences A	UBM Annaba	Président
REDJATI A.	Maître de conférences B	UBM Annaba	Encadrant

Année Universitaire : 2020/2021

Remerciment

*Je remercie en premier lieu ; Dieu tout puissant, le miséricordieux,
auquel je dois l'accomplissement de ce modeste travail.*

*Je tiens à remercier chaleureusement mon directeur de mémoire
Mr. REDJATI Abdelghani, pour avoir accepté de superviser ce travail,
pour m'avoir prodigué ses conseils et avoir accompagné mes
réflexions.*

*Je lui témoigne toute ma gratitude pour ses orientations, et
sa patience.*

*Mes remerciements vont également à chaque interlocuteur ayant
apporté sa pierre à l'édifice lors de l'élaboration de ce travail.*

Dédicaces

À cette source de tendresse, d'amour et de générosité, À mes parents ! Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Rien au monde ne vaudra les efforts consentis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de vos sacrifices.

À mes frères Med Badreddine, et A. Hakim, pour m'avoir toujours soutenu et guidé dans la bonne voie.

À ma sœur Loubna, et sa petite famille, pour leurs encouragements.

À tout le personnel et enseignants du département d'Electronique, pour tout le savoir qu'ils nous ont transmis.

À tous mes amis dans la famille universitaire et en dehors.

Résumé

Le projet présenté a pour but d'exploiter l'utilisation de commandes vocales pour l'aide à la navigation d'un fauteuil roulant électrique, comme alternative au joystick traditionnellement employé, tout en y intégrant un dispositif anticollision représenté par le capteur ultrason.

L'objectif est d'obtenir un système universel et ergonomique, pour assurer une certaine autonomie aux personnes handicapées en difficulté.

Le cœur de ce système est une carte Arduino Mega 2560 ; élément essentiel pour le contrôle du fauteuil roulant. Le but de nos travaux était d'aboutir à un ensemble fiable, peu coûteux, reproductible, sans danger, et d'utilisation aisée.

Mots clés : Fauteuil roulant électrique, Commande vocale, Capteur ultrason, Arduino, Bluetooth.

Abstract

The project aims to exploit the use of voice commands for the navigation assistance of an electric wheelchair, as an alternative to the joystick traditionally used, while integrating an anti-collision device represented by the ultrasonic sensor.

The objective is to obtain a universal and ergonomic system, to ensure a certain autonomy for struggling disabled people.

The core of this system is an Arduino Mega 2560 board; it is the main element to control the wheelchair, our goal was to achieve a reliable, inexpensive, reproducible, safe, and easy to use project set.

Keywords: Electric wheelchair, voice command, ultrasonic sensor, Arduino, Bluetooth.

ملخص

يهدف المشروع إلى استغلال استخدام الأوامر الصوتية للمساعدة في تسخير كرسي متحرك كهربائي كبديل لعصا التحكم المستخدمة تقليدياً ، مع دمج جهاز مضاد للتصادم يمثل جهاز استشعار الموجات فوق الصوتية الهدف هو الحصول على نظام عالمي ومريح ، لضمان درجة استقلالية معينة للأشخاص ذوي الإعاقة الذين يواجهون صعوبات

جوهر هذا النظام هو لوحة اردوينو ميغا 2560 التي تمثل العنصر الرئيسي للتحكم في الكرسي المتحرك وكان هدفنا هو تحقيق مجموعة مشاريع موثوقة، غير مكلفة، قابلة للتكرار، أمانة وسهولة الاستخدام

كلمات مفتاحية : كرسي متحرك كهربائي ، الأوامر الصوتية ، جهاز استشعار الموجات فوق الصوتية أردوينو ، بلوتوث

Liste des abréviations

DDL : Degrés De Liberté

FRE : Fauteuil Roulant Electrique.

IDE : Integrated Development Environment.

LED : Light Emitting Diode.

MCC : Moteur à courant continue.

MVFH : Modified Vector Field Histogram.

PWM : Pulse-Width Modulation.

SCARA : Selective Compliance Assembly Robot Arm.

TTL : Transistor-Transistor Logic.

UART : Universal Asynchronous Receiver-Transmitter.

VFH : Vector Field Histogram.

Liste des tableaux

Tableau 1 : Avantages et inconvénients de la robotique.....	07
Tableau 2 : Spécifications techniques de la carte Arduino Mega 2560 Rev3.....	17
Tableau 3 : Structures de contrôle.....	20
Tableau 4 : Fonctionnement du moteur DC en utilisant L293D.....	23
Tableau 5 : Brochage du L293D avec Arduino.....	37
Tableau 6 : Coût total du matériel utilisé.....	42

Liste des figures

Figure 1 : Différents types de robots manipulateurs.....	04
Figure 2 : Différents types de robots mobiles à roues.....	05
Figure 3 : Robots à chenilles.....	05
Figure 4 : Humanoïde (bipède), quadrupèdes, et hexapode.....	06
Figure 5 : Robot « snake-like »	06
Figure 6 : Constituants d'un FRE intelligent.....	09
Figure 7 : Les distances de détection d'obstacles à partir de la base d'un FRE....	13
Figure 8 : Carte Arduino Mega 2560.....	16
Figure 9 : Interface IDE Arduino.....	19
Figure 10 : Shield Moteur.....	21
Figure 11 : Différentes broches du composant L293D.....	21
Figure 12 : Raccordement des moteurs.....	22
Figure 13 : Capteur ultrason « HC-SR04 »	23
Figure 14 : Montage module Bluetooth HC-05.....	24
Figure 15 : Montage d'un servomoteur.....	24
Figure 16 : Redmi note 8.....	25
Figure 17 : Interface de l'application.....	26
Figure 18 : Kit de châssis utilisé.....	26
Figure 19 : Prototype mis en marche.....	28
Figure 20 : Schéma synoptique.....	29
Figure 21 : Schéma électronique du circuit réalisé.....	32
Figure 22 : Communication Bluetooth par terminal « arduino »	33

Figure 23 : Simulation de l'ultrason et arduino.....	34
Figure 24 : Utilisation du pont en H pour commander un MCC.....	36
Figure 25 : Commande du MCC.....	37
Figure 26 : Joystick analogique (DB15)	39
Figure 27 : Câblage des fils du joystick.....	39
Figure 28 : Brochage du joystick.....	40
Figure 29 : Joystick Axe "Y" Positive.....	40
Figure 30 : Joystick Axe "Y" negative.....	40
Figure 31 : Joystick Axe "X" Positive.....	41
Figure 32 : Joystick Axe "X" negative.....	41
Figure 33 : Valeurs obtenues.....	41

Liste des organigrammes

Organigramme 1 : Différents types de capteurs disponibles sur les FRE.....	12
Organigramme 2 : Chaîne de mesure.....	25
Organigramme 3 : Organigramme du fauteuil roulant.....	30
Organigramme 4 : Organigramme de la commande du bras.....	31
Organigramme 5 : Organigramme de détection d'obstacles.....	35
Organigramme 6 : Commande des moteurs.....	38

Sommaire

Remerciment

Dédicaces

Résumé

Liste des abréviations

Liste des figures

Liste des tableaux

Liste des organigrammes

Introduction générale.....	1
Chapitre 1 : Robotique.....	2
1. Définition.....	3
2. Historique	3
3. Evolution.....	4
4. Types de robots.....	4
4.1. Les robots mmanipulateurs.....	4
4.2. Les robots mobiles.....	4
4.2.1. Robots à roues	5
4.2.2. Robots à chenilles	5
4.2.3. Robots marcheurs.....	6
4.2.4. Robots ramplants.....	6
5. Domaines d'utilisation	7
6. Avantages et inconvénients.....	7
7. Conclusion.....	7
Chapitre 2 : FRE intelligents.....	8
1. Généralités.....	9
2. Cahier des charges.....	10
3. Classification.....	10
3.1. La Structure mécanique.....	10
3.2. Les interfaces de commande.....	10
3.3. Les capteurs.....	11
3.4. La commande.....	11
3.5. Les modes opératoires.....	11

3.6.	Les balises et les cartes internes de l'environnement.....	12
4.	Algorithme de detection d'obstacles.....	13
5.	Conclusion.....	14
	Chapitre 3 : Architecture et specifications.....	15
1.	Introduction.....	16
2.	Arduino.....	16
2.1.	Carte Arduino Atmega2560.....	16
2.2.	Software.....	18
2.2.1.	IDE Arduino	18
2.2.2.	Language Arduino	20
3.	Shield Moteur.....	21
4.	Capteur à ultrason HC-SR04	23
5.	Module Bluetooth HC-05.....	23
6.	Servomoteur	24
7.	Les capteurs	25
8.	Smartphone Android.....	25
9.	Kit de châssis	26
10.	Conclusion.....	26
	Chapitre 4 : Réalisation du projet.....	27
1.	Introduction.....	28
2.	Fonctionnement du projet.....	28
3.	Schema synoptique.....	29
4.	Organigrammes.....	30
5.	Schéma électrique.....	32
6.	Configuration du Bluetooth.....	32
7.	Utilisation de l'ultrason.....	33
8.	Le pont en H.....	36
9.	Variation de la Vitesse du MCC.....	36
10.	Adaptation du joystick à l'arduino.....	39
11.	Cout du matériel.....	42
12.	Conclusion.....	42
	Conclusion générale.....	43
	ANNEXES.....	44

Introduction générale

Un nombre conséquent de personnes invalides (1 personne sur 200 de nos jours) ne peuvent pas se servir d'un fauteuil électrique ordinaire ou le conduisent avec difficulté. Pour permettre un minimum d'autonomie, des chercheurs se sont penchés depuis plus de vingt ans sur cette problématique, en améliorant le contrôle manuel de l'utilisateur sur le fauteuil, ou en adaptant un système de commandes annexes.

Les progrès concernant les **fauteuils roulants électriques** (FRE) « intelligents », n'ont pas apporté les bénéfices espérés pour les personnes en situation de handicap : seules quelques fonctionnalités (suivi de ligne, détection d'obstacles) sont actuellement commercialisées, sans être à portée de tous.

La problématique abordée dans ce mémoire consiste à proposer une fonctionnalité d'aide au pilotage et de faire ainsi un pas vers le transfert technologique de fonctions de mobilité autonome plus complexes. Cette solution consiste à développer une gamme évoluée de commande vocale à l'aide de Bluetooth, à base d'Arduino pour la conduite des fauteuils électriques.

Ce mémoire s'attache à la définition de ce mode de commande, et à prouver l'intérêt de son utilisation.

Le premier chapitre présente des généralités sur la robotique.

Le second chapitre est consacré aux fauteuils roulants électriques.

Le troisième chapitre évoque le matériel utilisé dans le présent travail.

Quant au dernier chapitre, il relate les différentes phases de la réalisation du projet.

Et enfin, une conclusion générale et perspectives.

CHAPITRE 1 :

Robotique

1.1. Définition :

D'après l'association Française de Normalisation :

« Un robot est un manipulateur commandé en position, reprogrammable, polyvalent, à plusieurs degrés de libertés, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un ou plusieurs bras se terminant par un poignet. Son unité de commande utilise, notamment un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel ». [1]

1.2. Historique :

1920 : Apparition du mot robot L'origine du mot robot provient de la langue tchèque dans laquelle son ancêtre "robota" signifie travail forcé. Il a été introduit, par l'écrivain Karel Capek dans la pièce de théâtre Rossum's Universal Robots.

1942 : Isaac Asimov définit dans son œuvre de science-fiction les trois lois célèbres de la robotique

1950 : Article fondateur d'Alan Turing "Computing Machinery and Intelligence"

1954 : Premier robot programmable.

1956 : John McCarthy définit le terme "intelligence artificielle"

1961 : Utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une chaîne de montage de General Motors.

1978 : Le robot ARGOS. Développé à l'Université Paul Sabatier de Toulouse, et équipé d'un système de vision au fur et à mesure de ses déplacements.

1981 : Le robot VESA. Construit à l'I.N.S.A de Rennes, est équipé d'un arceau de sécurité pour réaliser la détection d'obstacles dans un environnement totalement inconnu.

1984 : Le robot FLAKEY, est le reflet des améliorations apportées par 14 années de développement. Avec une vitesse maximale de 66 cm/s, capable de naviguer dans des environnements réels.

🚦 Les robots mobiles actuels sont destinés à des applications hautement spécialisées, comme l'exploration sous-marine, les robots volants et le micro robot.

1.3. Evolution :

On peut distinguer 3 évolutions de robots :

-Les **Automates** : sont généralement programmés à l'avance et permettent d'effectuer des actions répétitives.

- Les Robots équipés de **capteurs** : de température, photoélectronique, ou à ultrasons. Ces capteurs vont permettre de prendre en compte des paramètres aléatoires de l'environnement qui n'aurait pu être envisagés lors de leur programmation initiale. Ces robots sont donc bien plus autonomes que les automates.

- Les robots dotés d'une IA : ceux disposant d'une intelligence dite "artificielle" et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones. En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme l'être humain.

1.4. Types de robots :

1.4.1. Les robots manipulateurs :

Un robot manipulateur se présente sous forme d'un bras doté d'un certain nombre de segments articulés. Il est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont utilisés dans des applications industrielles pour effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage. Et sont répartis en cinq grandes catégories : cylindriques, rectilignes, sphériques, articulés, et SCARA

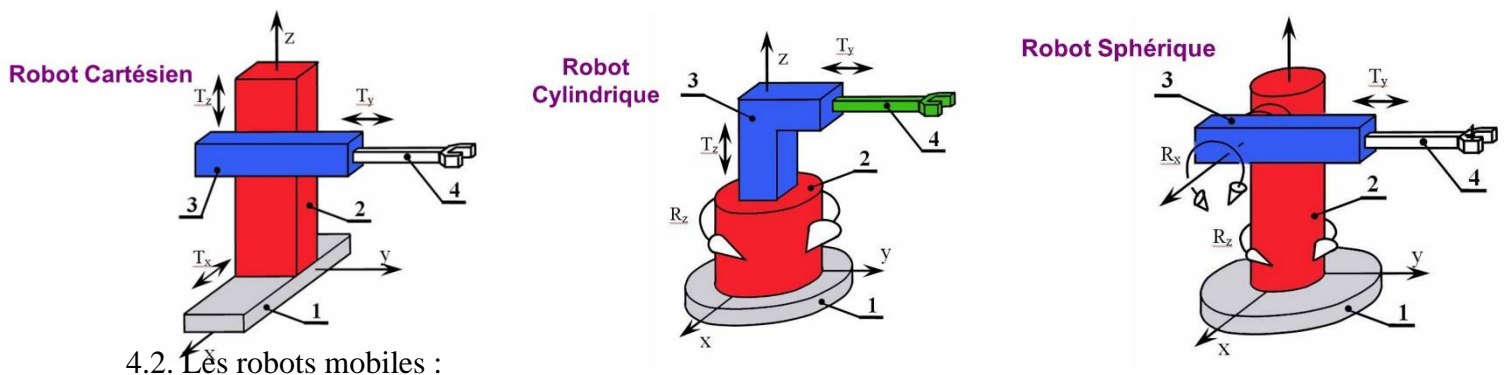


Fig 1 : Différents types de robots manipulateurs

Un robot mobile est celui qui peut se déplacer dans son environnement de façon autonome. Pour ce faire, le robot doit pouvoir naviguer avec une précision qui varie en fonction de la taille du robot et du type de la tâche à réaliser, selon le système de locomotion on peut distinguer :

1.4.2.1. Robots à roues : La mobilité par roues est la structure mécanique la plus utilisée. Ce type de robot assure un déplacement avec une accélération et une vitesse rapide mais nécessite un sol relativement plat. On distingue plusieurs classes de robots à roues, déterminées principalement par la position et le nombre de roues utilisées :

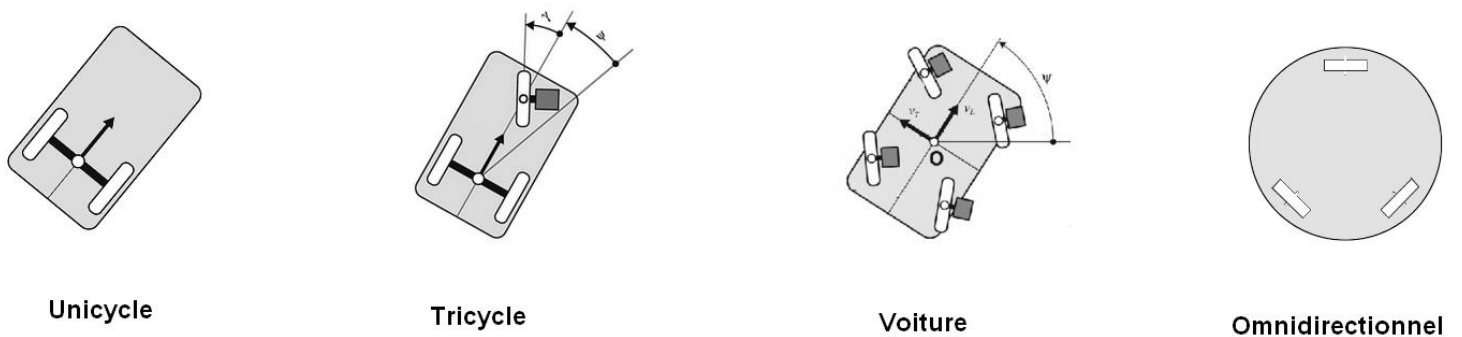


Fig 2 : Différents types de robots mobiles à roues

1.4.2.2. Robots à chenilles : présentent l'avantage d'une bonne adhérence au sol et d'une faculté de franchissement d'obstacles. L'utilisation de ces robots est orientée vers l'emploi sur sol accidenté ou de mauvaise qualité au niveau de l'adhérence (présence de boue, herbe...)

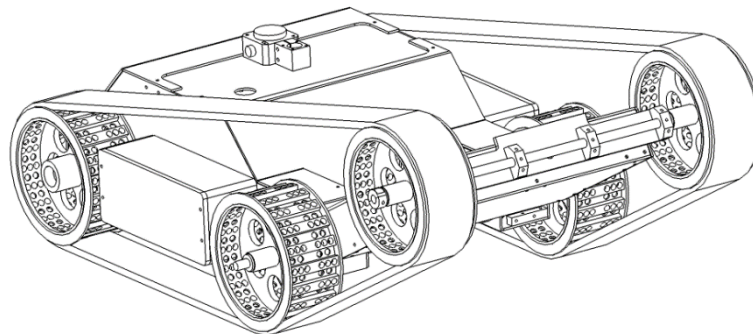


Fig 3 : Robot à chenilles

1.4.2.3. Robots marcheurs : sont destinés à réaliser des tâches variées dont l'accès à un site est difficile et dangereux pour l'être humain. Leur structure est dotée de plusieurs degrés de liberté.

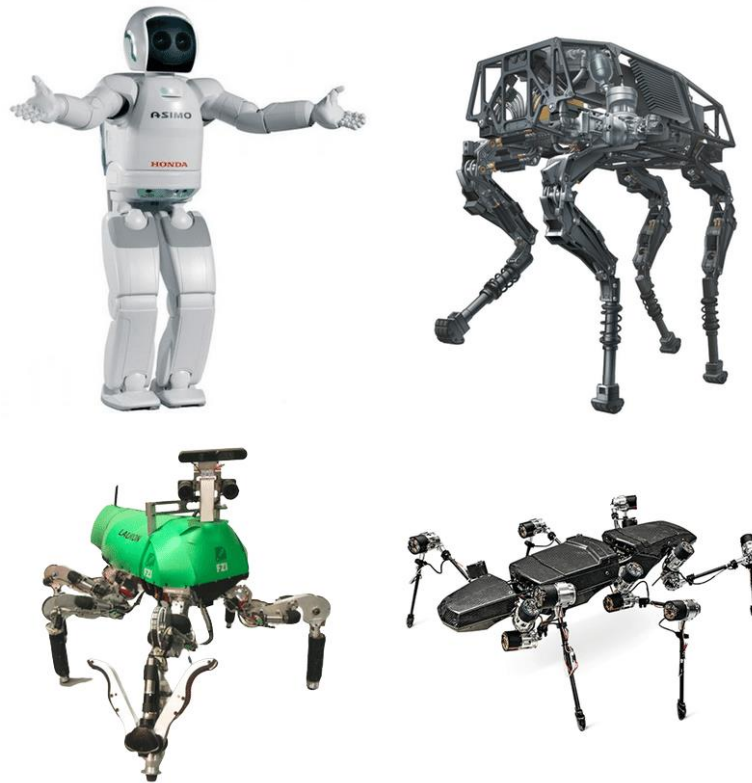


Fig 4 : Humanoïde (bipède), quadrupèdes, et hexapode

1.4.2.4. Robots rampants : Les applications de ce type de robots sont très spécialisées et les architectures des robots sont en général spécifiques à l'application visée, c'est une locomotion pour un environnement de type « tunnel » qui conduit à réaliser des structures filiformes, et qui découle des méthodes de locomotion des animaux et des insectes.

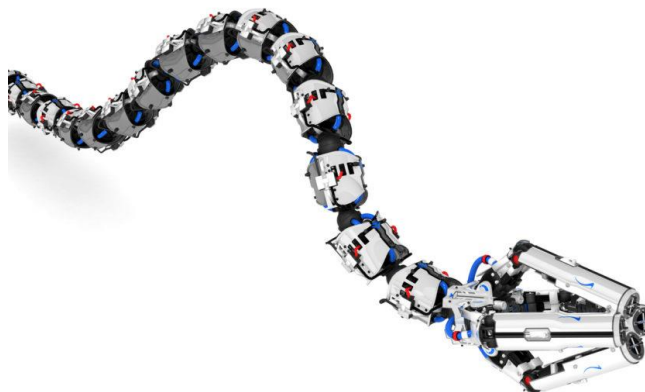


Fig 5 : Robot « snake-like »

1.5. Domaines d'utilisation :

- Les robots militaires : sont des robots principalement utilisés pour la surveillance aussi bien dans les airs que dans la mer. Les drones sont une sous-classe.
- Les robots explorateurs : ils permettent de par leurs tailles ou leurs capacités particulières d'explorer de nouveaux environnements. Cela permet parfois de limiter les pertes humaines, par exemple les naufrages de bateaux, l'exploration de territoires ennemis, ou encore des reliefs terrestres inconnus ou même des autres planètes.
- Les robots industriels : sont 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.
- Robots domestiques ou de service : utilisés à la maison. Ce type de robots comprend de nombreux appareils très différents, tel que les aspirateurs robotiques, robots nettoyeurs de piscines, balayeuses, nettoyeurs gouttières et autres robots qui peuvent faire différentes tâches.
- Robots en médecine et chirurgie : sont utilisés dans le cadre d'une application thérapeutique, par exemple lors d'une chirurgie ou au cours d'un programme de réhabilitation neuromotrice. Du fait des contraintes importantes en matière de sécurité, leur niveau d'autonomie est faible.

1.6. Avantages et inconvénients :

Avantages	Inconvénients
<ul style="list-style-type: none">- Productivité, sécurité, efficacité, et qualité, accrues des produits.- Pas de fatigue/vacance/assurance médicale.- Précision linéaire de 10 à 20 microns.- Perfectionnement continu.	<ul style="list-style-type: none">- Coût élevé, et nécessité de programmation.- Peur de la perte d'emploi pour les humains- Consommation d'énergie- Manque de réactivité/prise de décision.- Capacités encore limitées.

Tab 1 : Avantages et inconvénients de la robotique

1.7. Conclusion :

A travers ce chapitre nous avons présenté une introduction à la robotique. Nous avons pu constater l'évolution et la diversité des robots existants.

Le chapitre suivant s'intéresse plus en détails à la présentation des fauteuils roulants.

CHAPITRE 2 :

Fauteuils roulants

électriques

« intelligents »

2.1. Généralités :

Le fauteuil roulant électrique (FRE) est un élément essentiel pour l'autonomie des personnes handicapées. Mais pour certains, l'utilisation peut parfois être difficile, voire impossible, du fait de capacités physiques résiduelles trop faibles ou de troubles cognitifs.

L'application des méthodes issues de la robotique mobile (de service) à des fauteuils roulants électriques ou manuels a donné naissance à un nouveau type de fauteuils dits « intelligents » (smart wheelchairs).

Dans ce chapitre, nous allons nous intéresser tout d'abord au cahier des charges et à la classification des FRE intelligents qui constituent la machine dans notre système coopératif. Nous étudierons ensuite en bref les parties software et hardware du FRE intelligent.



Fig 6 : Constituants d'un FRE intelligent

2.2. Cahier des charges :

- Détecter des obstacles : Le module anti-collision devra être capable de détecter des obstacles (murs, radiateurs...etc) à courte distance et se fera presque exclusivement en intérieur.
- Limiter la vitesse du fauteuil : Plus l'obstacle sera proche, plus la vitesse diminuera jusqu'à provoquer l'arrêt du FRE.
- Garantir la discrétion : Le système devra être discret, non encombrant et ne devra pas comporter de fils susceptibles d'être arrachés.
- Prévenir le blocage du fauteuil : Le système doit permettre le déplacement libre sans blocage dans des zones étroites (prévoir un système By-pass afin de reprendre la main sur le fauteuil).
- Sauvegarder la garantie du fauteuil : Le système anti-collision ne devra pas empêcher le bon fonctionnement de la garantie du FRE (pas de modification de l'électronique existante).

2.3. Classification :

L'« intelligence » d'un fauteuil peut être définie comme la capacité de percevoir son environnement extérieur et d'en tirer des informations pertinentes dans l'objectif de réaliser des mouvements autonomes ou semi-autonomes : évitement d'obstacle, passage de porte, accostage, suivi automatique de parcours. Les fauteuils roulants intelligents ont vu le jour au début des années 1980, et continuent aujourd'hui à être l'objet d'attention des chercheurs.

Nous pouvons les classer selon plusieurs critères :

2.3.1. La structure mécanique

Les premiers prototypes de fauteuils intelligents étaient des robots mobiles dotés d'une chaise, actuellement ils sont plutôt construits à partir de fauteuils roulants électriques commercialisés, et plus rarement, il s'agit de modules qui peuvent être ajoutés aux fauteuils roulants standards.

2.3.2. Les interfaces de commande

L'interface la plus classique est le joystick qui peut prendre de multiples formes (mini-joystick, joystick au menton...).

La commande vocale est utilisée sur plusieurs projets, son coût reste abordable et les logiciels de reconnaissance vocale sont disponibles.

L'oculométrie a l'avantage de ne nécessiter que le contrôle des muscles oculaires, mais présente certaines contraintes comme la fatigue.

2.3.3. Les Capteurs

Afin d'éviter d'éventuels obstacles ou de se localiser les fauteuils roulants ont besoin de capteurs de perception de l'environnement.

-Les plus fréquemment utilisés sont les télémètres à ultrasons : en mesurant le temps entre l'envoi de l'onde et la réception de son écho, nous pouvons calculer la distance qui sépare le capteur de l'obstacle. Ces capteurs rencontrent de nombreuses limitations, en relation avec la largeur du faisceau, la lecture rapide de données, et la perception des surfaces.

-Il existe un autre type de capteurs « infrarouges » avec une portée moins grande mais une meilleure précision, une faible consommation et un taux d'acquisition constant.

-Les capteurs laser ont une grande précision avec un coût et une consommation d'énergie élevés.

-Les capteurs les plus prometteurs sont les systèmes de vision, à l'aide de caméras pour la détection de l'environnement, et/ou l'identification des commandes données par l'utilisateur.

Plusieurs fauteuils utilisent une combinaison de deux types de capteurs ou plus pour compenser les limites des uns et des autres.

2.3.4. La Commande

Il s'agit de l'ensemble des méthodologies et de leur organisation permettant d'atteindre un certain objectif de mobilité. Certaines équipes de recherche ont développé des algorithmes spécifiques (VFH et MVFH), pour traiter les différentes lectures des capteurs ultrasons et le calcul de la commande en fonction des consignes de l'utilisateur. D'autres projets sont basés sur le concept comportemental. Les divers comportements primitifs, sont combinés pour produire d'autres comportements plus sophistiqués.

2.3.5. Les modes opératoires

-Quelques fauteuils roulants électriques intelligents fonctionnent comme des robots autonomes ; l'utilisateur donne au fauteuil un but qui peut être un emplacement géographique (la cuisine) ou

un objectif (aller téléphoner) ; le fauteuil réalise cette demande sans autre intervention humaine (exemple : TetraNauta, le fauteuil de l'Université de Kanazawa).

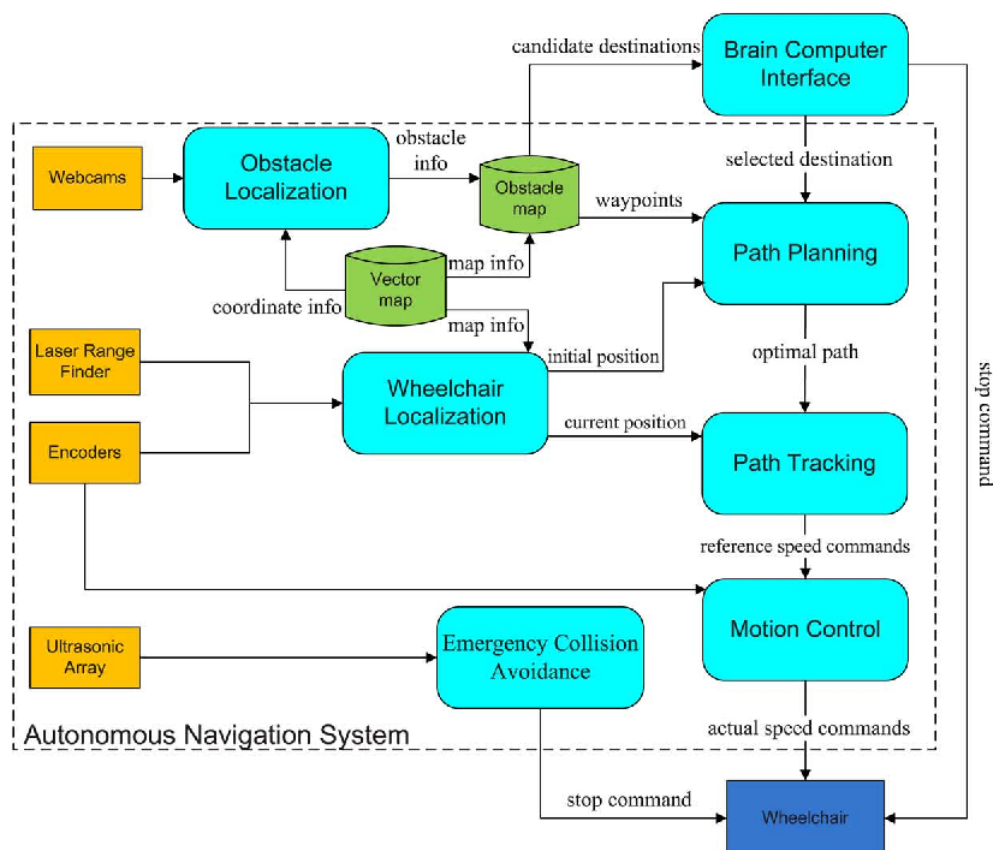
-D'autres fauteuils intelligents partagent la commande avec l'utilisateur. Ils s'occupent entièrement de quelques fonctions primitives comme l'évitement d'obstacles, et lui laissent d'autres tâches comme le choix des directions (exemple : NavChair et le TinMan).

-Un dernier groupe de fauteuils est celui qui offre à la fois la navigation autonome et semi-autonome ; le choix de ces modes peut être automatique ou manuel. A titre d'exemple, le VAHM offre cinq fonctions : (1) suivi de mur ; (2) suivi de trajectoires ; (3) suivi de direction ; (4) évitement d'obstacles ; (5) back tracking.

2.3.6. Les balises et les cartes internes de l'environnement

Les fauteuils roulants qui utilisent la navigation automatique disposent en général d'une carte interne de l'environnement ; cette carte peut encoder les distances dans l'environnement ou se limiter à donner des relations entre la position actuelle du fauteuil et quelques objets autour.

Certains fauteuils ne sont pas équipés de carte interne et utilisent donc des balises placées dans l'environnement (Computer Controlled Power Wheelchair Navigation System).



2.4) Algorithme de détection d'obstacles :

L'évitement des obstacles grâce à une décélération appropriée du fauteuil, se fait selon les distances suivantes :

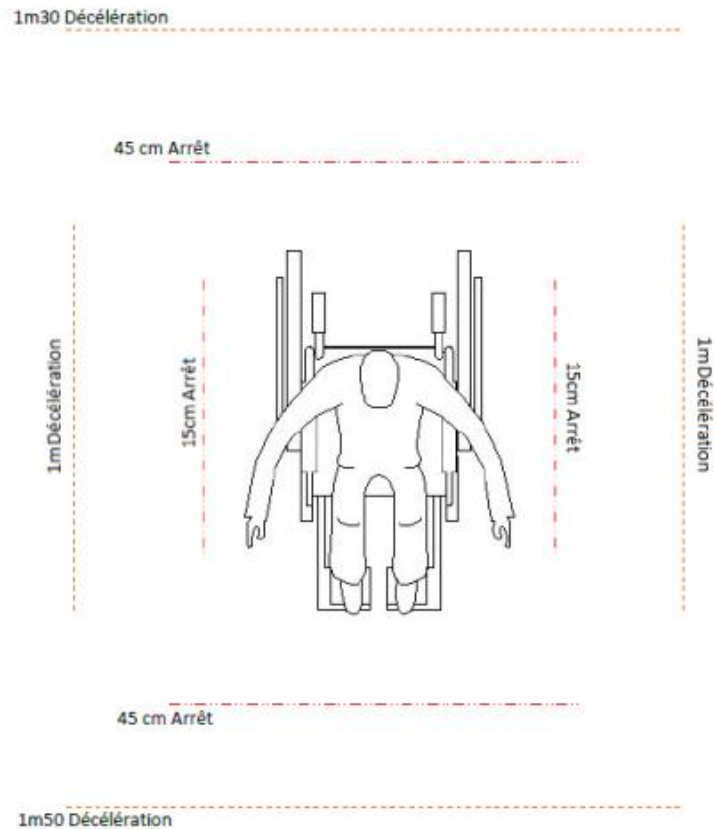


Fig 7 : Les distances de détection d'obstacles à partir de la base d'un FRE

Les principes sont :

- Scan en continu de la valeur des distances renvoyées par les capteurs ;
- Scan en continu la position du joystick (valeurs analogiques droite/gauche et avant/arrière)
- Si un obstacle est détecté comme étant proche et que le joystick est dans la direction de l'obstacle : la vitesse est limitée dans cette direction, jusqu'à l'arrêt (obstacle est trop proche).
- S'il n'y a pas d'obstacle dans la direction du joystick, les entrées analogiques sont recopiées dans les sorties PWM : le fauteuil se déplace normalement.

5) Conclusion

Malgré le grand nombre de recherches effectuées sur les fauteuils roulants électriques, très peu de prototypes sont commercialisés (accessibles), plusieurs explications peuvent être invoquées.

La première raison est sans doute technique : la fiabilité, en particulier des imperfections des capteurs d'environnement, n'est pas toujours suffisante, en particulier dans un environnement non-structuré comme l'est a priori l'espace de vie d'une personne handicapée.

Une seconde raison est sans doute la difficulté à dupliquer les prototypes, et surtout de convaincre de leur utilité les industriels, voire les utilisateurs potentiels eux-mêmes.

CHAPITRE 3 :

Architecture et Spécifications

3.1. Introduction :

Dans ce chapitre nous allons présenter le matériel utilisé dans notre projet, tout en donnant ses différentes spécifications et caractéristiques. Nous donnerons aussi, un aperçu sur les composants et logiciels qui ont pu aboutir à la mise en œuvre de notre système.

3.2. Arduino :

C'est une plate-forme open source de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation, qu'on considère comme pont tendu entre le monde réel et le monde numérique, en effet il permet d'étendre les capacités de relations humain/machine ou environnement/machine.

Sous ce nom Arduino se cachent non seulement du matériel mais aussi un logiciel. On parle alors de Physical Computing,

3.2.1. Carte Arduino Atmega2560 :

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM (Power Width Modulation (ou Modulation de Largeur d'Impulsion 'MLI'), 16 entrées analogiques, 4 UARTs, un bouton reset et un connecteur ICSP Elle est idéale pour des applications exigeant des caractéristiques plus complexes. Elle dispose d'une mémoire flash de 256 KB, dont 8 KB dévolus au Bootloader qui permet de télécharger les programmes entre l'IDE Arduino (interface de développement) et l'Arduino; également la possibilité de pouvoir exécuter le programme lors du démarrage de l'Arduino.

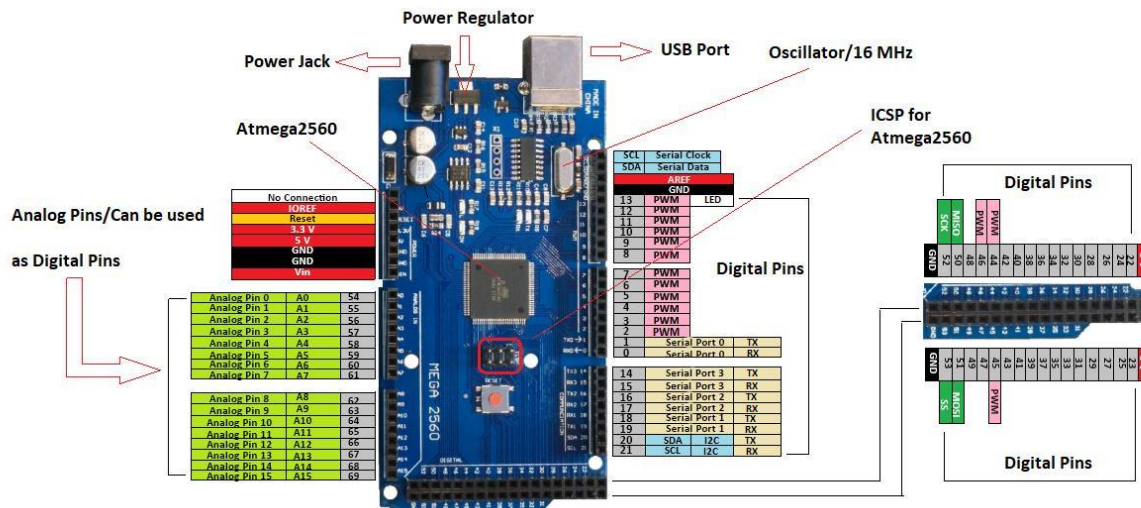


Fig 8 : Carte Arduino Mega 2560

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 Ma
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB sont utilisés par le Bootloader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 MHz

Tab 2 : Spécifications techniques de la carte Arduino Mega 2560 Rev3

Les broches d'alimentation pour la carte Arduino Mega

VIN : La tension d'entrée de la carte Arduino lorsqu'elle utilise une source d'alimentation externe (par opposition à 5 volts provenant de la connexion USB ou d'une autre source d'alimentation régulée). Vous pouvez fournir la tension à travers cette broche

5V : C'est l'alimentation régulée utilisée pour alimenter le microcontrôleur et les autres composants de la carte. Cela peut provenir de VIN via un régulateur embarqué, ou être fourni par l'USB ou toute autre alimentation 5V régulée.

V3 : Une alimentation de 3,3 volts générée par la puce FTDI embarquée. La consommation de courant maximale est de 50 mA.

GND : Masse de la carte.

Brochage de la carte Arduino Mega 2560

Broches numériques : 54 broches numériques sur le Mega peut être utilisé comme une entrée ou une sortie, en utilisant `pinMode ()`, `digitalWrite ()` et les fonctions `digitalRead ()`. Elles fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir un maximum de 40 mA et a une résistance de pull-up interne (déconnecté par défaut) de 20-50 kOhms.

En outre, certaines broches ont des fonctions spécialisées:

Serial : 0 (RX) et 1 (TX), série 1: 19 (RX) et 18 (TX), série 2: 17 (RX) et 16 (TX), série 3: 15 (RX) et 14(TX) . Permet de recevoir (RX) et transmettre (TX) des données du port série TTL. Broches 0 et 1 sont également reliés à des broches correspondantes du FTDI USB-TTL puce série.

Interruptions externes : 2 (interruption 0), 3 (alarme 1), 18 (interruption 5), 19 (interruption 4), 20 (interruption 3) et 21 (interruption 2). Ces broches peuvent être configurées pour déclencher une interruption sur une valeur faible, un front montant ou descendant, ou un changement de valeur.

PWM : 2 à 13 et 44 à 46. Fournir 8 bits sortie PWM avec la fonction `analogWrite ()`.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI, qui, bien que fourni par le matériel sous-jacent, n'est pas actuellement inclus dans le langage Arduino.

LED 13 : Il est équipé d'une LED connectée à la broche numérique 13. Lorsque la broche est haute valeur, la LED est allumée, lorsque la broche est faible, elle est éteinte.

I2C : 20 (SDA) et 21 (SCL). Soutenir la communication I2C (TWI) en utilisant la bibliothèque de fil.

Broches analogiques : La carte dispose de 16 entrées analogiques, chacun qui fournissent 10 bits de résolution (c. 1024valeurs différentes). Par défaut, ils mesurent à partir de la masse à 5 volts, mais est-il possible de changer l'extrémité supérieure de sa gamme en utilisant la broche AREF et la fonction `analogReference ()`. [2]

3.2.2. Software :

IDE Arduino :

Un IDE (environnement de développement) libre et gratuit est distribué sur le site officiel d'Arduino (compatible Windows, Linux et Mac).

D'autres alternatives existent pour développer l'Arduino (Visual Studio, Eclipse, XCode, etc.)

L'interface de l'IDE Arduino est plutôt simple, avec une interface minimale et épurée, un éditeur de code avec coloration syntaxique, et d'une barre d'outils rapide.

On retrouve aussi une barre de menus, et enfin, une console affichant les résultats de la compilation du code source.

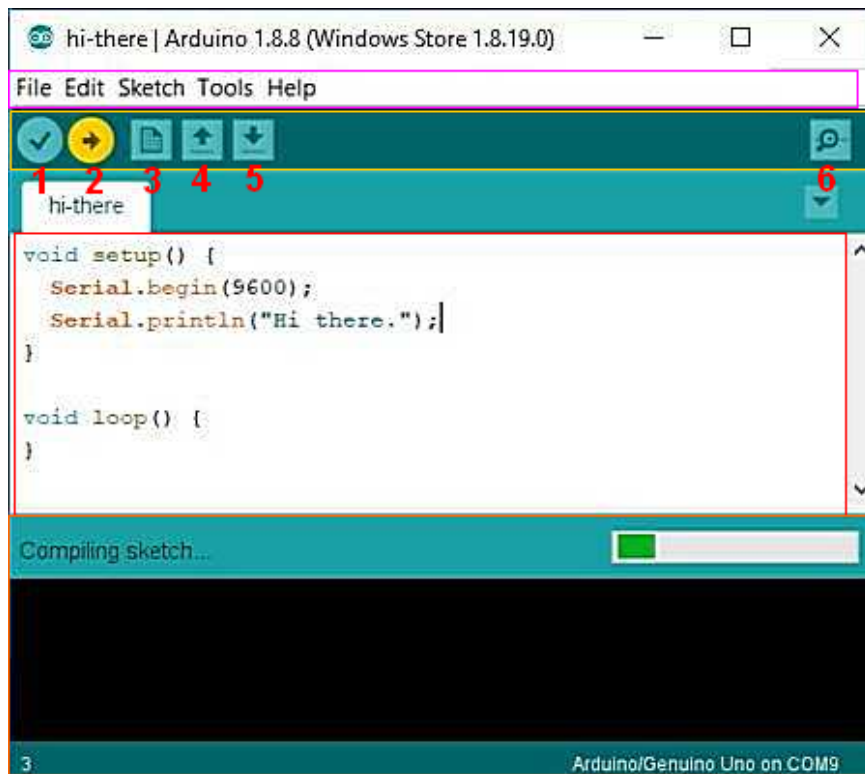


Fig 9 : Interface IDE Arduino

Bouton 1 (verify): Actionne un module qui cherche les erreurs dans le programme.

Bouton 2 (upload): Compiler et envoyer le programme vers la carte.

Bouton 3 (new): Créer un nouveau fichier.

Bouton 4 (open): Charger un programme existant.

Bouton 5 (save): Sauvegarder le programme en cours.

Bouton 6 (serial monitor) : Permet d'accéder au port série (en RX/TX).

Langage Arduino

C'est un langage inspiré de plusieurs autres langages. On retrouve notamment des similarités avec le C, le C++, et le Java.

Il impose une structure particulière typique de l'informatique embarquée :

-La fonction contient toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.).

-La fonction loop est exécutée en boucle après l'exécution de la fonction setup. Elle est absolument nécessaire sur les microcontrôleurs étant donné qu'ils n'ont pas de système d'exploitation, pour qu'il soit ainsi possible de reprendre la main sur la carte Arduino.

Nom	Utilité	Syntaxe
If-else	Condition logique	<pre>If (<valeur booléenne>) { <instruction> } else { <instruction> }</pre>
If-else if - else	Condition logique multiples	<pre>If (<valeur booléenne>) { <instruction> } else if (<valeur booléenne>) { <instruction> } else { <instruction> }</pre>
Switch	Sélecteur	<pre>Switch (<variable>) { case <valeur> : <instruction> break ; default : <instruction> }</pre>
While	Boucle	<pre>While (<valeur booléenne>) { <instruction> }</pre>
For	Boucle itérative	<pre>For (<initialisation> ; <valeur booléenne> ; <évolution>) { <instruction> }</pre>

Tab 3 : Structures de contrôle

3.3. Shield moteur

C'est un moteur facile à utiliser pour l'Arduino, capable de piloter différents moteurs.

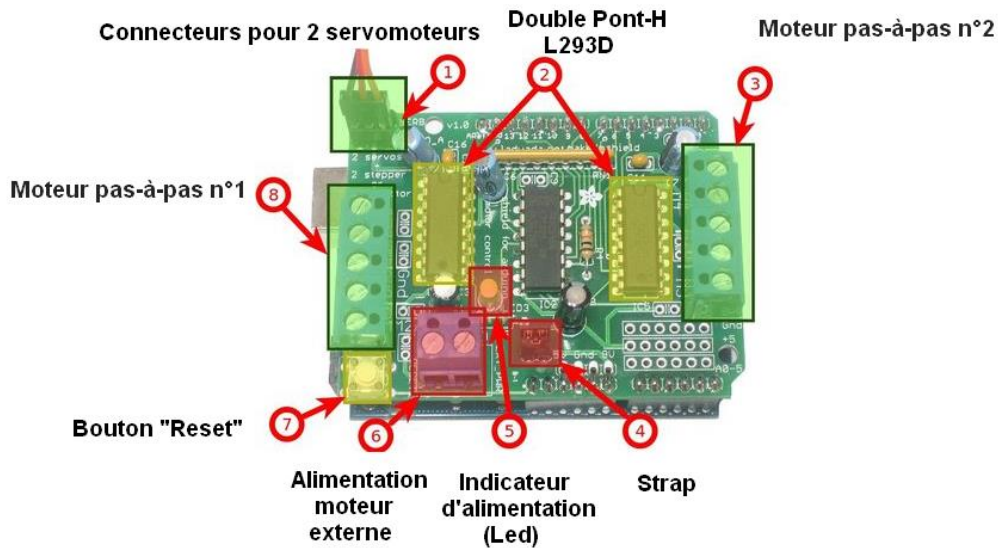


Fig 10 : Shield Moteur

Circuit L293D

Le L293D est un double pont-H, il est utilisé pour commander la rotation de deux moteurs dans les deux sens en raccordant les sorties de façon appropriées, Le L293 est un circuit intégré monolithique, à haut voltage, grand courant et 4 canaux ; cela veut dire que ce circuit intégré peut être utilisé pour des moteurs DC alimentés jusqu'à 36 volts et il est capable de piloter une paire de moteurs à courant continu ou un moteur pas à pas. Le circuit peut fournir un maximum de 600 mA par canal.

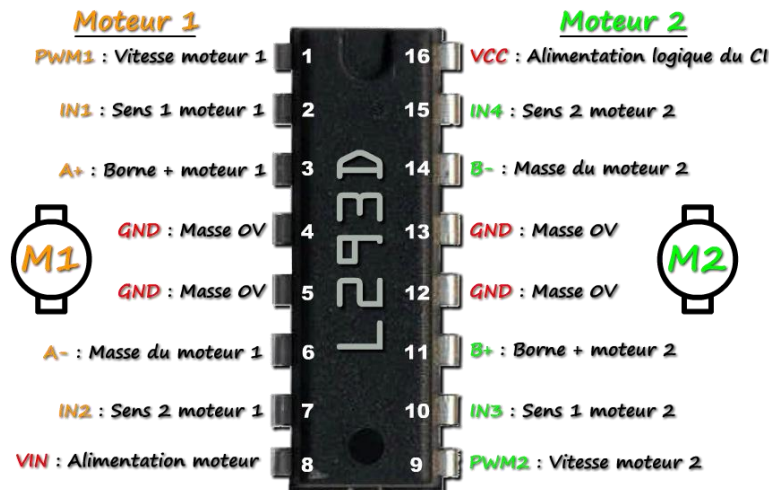


Fig 11 : Différentes broches du composant L293D

1. Enable1 : permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 et OUTPUT2 et commande l'activation/désactivation du premier Pont-H.
 2. Input1 : avec Input 2, sont les broches de commande du Pont-H Output1/Output2. Il sera directement branché à l'Arduino pour commander le sens du courant entre Output 1 et Output 2, et entre Output 3 et Output4 pour Input2.
 3. Ouput1 : avec Output 2, sera branché directement sur le moteur.
 - 4.5.12.13 GND : qui doit être raccordé à la masse de la source d'alimentation de puissance VS et à la masse de la source d'alimentation de la logique "VSS".
 8. VS : Alimentation de puissance des moteurs.
 9. Enable2 : commande l'activation du second pont-H constitué d'Output3/Output4.
 10. Input3 : avec Input 4, sont les broches de commande du Pont-H Output3/Output4. Il sera directement branché à l'Arduino pour commander le sens du courant entre Output 3 et Output 4.
 11. Ouput3 : avec Output 4, sera branché directement sur le moteur.
 16. VSS : Alimentation de la logique de commande (5V). A raccorder à la borne +5V l'Arduino.
- NB : les pins Enable1 et Enable2 permettent moduler la vitesse du moteur en utilisant des broches PWM de l'Arduino.

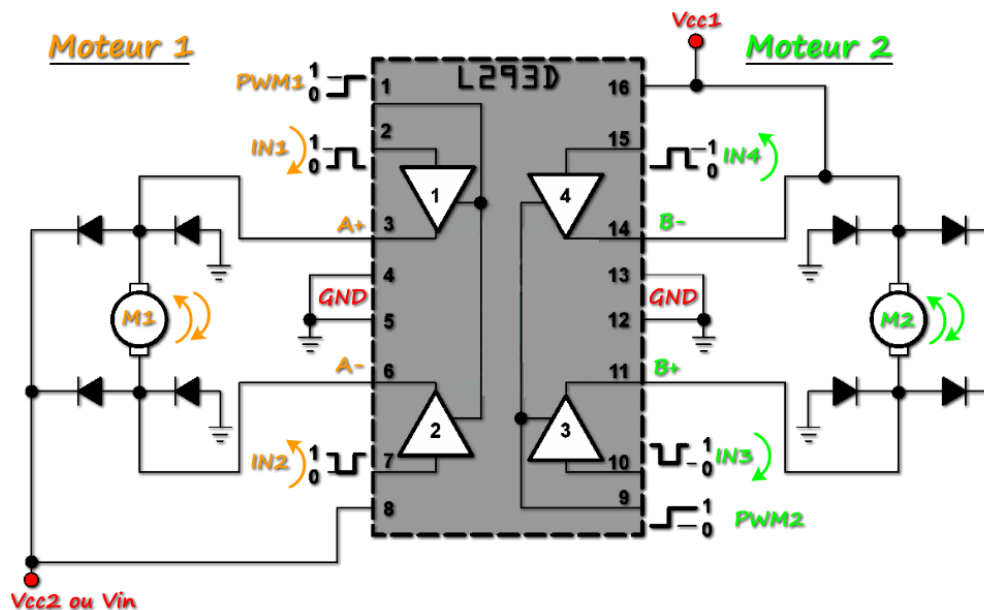


Fig 12 : Raccordement des moteurs

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	Tourne dans le sens anti-horlogique
High	Low	Low	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

Tab 4 : Fonctionnement du moteur DC en utilisant L293D

3.4. Capteur à ultrason HC-SR04

Ce capteur utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter.

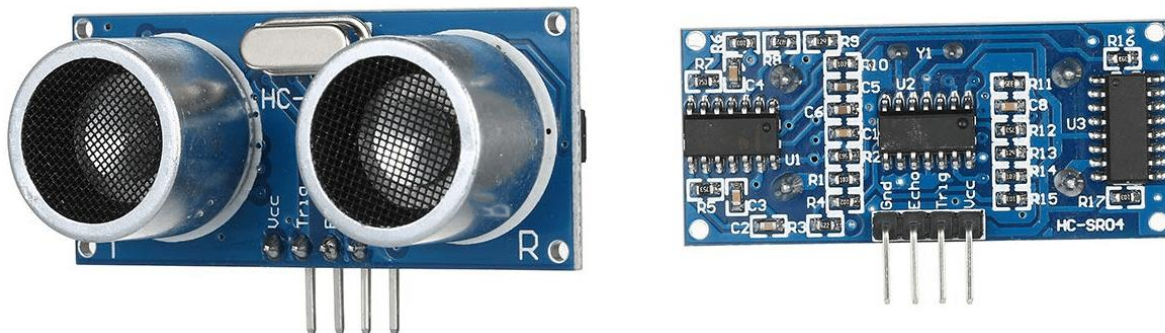


Fig 13 : Capteur ultrason « HC-SR04 »

3.5. Module Bluetooth HC-05 :

Le Bluetooth est un protocole de communication sans fil utilisant la même gamme de fréquences (2.4 GHz) que le WiFi. C'est une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps.

Le module Bluetooth HC-05 est constitué d'un ensemble de broches. VCC et GND pour l'alimentation (5V), Rx/Tx pour la communication. Et d'une broche «Enable» qui servira à envoyer des commandes de configuration au module. La dernière broche nommée « Led » permet de brancher une LED pour obtenir un signal sur l'état du module.

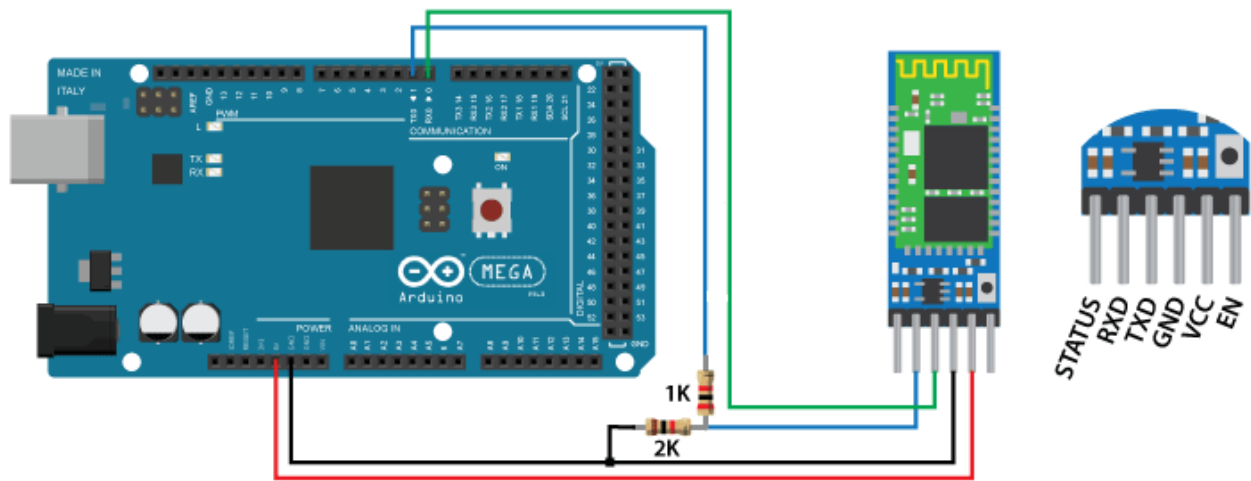


Fig 14: Montage module Bluetooth HC-05

3.6. Servomoteur :

C'est un type spécial de moteur à courant continu qui utilise un codeur pour déterminer la position de l'arbre de sortie, et qui obéit à la loi de Faraday.

Ces moteurs peuvent se déplacer rapidement vers une position spécifique et ne nécessitent que trois fils pour s'interfacer avec l'Arduino (signal, alimentation et terre). Le signal utilisé pour faire fonctionner le servo est une impulsion électrique synchronisée avec précision qui varie de 1 milliseconde à 2 millisecondes.

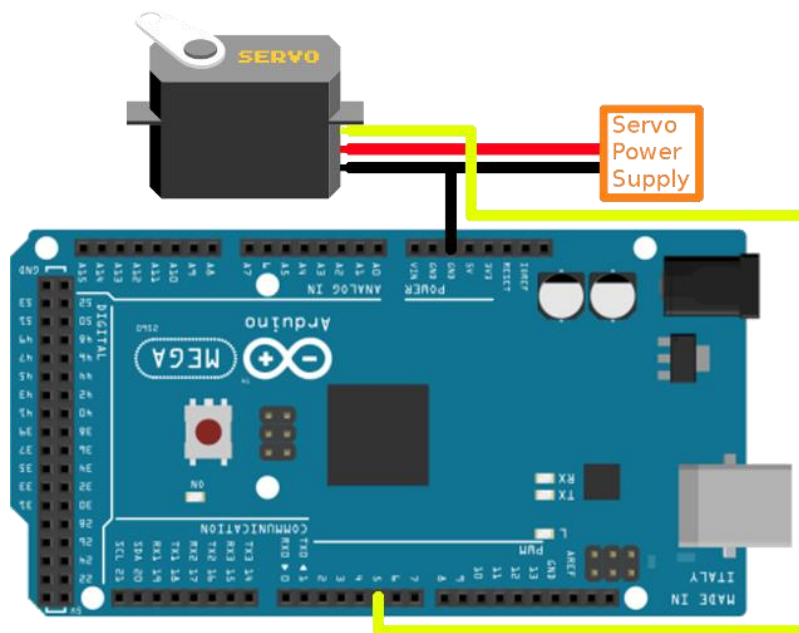
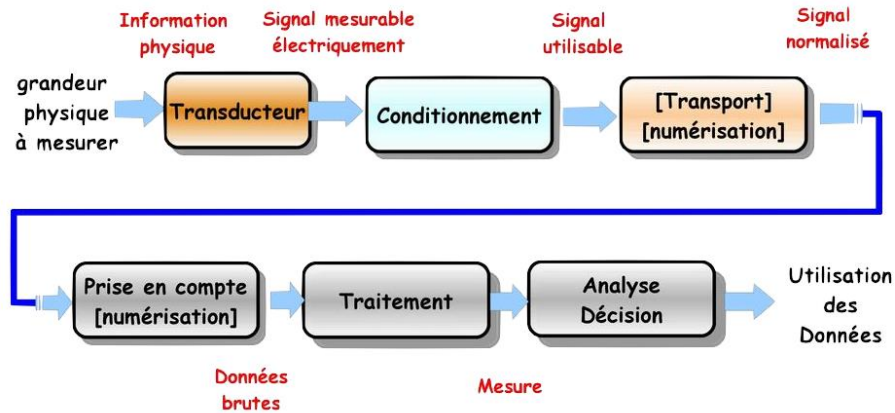


Fig 15 : Montage d'un servomoteur

3.7. Les capteurs :

C'est des organes de prélèvement d'informations qui élaborent à partir d'une grandeur physique (Information entrante) une autre grandeur physique de nature différente (Information sortante : très souvent électrique). Cette grandeur, est utilisable à des fins de mesure ou de commande.

Les grandeurs physiques viennent modifier les caractéristiques du capteur. Ainsi, on peut écrire : $g = f(m, \theta)$ Où : g : la grandeur de sortie. m : le mesurande. θ : les grandeurs d'influence.



Org 2 : Chaîne de mesure

Le capteur peut être passif si la grandeur de sortie est une variation de la valeur d'une résistance ou d'une impédance. Ce circuit appelé conditionneur, a pour rôle de conditionnement pour convertir la variation d'impédance en une variation de tension ou de courant.

Il est actif (transducteur) : s'il fournit en sortie une tension ou un courant, en prélevant une partie de l'énergie utilisée par la grandeur physique la transformant en énergie électrique.

3.8. Smartphone Android :



Fig 16 : Redmi note 8

3.8.2. Application “Arduino Bluecontrol”

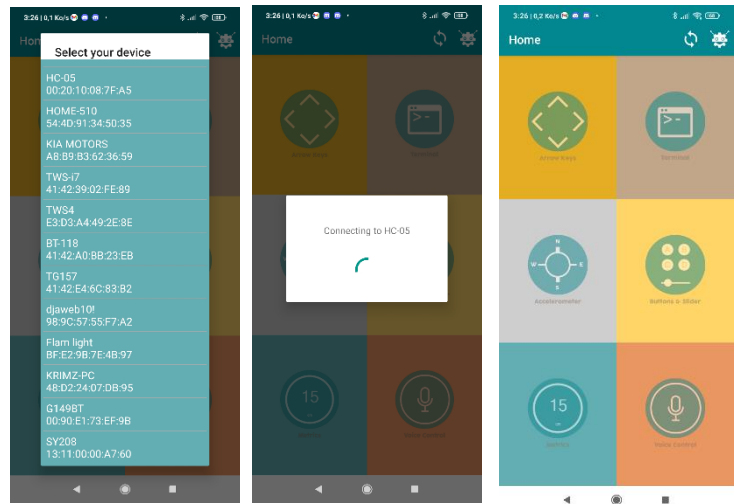


Fig 17 : Interface de l'application

3.9. Kit de châssis

Le châssis utilisé est composé de :

- (1) Châssis acrylique pour robot intelligent x1
- (2) Support acryliques pour les moteurs x4
- (3) Motoréducteur avec double axe et câbles x2
- (4) Roue Ø65mm avec pneu en caoutchouc x2
- (5) Vis M3×30 mm x4
- (6) Écrous M3 x10
- (7) Vis M3×8 mm x4
- (8) Roue libre robotique x1
- (9) Outil nécessaire : Tournevis x1

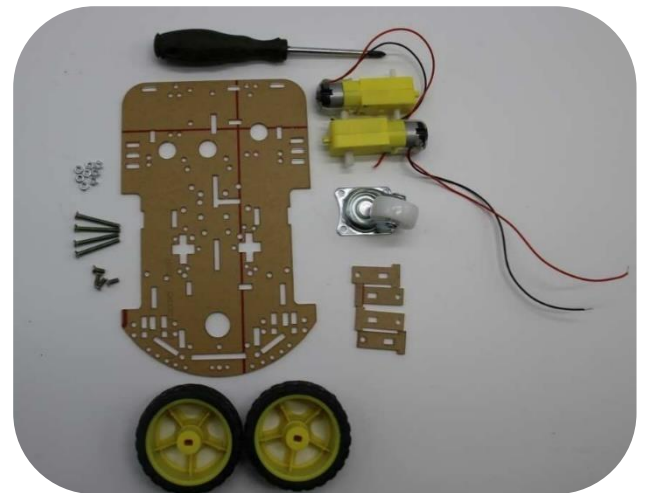


Fig 18 : Kit de châssis utilisé

3.10. Conclusion :

Dans ce chapitre, nous avons présenté les différents composants utilisés dans le projet et leurs spécifications et caractéristiques, ainsi que les logiciels utilisés pour la programmation du système.

Chapitre 4 :

Réalisation du projet

4.1. Introduction :

Dans ce chapitre, on présentera la partie réalisation et conception de notre travail « Fauteuil roulant électrique », après avoir exposé dans le chapitre précédent, la description théorique des dispositifs utilisés. Pour la réalisation de notre travail, plusieurs outils de développement sont disponibles et vu les contraintes matérielles et logicielles on a opté pour la carte « Arduino »

4.2. Fonctionnement du projet

Le fauteuil est commandé avec la voix en utilisant un Smartphone Android, le signal vocal est converti en texte à l'aide d'une application Android. Ce texte est transféré via un module Bluetooth vers l'Arduino qui contrôle le mouvement et la direction du fauteuil roulant.

L'Arduino décide du fonctionnement des deux moteurs à courant continu en fonction du texte reçu, à travers le composant L293D.

Les directions et les mouvements possibles du fauteuil roulant sont indiqués ci-dessous.

En avant « avance » : les deux moteurs sont en marche avant.

En arrière « recule » : les deux moteurs sont en marche arrière.

Gauche « gauche » : moteur gauche en arrêt, et moteur droit en marche avant.

Droite « droite » : moteur droit en arrêt, et moteur gauche en marche avant.

Arrêt « stop » : les deux moteurs sont arrêtés

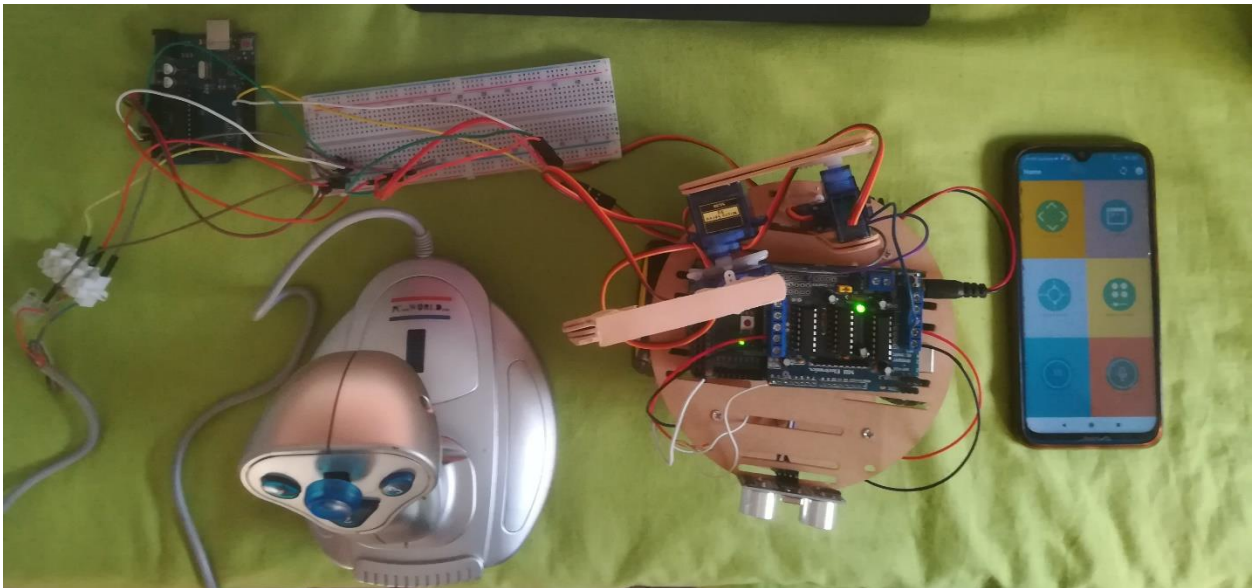


Fig 19 : Prototype mis en marche

4.3. Schéma synoptique :

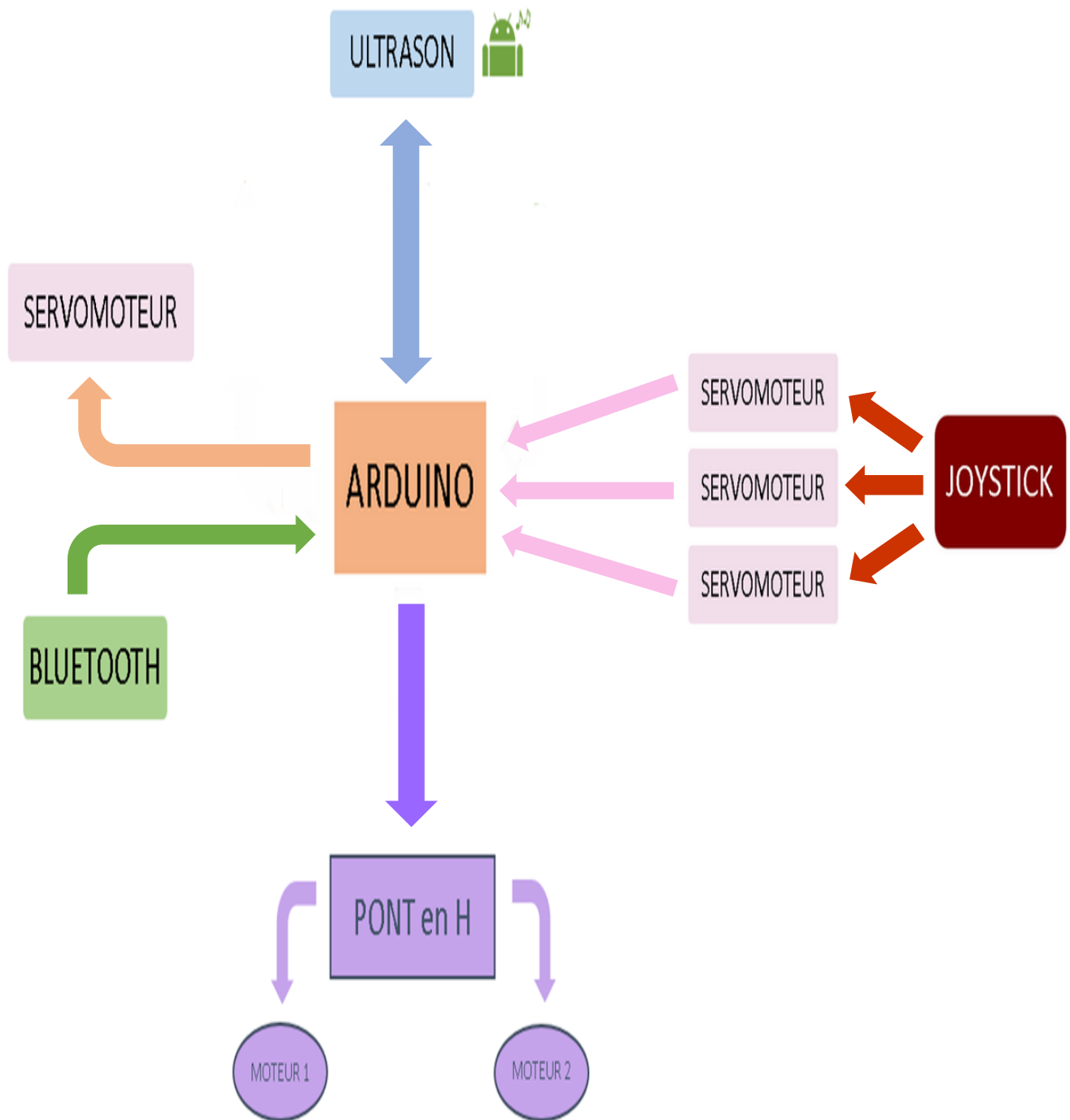
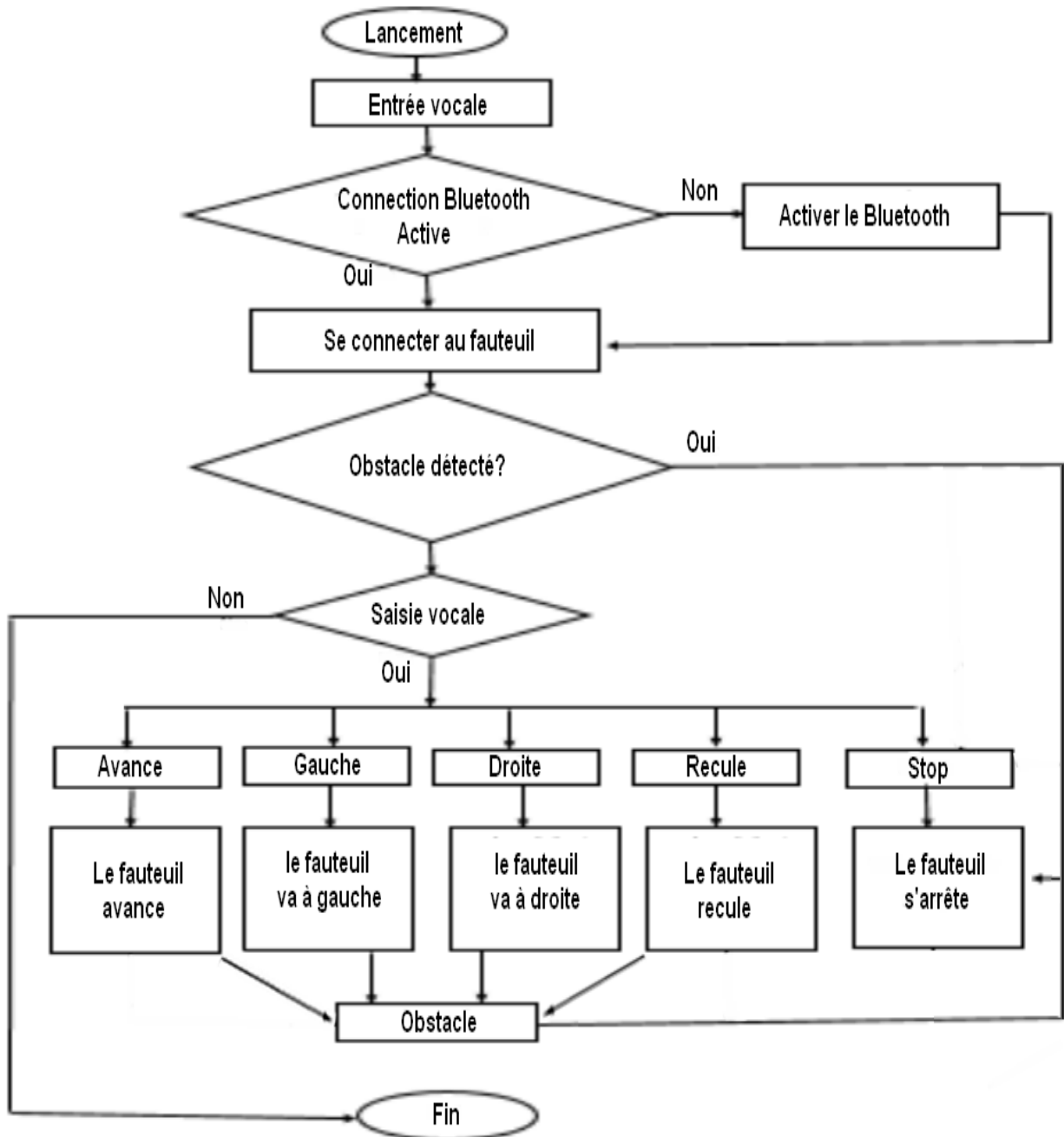
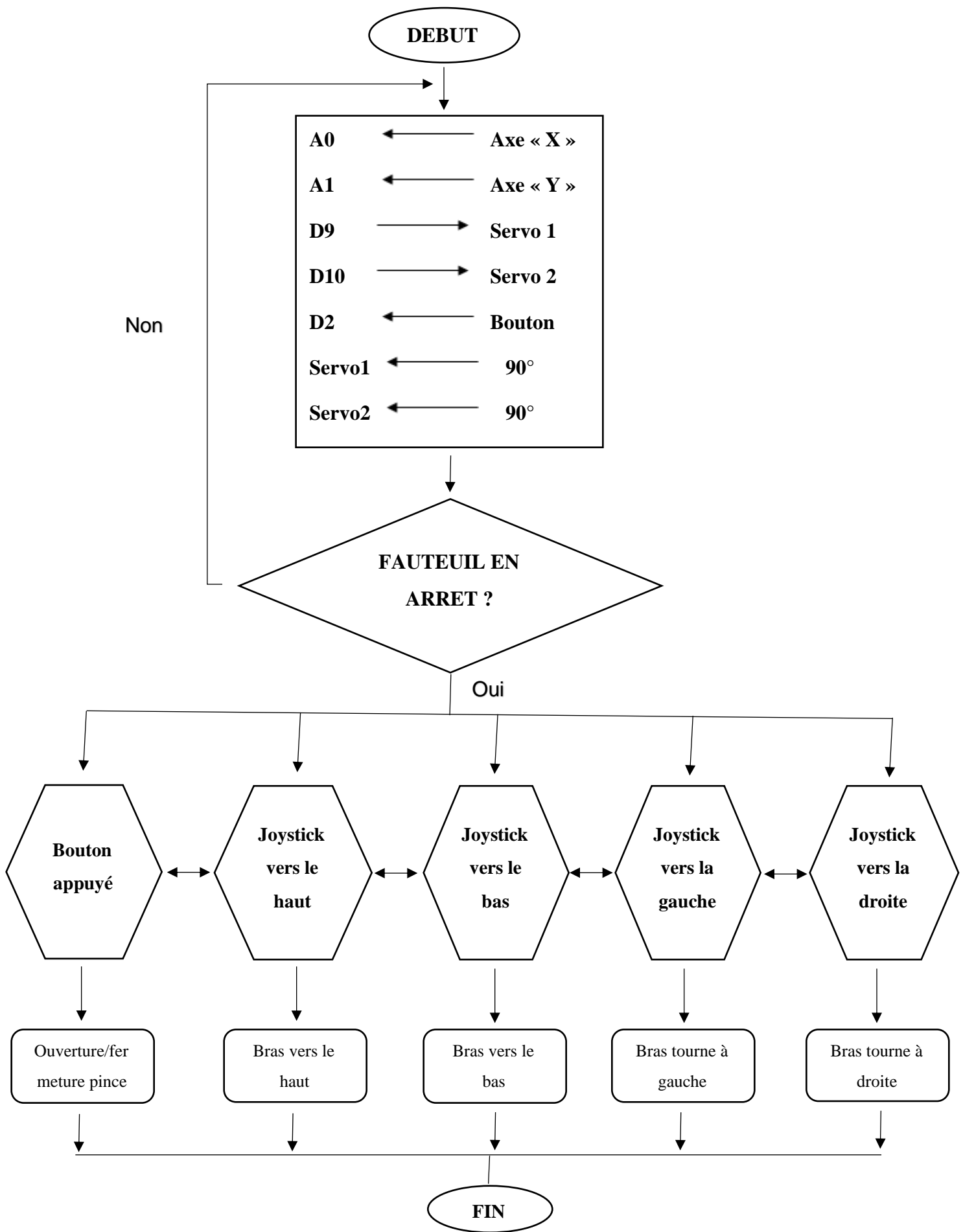


Fig 20 : Schéma synoptique

4.4. Organigrammes :



Org 3 : Organigramme du fauteuil roulant



Org 4 : Organigramme de la commande du bras

4.5. Schéma électrique :

Le schéma électronique présentant la connexion électrique des différents composants et circuits est donné par la figure suivante, obtenue en utilisant le logiciel « Proteus » :

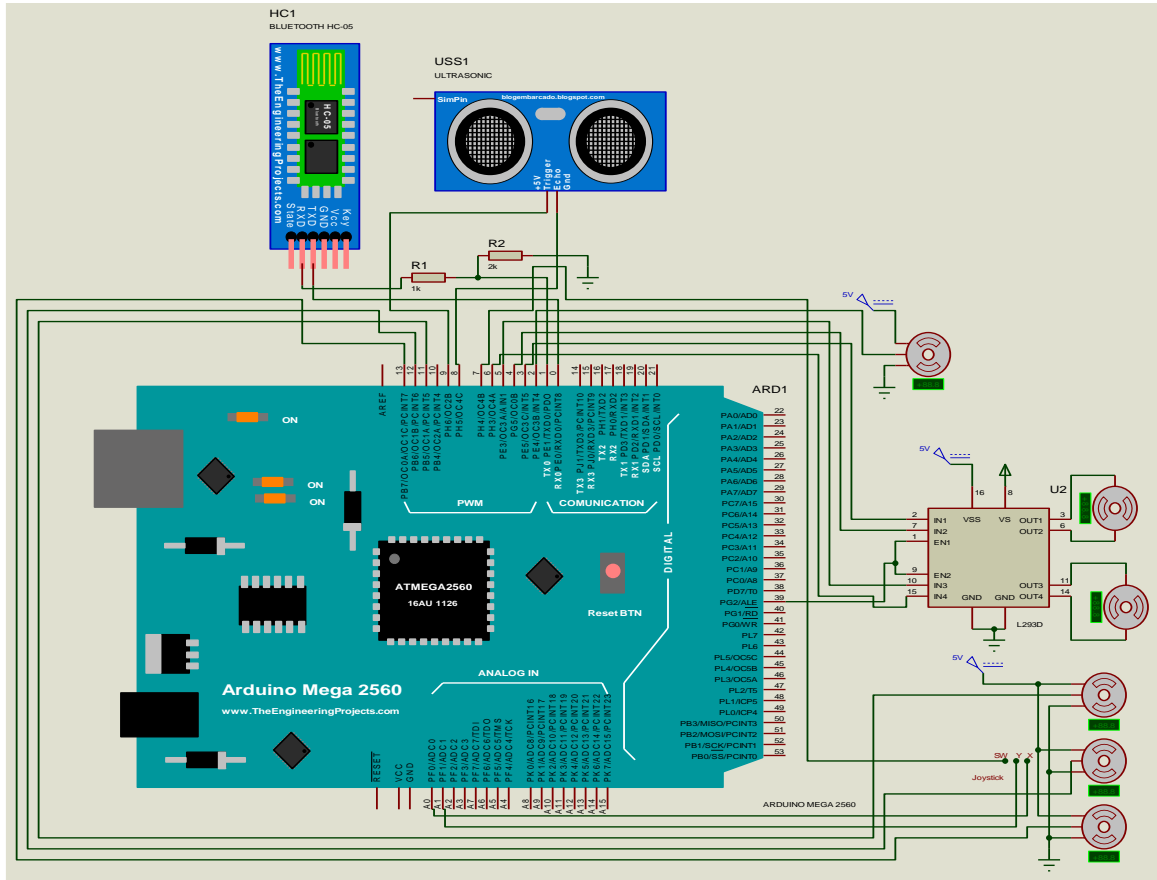


Fig 21 : Schéma électronique du circuit réalisé

4.6. Configuration du Bluetooth HC05 :

Connectez la broche Rx du module Bluetooth à la broche 11 de l'Arduino et la broche Tx de Bluetooth à la broche 10 de l'Arduino. Connectez également la broche KEY du module à la broche 9 de l'Arduino et la masse à la masse. Vous devez d'abord allumer l'Arduino, puis connecter Vcc. La LED du module clignotera à l'intervalle de 2 secondes. Cela signifie qu'il est entré en mode AT. Si elle clignote plus rapidement, déconnectez Vcc, vérifiez votre circuit et réessayez.

Pour bien vérifier que le HC05 est en mode AT , ouvrir le SERIAL MONITOR et écrivez « AT » Si le module est en mode AT il va répondre « OK ».

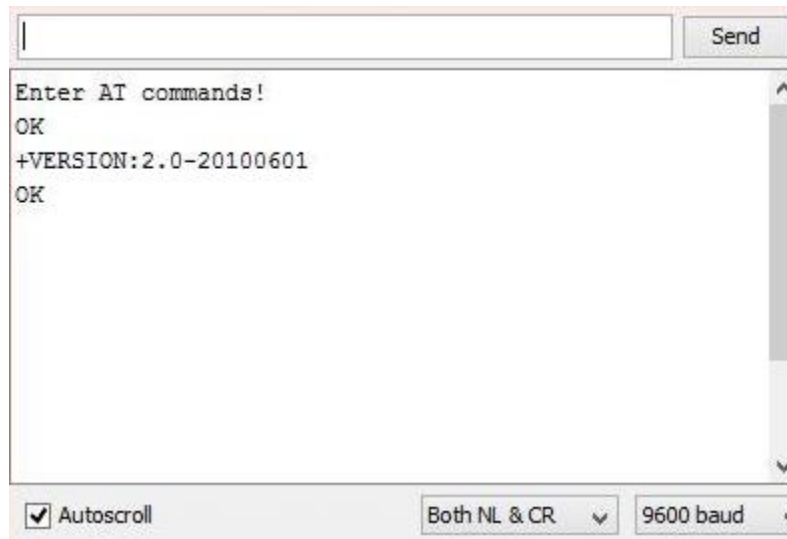


Fig 22 : Communication Bluetooth par terminal « arduino »

4.7. Utilisation de l'ultrason (capteur) :

Le détecteur HC-SR04 utilise les ultrasons pour déterminer la distance à laquelle se trouve un objet. Peu importe l'intensité de la lumière, la température ou le type de matière, le capteur pourra le plus souvent, détecter s'il y a un obstacle devant lui. Si une impulsion de plus de $10\mu\text{s}$ est détectée, alors le capteur envoie une série de 8 impulsions à ultrason de 40kHz et attend le réfléchissement du signal. Ensuite, en ayant en tête la vitesse du son, il effectue un rapide calcul pour déterminer la distance, comme suit :

$$d = v.t$$

"d" pour la distance,

"v" pour la vitesse

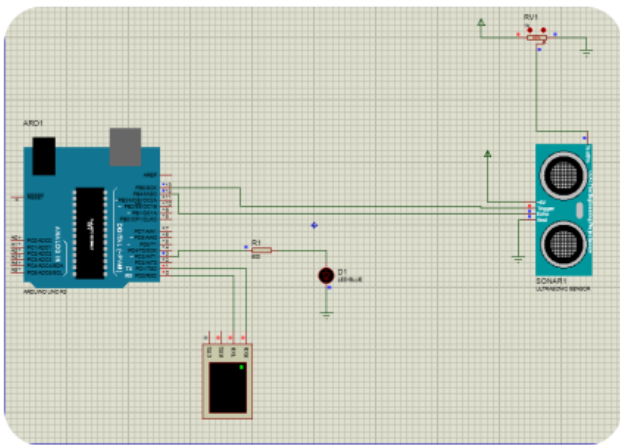
"t" pour le temps.

On connaît la vitesse d'impulsion qui est égale à $10\mu\text{s}$. Le son doit effectuer un aller (en partant du capteur) pour ensuite rebondir sur l'obstacle et revenir à ce point de départ. Nous diviserons donc son temps par deux.

La vitesse du son est égale à environ 340 m/s ce qui nous donne:

$$d = 340.t/2 = 170. \text{ donc } d = t / 0.0058.$$

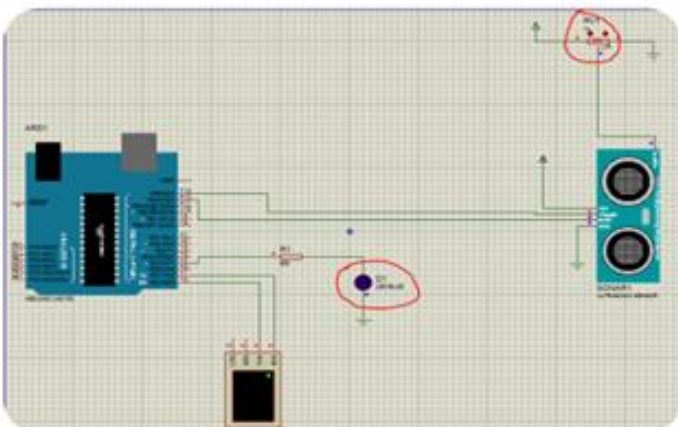
Pour la distance en cm, nous obtenons : $d = t / 0.58$



```

Virtual Terminal
Distance: 887.06mm (88.71cm, 0.89m)
Distance: 886.04mm (88.60cm, 0.89m)
Distance: 887.23mm (88.72cm, 0.89m)
Distance: 887.06mm (88.71cm, 0.89m)
Distance: 887.06mm (88.71cm, 0.89m)
Distance: 887.06mm (88.71cm, 0.89m)
Distance: 887.06mm (88.71cm, 0.89m)
Distance: 886.04mm (88.60cm, 0.89m)
Distance: 887.06mm (88.7
  
```

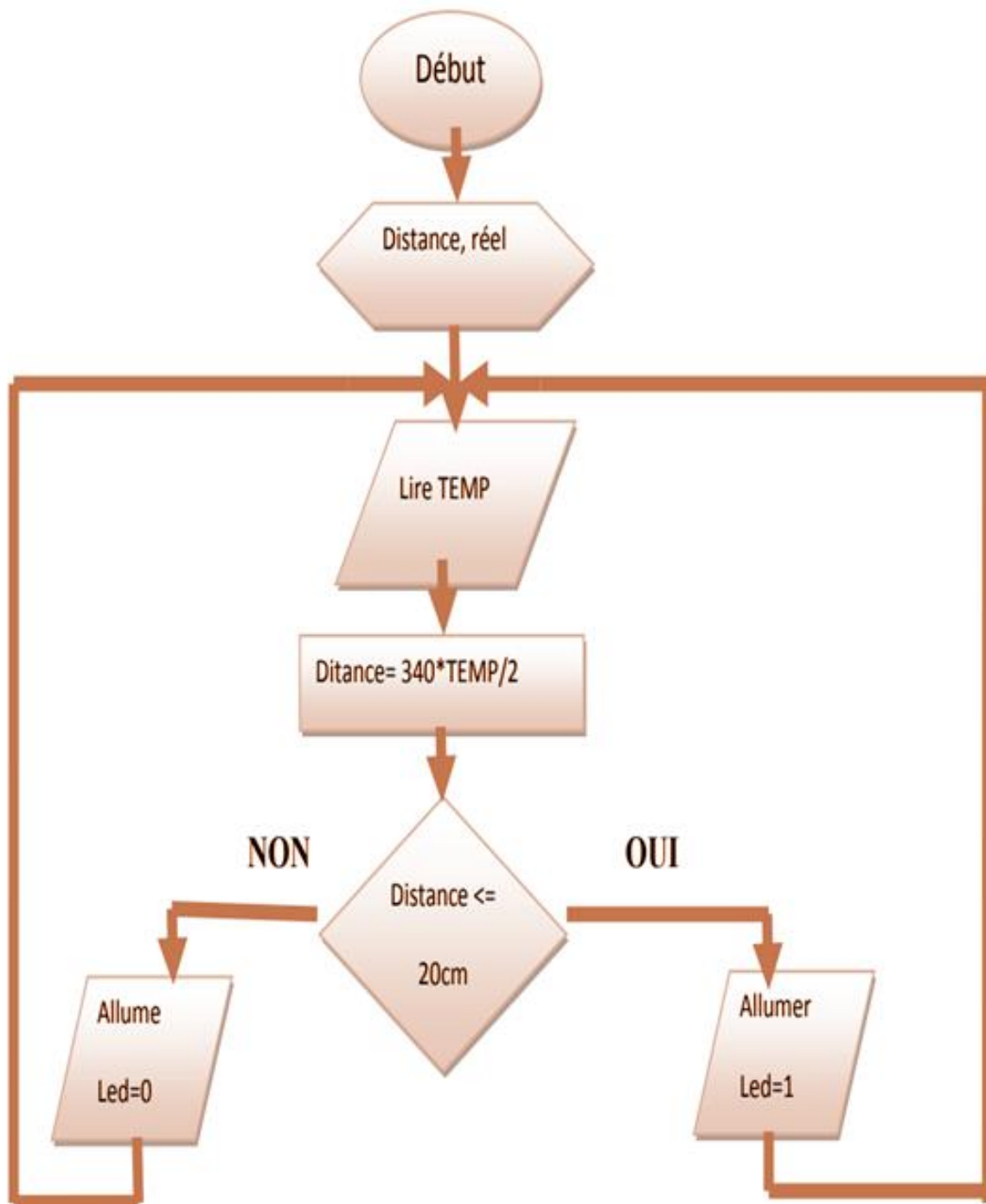
En variant le potentiomètre qui correspond à la distance visée, la LED s'allume :



```

Virtual Terminal
Distance: 335.41mm (33.54cm, 0.34m)
Distance: 260.44mm (26.04cm, 0.26m)
Distance: 259.42mm (25.94cm, 0.26m)
Distance: 108.12mm (10.81cm, 0.11m)
Distance: 109.14mm (10.91cm, 0.11m)
Distance: 108.12mm (10.81cm, 0.11m)
Distance: 108.12mm (10.81cm, 0.11m)
Distance: 109.14mm (10.91cm, 0.11m)
Distance: 108.12mm (10.81cm, 0.11m)
Distance: 108.12mm (10.81cm, 0.11m)
Distance: 109.14mm (10.91cm, 0.11m)
Distance: 108.12mm (10.81cm, 0.11m)
Distance: 108.12mm (10.81cm, 0.11m)
Distance: 109.14mm (10.91cm, 0.11m)
  
```

Fig 23 : Simulation de l'ultrason et arduino



Org 5 : Organigramme de détection d'obstacles

4.8. Le pont en H :

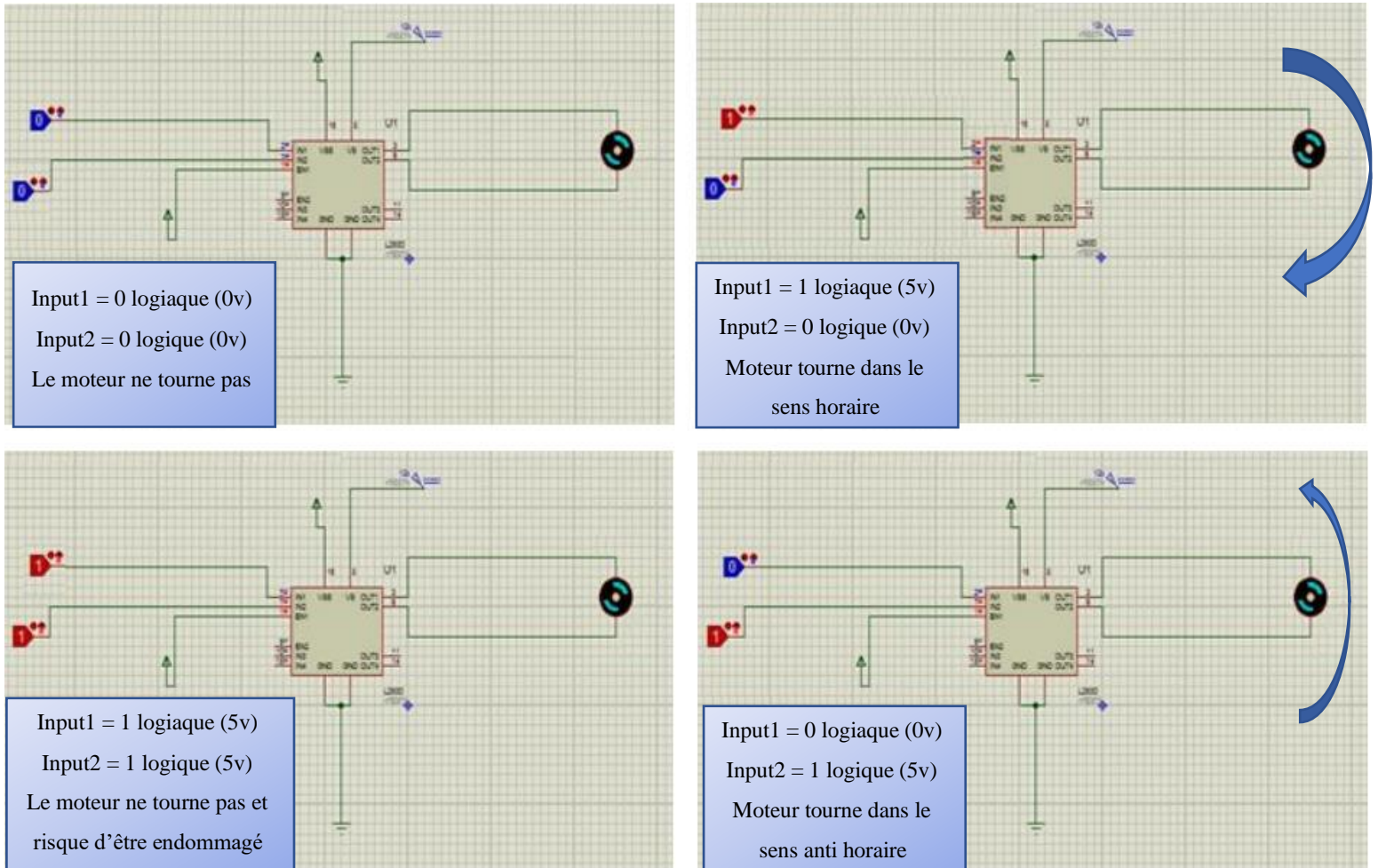


Fig 24 : Utilisation du pont en H pour commander un MCC

4.9. Variation de la vitesse du moteur à courant continu (MCC) :

Sur la carte ARDUINO il ya des broches spécifiées pour l'utilisation de MLI : 3, 5, 6, 9,10 et 11 ils sont symbolisés pas le symbole alternatif. Sur le pont H elle est utilisée sur les broches ENEBLE (sur des drivers moteurs elle est écrite PWM).

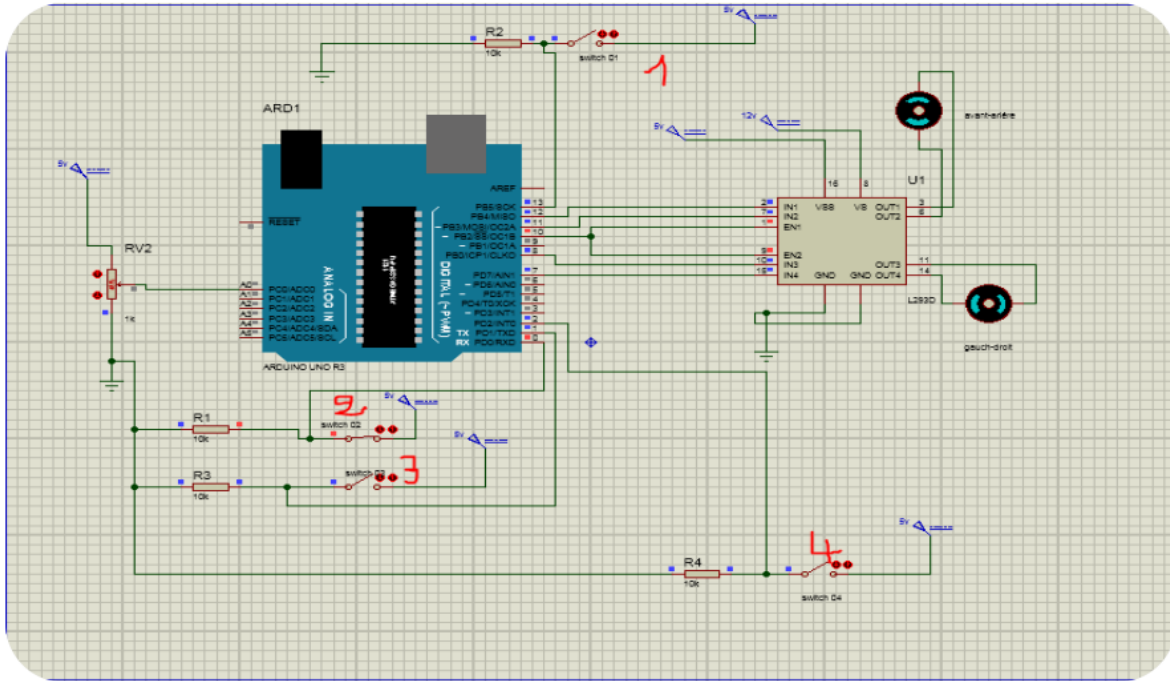
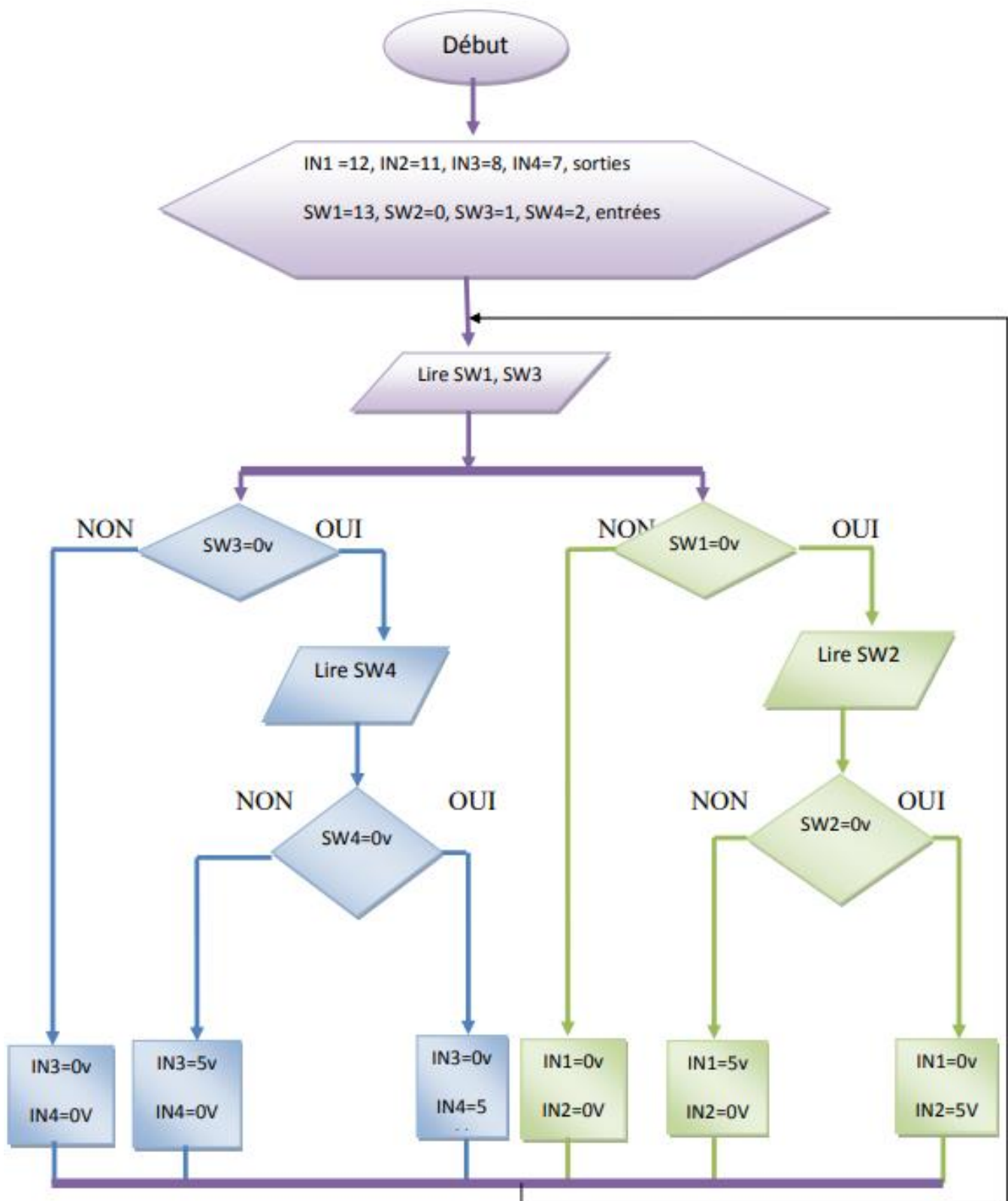


Fig 25 : Commande du MCC

	L293	ARDUINO
Moteur 1	INPUT1 = pin 2	PIN 12
	INPUT2 = pin 7	PIN 11
	ENABLE1 = pin 1	PIN 10
Moteur 2	ENABLE1 = pin 9	PIN 10
	INPUT3 = pin10	PIN 8
	INPUT4 = pin15	PIN 7

Tab 5 : Brochage du L293D avec Arduino



Org 6 : Commande des moteurs

4.10. Adaptation d'un ancien joystick analogique à arduino :



Fig 26 : Joystick analogique (DB15)

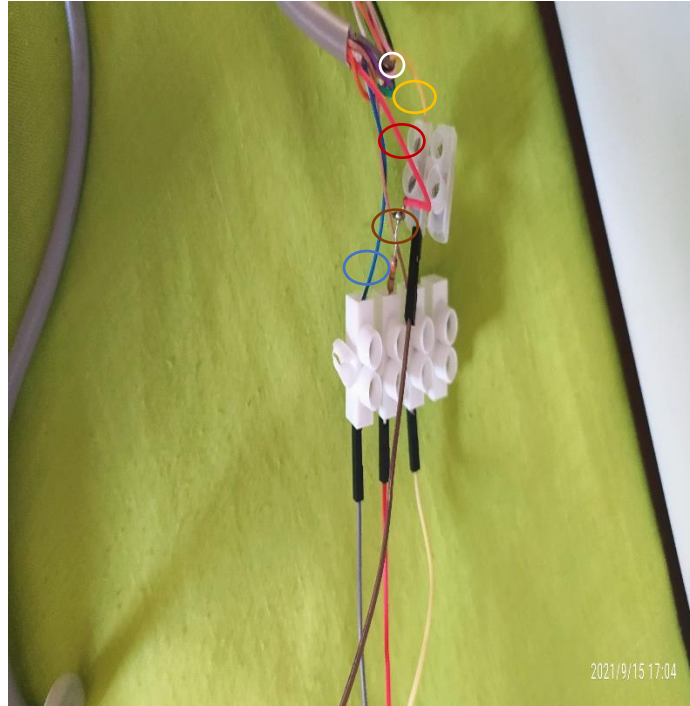


Fig 27 : Câblage des fils du joystick

Fil rouge +5v

Fil bleu Gnd

Fil jaune Axe y

Fil marron Axe x

Fil blanc Boutton 2

Brochage du Joystick avec Arduino

Le joystick est composé de deux potentiomètres couplés mécaniquement placés de manière à détecter les composantes horizontale et verticale du joystick. Ainsi les valeurs de résistance des potentiomètres varient indépendamment en fonction de la position du joystick. [2]

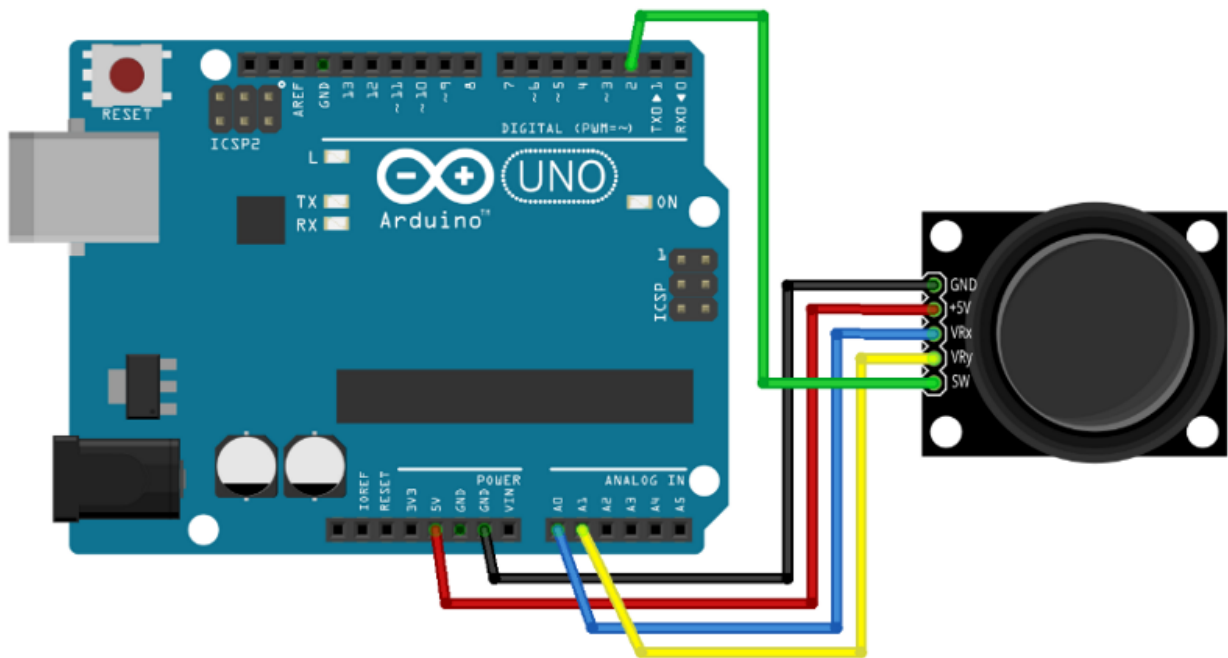


Fig 28 : Brochage du joystick

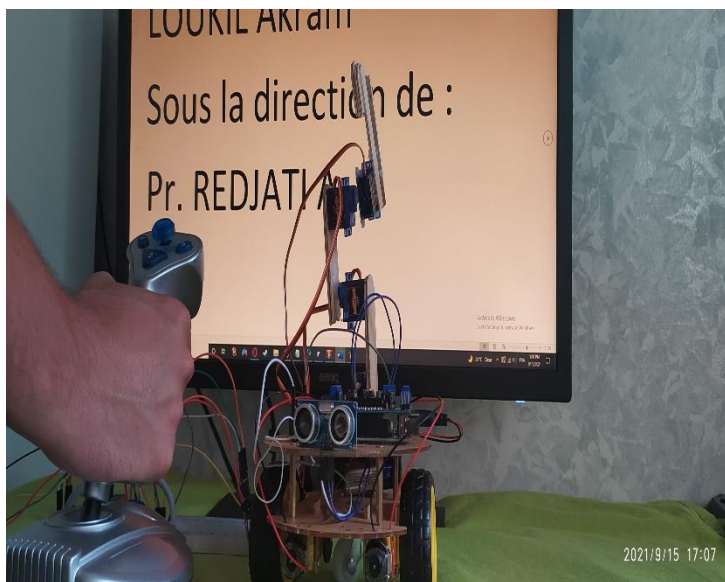


Fig 29 : Joystick axe « Y » Positive

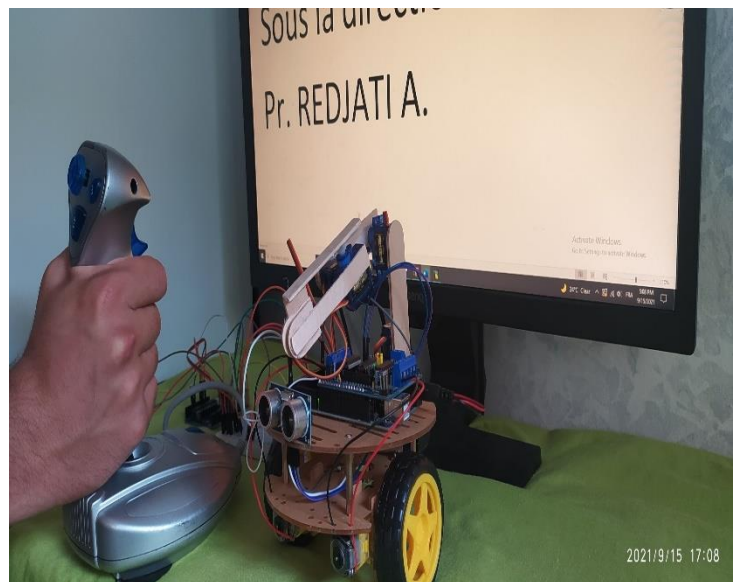


Fig 30 : Joystick axe « Y » Negative



Fig 31 : Joystick axe « X » Positive



Fig 32 : Joystick axe « X » Negative

Résultats :

Une fois le code chargé dans le microcontrôleur vous devriez voir les valeurs de X ,Y et le bouton switch (ou S2 pour notre projet) évoluer en fonction de la position du Joystick

```

Switch: 1 | X-axis: 1022 | Y-axis: 512
Switch: 1 | X-axis: 1023 | Y-axis: 512
Switch: 1 | X-axis: 1023 | Y-axis: 513
Switch: 1 | X-axis: 0 | Y-axis: 511
Switch: 1 | X-axis: 0 | Y-axis: 512
Switch: 1 | X-axis: 0 | Y-axis: 512
Switch: 1 | X-axis: 520 | Y-axis: 0
Switch: 1 | X-axis: 520 | Y-axis: 0
Switch: 1 | X-axis: 520 | Y-axis: 0
Switch: 1 | X-axis: 520 | Y-axis: 1023
Switch: 1 | X-axis: 520 | Y-axis: 1023
Switch: 1 | X-axis: 520 | Y-axis: 1023
Switch: 0 | X-axis: 520 | Y-axis: 512
Switch: 0 | X-axis: 520 | Y-axis: 512
Switch: 0 | X-axis: 520 | Y-axis: 512
  
```

Fig 33 : Valeurs obtenues

4.11. Coût du matériel :

Composants	Prix
Arduino Mega 2560	3000 DA
Shield L293D	1400 DA
SERVOMOTEUR x4	450 DA x4
Joystick	2000 DA
Capteur Ultrason	700DA
PACK châssis roues Arduino	4000DA
Batterie 9V	700DA
	Prix total : 13600 DA

Tab 6 : Coût total du matériel utilisé

4.11. Conclusion :

Dans ce quatrième chapitre, nous avons le schéma global de notre système ainsi que le schéma synoptique, nous avons expliqué son principe de fonctionnement et nous avons donné l'organigramme du programme que nous avons réalisé pour le bon fonctionnement de notre système.

Conclusion générale

Ce projet nous a permis d'étudier et de comprendre le cahier des charges, et le fonctionnement d'un FRE intelligent, et de mettre en place une feuille de route pour dépasser les différentes contraintes de déplacement de ceux-ci.

La conception d'un modèle réduit nous a permis de représenter les déplacements d'un FRE équipé d'un module anticollision et commandé par la voix, cette réflexion autour de l'algorithme de commande, nous a mené dans un esprit d'originalité et de découverte, à l'enrichir par adjonction d'un bras à quatre degrés de liberté, dont les mouvements sont régis par les signaux émis par un joystick.

La réponse aux besoins des personnes à fort handicap moteur en matière d'aide à la navigation, passe d'abord par une méthodologie d'évaluation de ces besoins, des possibilités techniques, et des capacités résiduelles de ces personnes.

Ce projet nous a permis de mettre à profit nos connaissances pour une cause humaine, de mener une mission à bien, depuis la compréhension de la demande jusqu'à la création d'un prototype.

Parmi les contraintes qu'on a rencontré lors de la réalisation de ce projet, le manque de temps pour pouvoir approfondir ce travail, et le manque de matériels dans le contexte pandémique actuel, car au final ce projet pourrait être amélioré.

En effet, pour un passionné, les possibilités sont nombreuses, tant au niveau des équipements utilisables que des opérations effectuelles.

Les perspectives essentielles des travaux énoncés résident, d'une part dans l'intégration du principe de signature vocale, et d'autre part dans l'amélioration de l'interface homme/machine, qui doit se rapprocher d'un dialogue de type naturel pour être plus conviviale.

Enfin, un transfert technologique de notre projet sur un fauteuil roulant grandeur nature s'avère indispensable pour valider le système dans des conditions se rapprochant de l'utilisation finale attendue de cette application.

Annexes

Programme de Controller des 2 Moteurs :

```
int motor1_enablePin =11;//pwm
int motor1_in1Pin =13;
int motor1_in2Pin =12;
int motor2_enablePin =10;//pwm
int motor2_in1Pin =8;
int motor2_in2Pin =7;
Voidsetup ()
{
//on initialise les pins du moteur 1
pinMode (motor1_in1Pin, OUTPUT);
pinMode (motor1_in2Pin, OUTPUT);
pinMode (motor1_enablePin, OUTPUT);
//on initialize les pins du moteur 2
pinMode (motor2_in1Pin, OUTPUT);
pinMode (motor2_in2Pin, OUTPUT);
pinMode (motor2_enablePin, OUTPUT);
}
Voidloop ()
{
SetMotor2(175, true);
SetMotor1(255, false);
}
//Function qui set le moteur1
void SetMotor1(intspeed, Boolean reverse)
{
analogWrite (motor1_enablePin, speed);
digitalWrite (motor1_in1Pin, reverse);
digitalWrite (motor1_in2Pin, reverse);
}
//Fonction qui set le moteur2
void SetMotor2(intspeed, Boolean reverse)
{
analogWrite (motor2_enablePin, speed);
digitalWrite (motor2_in1Pin, reverse);
digitalWrite (motor2_in2Pin, reverse);
}
```


Programme de détection d'obstacles (anticollision) :

```
int trig = 2;
int echo = 3;
long lecture_echo;
long cm;

void setup(){

pinMode(trig, OUTPUT);
digitalWrite(trig, LOW);
pinMode(echo, INPUT);
Serial.begin(9600);
}

void loop(){

digitalWrite(trig, HIGH);
delayMicroseconds(10);
digitalWrite(trig, LOW);
lecture_echo = pulseIn(echo,HIGH);
cm = lecture_echo /58;
Serial.print("Distance en cm :");
Serial.println(cm);
delay(1500);
}

if (distance <= 20) {
  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
  digitalWrite(m3,LOW);
  digitalWrite(m4,LOW);
}
}
}
```

Configuration du Bluetooth HC 05 :

```
#include
SoftwareSerial mySerial(10, 11); // RX, TX
void setup() {
  Serial.begin(9600);
  pinMode(9,OUTPUT); digitalWrite(9,HIGH);
  Serial.println("Enter AT commands:");
  mySerial.begin(38400);
}
void loop()
{
  if (mySerial.available())
  Serial.write(mySerial.read());
  if (Serial.available())
  mySerial.write(Serial.read());
}
```

Programme Transformant saisie vocale en locomotion :

```
#include <Servo.h>
#include <AFMotor.h>
#define Echo A0
#define Trig A1
#define motor 10
#define Speed 170
#define spoint 103
char value;
int distance;
int duration;
int Left;
int Right;
int L = 0;
int R = 0;
int L1 = 0;
int R1 = 0;
Servo servo;
AF_DCMotor M1(1);
AF_DCMotor M3(3);

void setup() {
  Serial.begin(9600);
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  servo.attach(motor);
  M1.setSpeed(Speed);
  M3.setSpeed(Speed);
}
void loop() {
  long duration, distance;
  digitalWrite(Trig, HIGH);
  delay(1);
  digitalWrite(Trig, LOW);
  duration = pulseIn(Echo, HIGH);
  distance = (duration / 2) / 29.1;
  Serial.print(distance);
  Serial.println("CM");
  delay(10);
  if ( distance <= 17)
  {
    Stop();
  }
  if (Serial.available() > 0) {
    value = Serial.read();
    if (value == '-') {
      backward();
    } else {
```

```
    if (value == '^') {
        forward();
    }
    else if (value == '-') {
        backward();
    }
    else if (value == '<') {
        left();
    }
    else if (value == '>') {
        right();
    }
    else if (value == '*') {
M1.setSpeed(50);
M3.setSpeed(50);
delay (550);
Stop();
M1.setSpeed(Speed);
M3.setSpeed(Speed);
    }
    else {
        Stop();
    }
}

}
```

```
void forward() {
    M1.run(FORWARD);
    M3.run(FORWARD);
}
void backward() {
    M1.run(BACKWARD);
    M3.run(BACKWARD);
}
void right() {
    M1.run(BACKWARD);
    M3.run(FORWARD);
}
void left() {
    M1.run(FORWARD);
    M3.run(BACKWARD);
}
void Stop() {
    M1.run(RELEASE);
    M3.run(RELEASE);
}
```

Code Arduino « bras manipulateur à 4 DDL »

```
#include <Servo.h>

Servo servo1;
Servo servo2;
int x_key = A0;
int y_key = A1;
int push_button=2;
int x_pos;
int y_pos;
int z_button;
int servo1_pin = 10;
int servo2_pin = 9;
int servo3_pin=11;
int initial_position = 90;
int initial_position1 = 90;
int initial_position2 =90;

void setup ( ) {
  Serial.begin (9600) ;
  servo1.attach (servo1_pin ) ;
  servo2.attach (servo2_pin ) ;
  servo1.write (initial_position);
  servo2.write (initial_position1);
  servo2.write (initial_position2);
  pinMode(2, INPUT_PULLUP);
  pinMode (x_key, INPUT) ;
  pinMode (y_key, INPUT) ;
}
```

```

void loop ( ) {
x_pos = analogRead (x_key) ;
y_pos = analogRead (y_key) ;

if (x_pos < 300){
if (initial_position < 10) { } else{ initial_position = initial_position - 20; servo1.write (
initial_position ) ; delay (100) ; } } if (x_pos > 700){
if (initial_position > 180)
{
}
else{
initial_position = initial_position + 20;
servo1.write ( initial_position ) ;
delay (100) ;
}
}

if (y_pos < 300){
if (initial_position1 < 10) { }
else{ initial_position1 = initial_position1 - 20; servo2.write ( initial_position1 ) ;
delay (100) ;
}
} if (y_pos > 700){
if (initial_position1 > 180)
{
}
else{
initial_position1 = initial_position1 + 20;
servo2.write ( initial_position1 ) ;

```

```
delay (100) ;  
}  
}  
if(z_button == 0){  
  if (initial_position2 == 180 && z_button == 1){  
    servo2.write (90);  
  }  
  else{  
    (initial_position2 == 90 && z_button == 1){  
      servo2.write(180);  
  
    }  
  }  
}
```

Arduino™ MEGA 2560

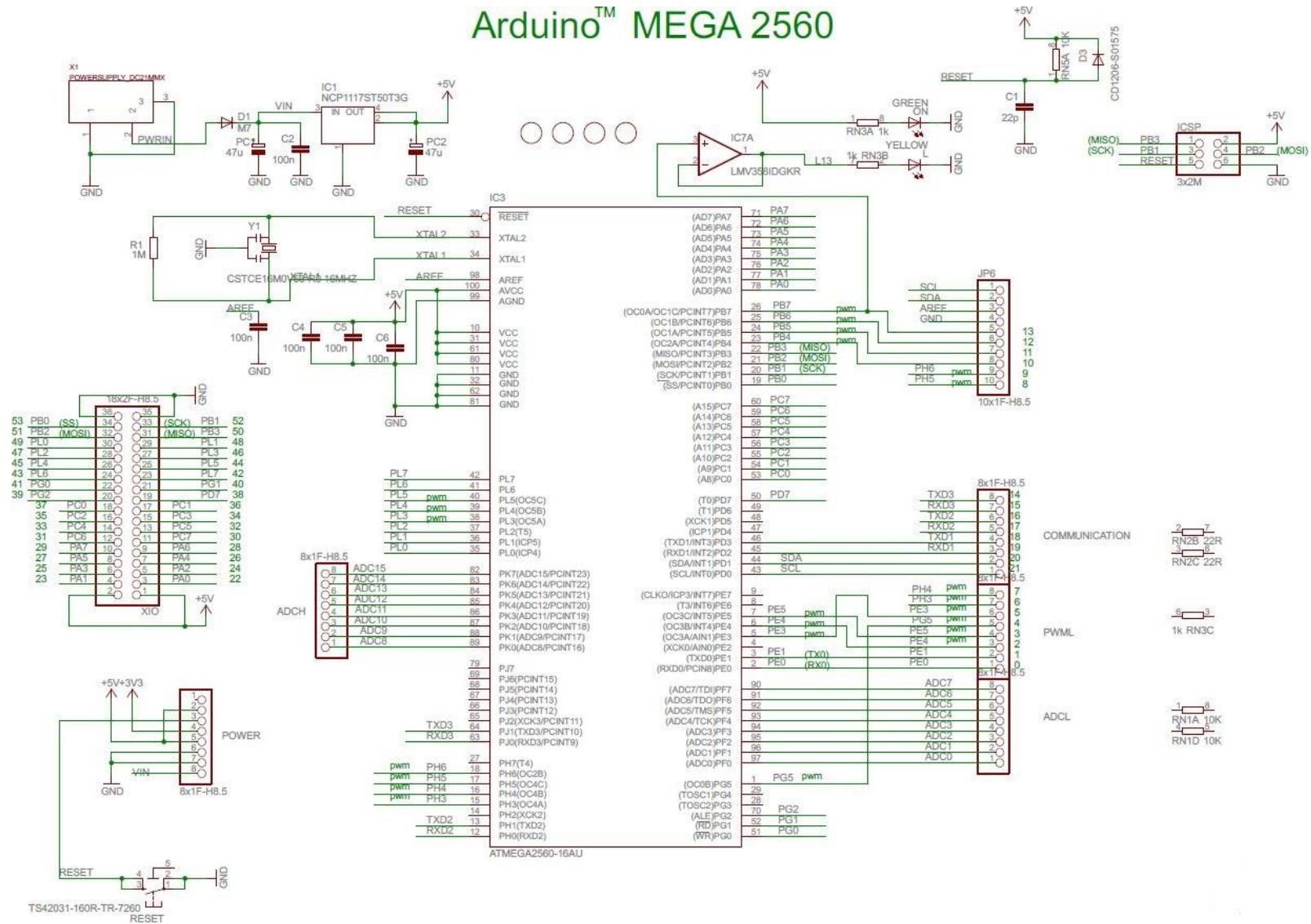


Schéma simplifié Arduino Mega

Joystick

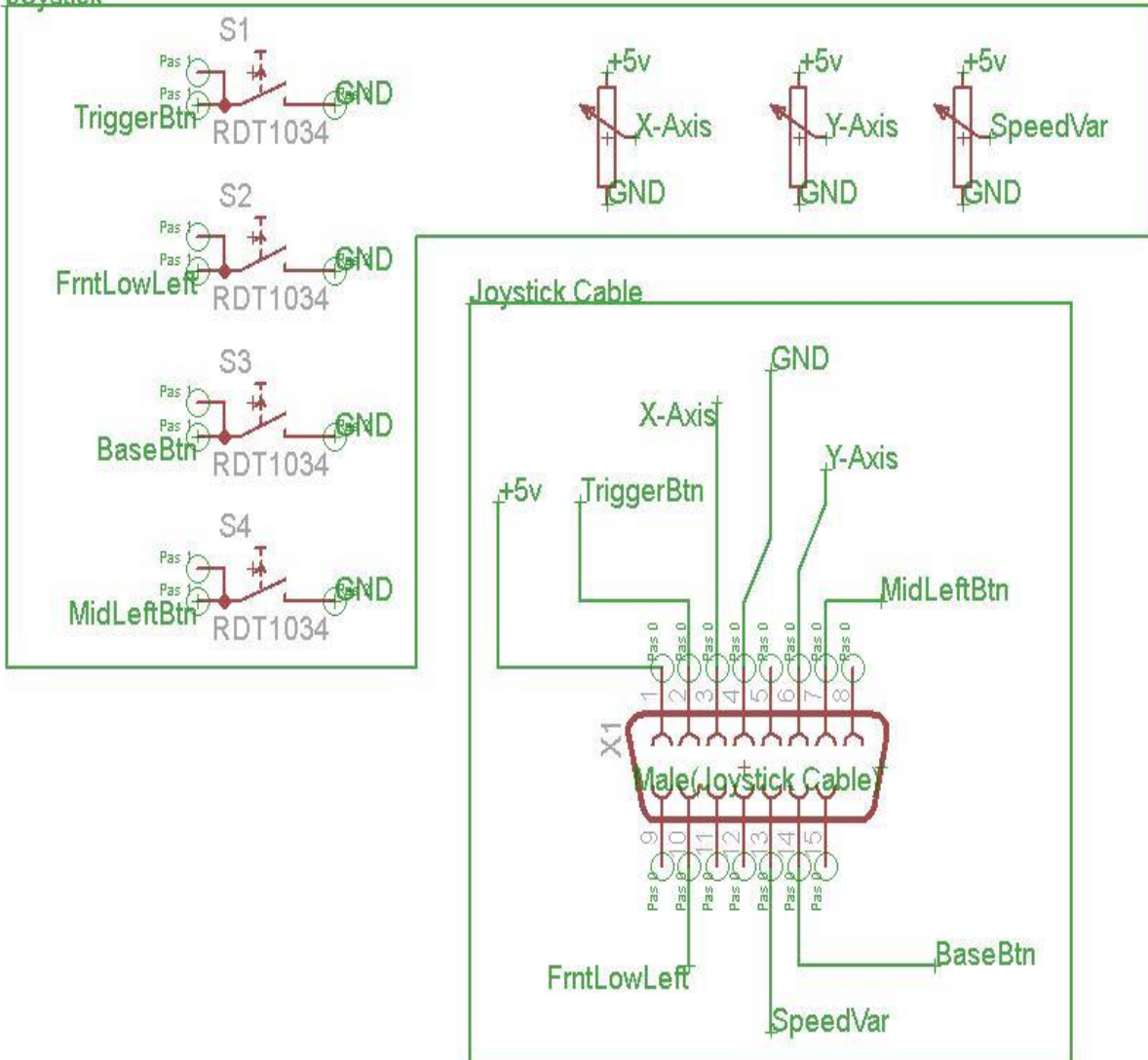
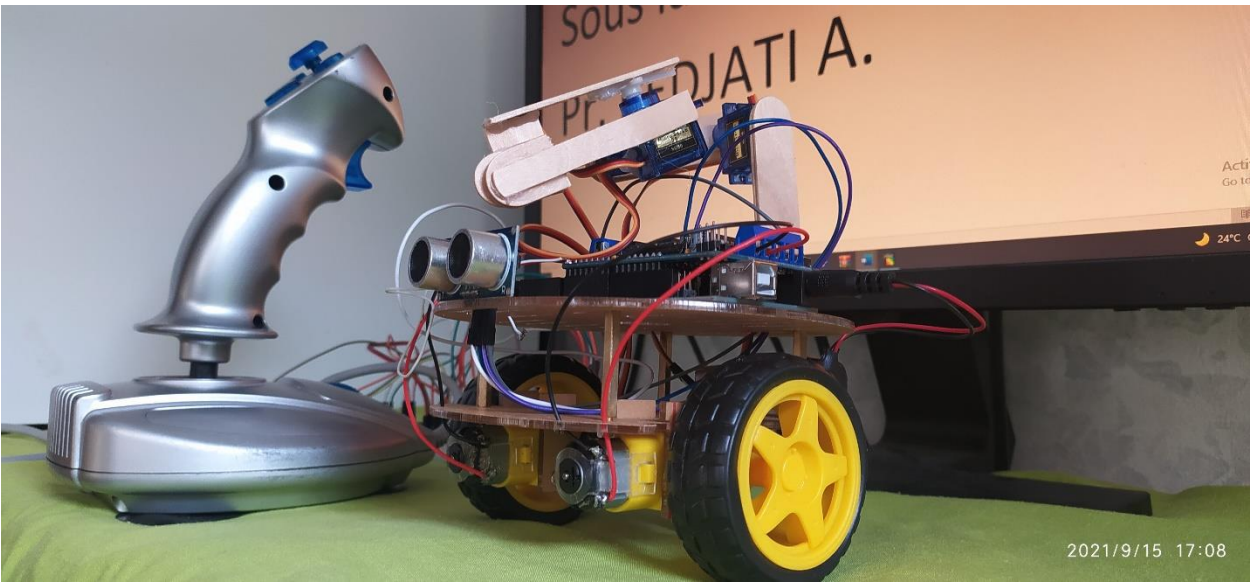


Schéma Joystick Analogique DB15



Prototype sous différents angles

Bibliographie :

[1] K. BRAHIM, FreeMove : un fauteuil roulant automatisé de nouvelle génération, 2014

[2] A. REDJATI, A Verbal Guidance System for Severe Disabled People, AIP 2008

[3] [https://exploreembedded.com/wiki/Analog_JoyStick_with_Arduino]

[4] M. FEZZARI, Integrated Development Environment "IDE" For Arduino, 2018

[5] <https://www.robot-electronics.co.uk/htm/>

[6] S. Patel, Sensor based door navigation for a nonholonomic vehicle, ICRA 2002

[7] F. DESCHAMP, « Convertir L'énergie – Machine à Courant Continu » Sciences de L'ingénieur, LYCEE JACQUES PREVERT.

[8] S. BENMOUSSA, Modélisation et Commande De Robots Industriels, UBMA 2017

[9] J. P. Haton et Al, "Reconnaissance automatique de la Parole" dunod, 1991.

[10] D.K. Lodhi, Smart Electronic Wheelchair Using Arduino and Bluetooth Module, IJCSMC 2016