

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

University 8 Mai 1945 – Guelma

Faculty of Science and Technology

Department of Electronics and Telecommunications



**End of Study Memory
Presented for the Diploma of ACADEMIC MASTER**

Domain: Science and Technology

Sector: Electronics

Specialty: Instrumentation

Driving a wheelchair through an EEG signal using stepper motors

Presented by:

BERREHOUMA Yahia

Supervisor:

Pr. SEDRAOUI Moussa

JUNE 2022

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

ABSTRACT

We try through This Master thesis to develop a prototype for an electrical wheelchair controlled by human MindWave. The goal is to find a precise and effective way to help patients with reduced mobility, particularly stroke victims to find their normal life. To do so, we have combined a set of electronic and mechanic components to construct a remotely controlled omnidirectional mobile robot. After successfully building the robot chassis composed essentially of four mecanum wheels driven by four stepper motors, we started to control its motion with MindWaves and gyroscope where the captured signals are sent via a Bluetooth and WiFi connections. The obtained results were very positive compared to a conventional wheelchair, firstly, by the ability to move the robot laterally (to the right and to the left) without the need to turn, and secondly, by the precision and the stability of it's motion.

key words: Omnidirecionnal motion , Brain-computer interface , MindWave ,EEG,

DEDICATION

We dedicate this modest work to whom brought us love, support and happiness , to the dearest people to our hearts, To our dear parents, for their; love, sacrifice, patience, moral and material support from our childhood until this day, To all our brothers and sisters , who supported us, encouraged us and who took the way with us along all the courses of our studies, To all our dear nephews , to all our uncles and cousins and everyone in our big family. To our dear friends

ACKNOWLEDGMENT

*“ Thanks to the one God, Light of the heavens and the earth,
who help and guide ”*

*we must first of all thank “ ALLAH ” the almighty, who gave us the power,
the will and the patience to develop this work .*

*Our sincerest thanks to Pr. Sedraoui Moussa and my brother
Mr.Berrehouma nabil for thier availability, contributions, valuable
guidance and understanding throughout the development of this thesis.*

*We would also like to thank very much the members of the jury for
accepting, examine and evaluate this modest work .*

also thanks to all our teachers of Licence and Master .

Thank you.

TABLE OF CONTENTS

ABSTRACT	I
DEDICATION	II
ACKNOWLEDGMENT	III
TABLE OF CONTENTS	IV
GENERAL INTRODUCTION	1
CHAPTER I INTRODUCTION TO MOBILE ROBOT	3
I.1 INTRODUCTION	3
I.2 DEFINITION	3
I.3 HISTORICAL DEVELOPMENT	3
I.4 CLASSIFICATIONS OF MOBILE ROBOTS	6
I.5 MOBILE ROBOTS TYPES	7
I.6 FUNCTIONS AND USES OF MOBILE ROBOTS	7
I.7 ADVANTAGES OF MOBILE ROBOTS	8
I.8 CONCLUSION	8
CHAPTER II KINEMATICS OF MOBILE ROBOT	9
II.1 INTRODUCTION	9
II.2 MECANUM WHEELED MOBILE ROBOT	9
II.2.1 Omnidirectional motion	9
II.2.2 Mecanum wheel	9
II.3 KINEMATICS	10
II.4 THE RELATION BETWEEN MOTIONS AND THE TRANSLATION MATRIX	15
II.5 FOUR MECANUM OMNIDIRECTIONAL SOLUTION	15
II.6 THEORETICAL RESULTS	17
II.7 CONCLUSION	17
CHAPTER III ROBOT PROTOTYPE CONSTRUCTION	18
III.1 INTRODUCTION	18

III.2 PROTOTYPE DESIGN	18
III.3 EQUIPMENT USED	20
III.3.1 HC-SR04 sensor	20
III.3.1.1 Sensor pinout	21
III.3.1.2 Sensor features	21
III.3.2 MPU-6050	21
III.3.2.1 Sensor Pinout	23
III.3.2.2 Sensor Features	24
III.3.3 Mecanum wheels	24
III.3.4 Neurosky mind-wave mobile 2	25
III.3.4.1 Device features	25
III.3.5 HC-05 module	26
III.3.5.1 Module features	26
III.3.5.2 Module pinout	27
III.3.6 Microcontroller	27
III.3.6.1 Arduino uno	27
III.3.6.2 ESP32	29
III.3.7 A4988 driver	30
III.3.7.1 Driver Pinout	31
III.3.7.2 Driver Features and Benefits	32
III.3.7.3 A4988 Current Limiting	32
III.3.8 Steppers motor	33
III.3.8.1 Construction and Working Principle	33
III.3.8.2 Motor Features and Benefits	35
III.3.9 Arduino CNC Shield v3.0	35
III.3.10 Power supply	36
III.4 CIRCUIT DIAGRAM	37
III.5 CONCLUSION	38
CHAPTER IV EXPERIMENTAL STUDY	40
IV.1 Introduction	40
IV.2 Component Code	40
IV.2.1 Gyroscope code	40
IV.2.2 Mindwave	44
IV.2.3 Ultrasonic	46
IV.2.4 Steppers motor	47
IV.2.5 Communication code	48
IV.2.6 Bluetooth	48
IV.2.7 ESP-NOW	50

IV.2.8 Serial communication	54
IV.3 Main code	56
IV.3.1 ESP32 Transmitter	56
IV.3.2 ESP32 Receiver	56
IV.3.3 Arduino Uno	58
IV.4 EXPERIMENTAL AND ANALYTICAL RESULTS	58
IV.5 DISCUSSION	58
IV.6 CONCLUSION	59
CONCLUSION AND FURTHER WORK	59

List of Figures

I.1	Elmer and Elsie robots	4
I.2	The robot mobile "Shakey"	4
I.3	robotic rover Curiosity	5
I.4	Classification of robots	6
II.1	omni wheel mecanum wheel	10
II.2	Wheels Configuration and Posture definition	10
II.3	Parameters of ith wheel	12
II.4	Motions of Omnidirectional platform	17
III.1	prototype 3D design	18
III.2	robot chassis	19
III.3	HC-SR04 sensor	20
III.4	MPU-6050 sensor	22
III.5	acceleromete MEMS technology	22
III.6	gyroscope MEMS technology	23
III.7	mecanum wheel	24
III.8	MindWave Mobile 2 diagram	25
III.9	HC-05 Bluetooth Module	26
III.10	Arduino uno board	28
III.11	ESP32 board	29
III.12	description of the ESP32	29
III.13	A4988 Stepper Motor Driver	30
III.14	A4988 Current Limiting Potentiometer	32
III.15	NEMA17 stepper motor	33
III.16	stepper motor construction	34
III.17	four lead bipolar stepper motor	34
III.18	CNC shield V3.0	35
III.19	CNC v3 Micro-stepping jumper location	36
III.20	li-po battery	36
III.21	Circuit one	37
III.22	Circuit two	38
III.23	Circuit three	38

IV.1 The yaw pitch and roll angles in the human head motion	41
IV.2 gyroscope reading through the serial monitor	44
IV.3 the positioning of EEG electrode on the forehead	45
IV.4 Bluetooth communication	48
IV.5 ESP-NOW communication	50
IV.6 ESP32 MAC adress	51
IV.7 Serial communication	55

List of Tables

- II.1 Robot Parameters 15
- III.1 HC-SR04 sensor pins 21
- III.2 MPU-6050 sensor pins 23
- III.3 HC-05 module pins 27
- III.4 A4988 pins 31
- III.5 Microstepping Resolution Truth Table 31
- IV.1 Experimental and Analytical Results 58

GENERAL INTRODUCTION

In the recent years, The rapid growth of information and communication technologies, has led to increased interaction between human and machines. This situation has given rise to many solutions dealing with serious problems facing humanity and help them to improve their life.

Among these solutions, we can distinguish: wheeled mobile robots and Bio-Electronic devices. While mobile robots have definitively changed the lives of people with disabilities and people with reduced mobility by allowing them to move freely. bio-medical devices have added the ability to control these robots not only by the manipulation of conventional organs like hands or feet, but also by exploiting new body capabilities such as EEG signals and body language.

Our work in this thesis goes exactly in this direction, where we will try to design and implement a prototype of a wheeled mobile robot driven by EEG/gyroscope signals. Our solution, target especially stroke patients. This type of patient often suffers from almost total paralysis but can slightly control the higher parts of their bodies. Thereby, we decided to use two types of sensor: the first one is an EEG headset to captor eye blink EEG signal which will buy used as start/stop command for our robot, and the second is a gyroscope sensor to measure head inclination to command robot direction .

About the robot itself. We have noticed that Direct Current motors used to drive wheels present many disadvantages which we can resume in the next points:

- The lack of precision
- Starts and stops with hard motion
- No possibility to make lateral displacement

To overcome these limitations, we decide to use stepper motors combined with a special type of wheels called "mecanum wheels". Stepper motors offer a better control for the robot motion where macanum wheels add the omnidirectional ability which means that it

can go in any direction without the need to make the rotation.

The communication between the sensors placed on the patient's head and the robot is made through two communication protocols. The first is Bluetooth and the second is ESP-NOW which is a slight modification of the WIFI technology to eliminate the need for an access point. Our choice is motivated by the need for a robust and fault tolerant communication channel between the patient and the robot.

The remaining of the this thesis is organised as follows:

- In the first chapter, we will begin by introducing mobile robots, then we try to identify some important dates and milestones in their historical development. Next we will take a view on their types, classification and standards. After that we will see some of their main functions and use cases. We conclude this chapter by the discussion of their advantages and drawbacks.
- In the second chapter, we focus on the formalization of the motion of mobile robots. We have chosen a well studied mathematical model for forwards kinematics that have been published in international journals. The used model permits to calculate robot motion (velocity in the Cartesian plane and angular rotation of the robot) according to the angular rotation speed of each robot's wheels.
- The third chapter will be interested by the design and the construction of the robot prototype. We will highlight the different electronic and mechanic elements used to build the robot. At the end, We give the diagram of the main circuit and explain how it work.
- In the fourth chapter, we will go through test cases and experimentation. The goal is to adapt the programmed sketches to achieve the robot's better behavior. This chapter includes a various motivations about our choices and some explanations and interpretations of the obtained results.

Finally, we achieve this thesis by a general conclusion where we resume our main contribution with it's pros and cons of and give some orientation for further work.

I.1 INTRODUCTION

In this chapter. We will introduce the robot mobile then identify some important milestones in historic evolution, next in this chapter, we will view mobile robot type and classification standers, then we will see the features and uses of mobile robots and the end with the benefits of mobile robots.

I.2 DEFINITION

Over the years, robotics have largely contributed to the development of human society. Today, mobile robotics is one of the most rapidly growing fields of scientific research. Due to their capabilities, it is a field that integrates a variety of scientific areas such as engineering and scientific disciplines, from mechanical, electrical, and electronics engineering to computer and social sciences.

A mobile robot, it's machine that controlled by software that use sensors and other technology to identify its surroundings to move from one place to another around its environment autonomously, The perception action cycle principle borrowed from biological systems and human cognitive and adaptation capabilities [Tza13].

I.3 HISTORICAL DEVELOPMENT

This section presents the development of wheeled mobile robotics and some important milestones through the history. Focus is given to the application of wheeled mobile robots; however, some other technological achievements that influenced the field are mentioned.

1898: in New York City, Nikola Tesla demonstrated a radio-controlled unmanned boat.

Nikola Tesla holds a patent for the invention.

1948: Elmer and Elsie, the world's first electronic autonomous mobile robots were created by Dr. William Grey Walter in Bristol I.1,

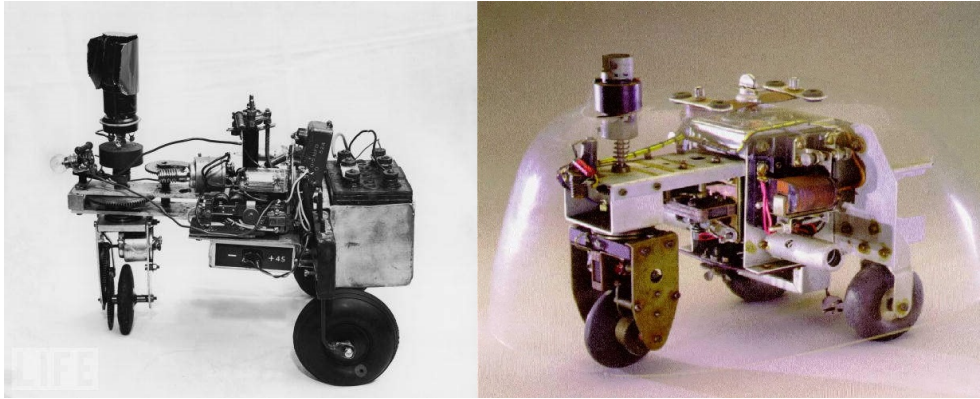


Figure I.1: Elmer and Elsie robots

1961: UNIMATE, the first industrial robot, was introduced at the General Motors factory in New Jersey.

1963: The Johns Hopkins University develops 'Beast'. It used a sonar to move around. When its batteries ran low it would find a power socket and plug itself in.

1970: SRI International introduced Shakey, the first mobile robot controlled by AI.

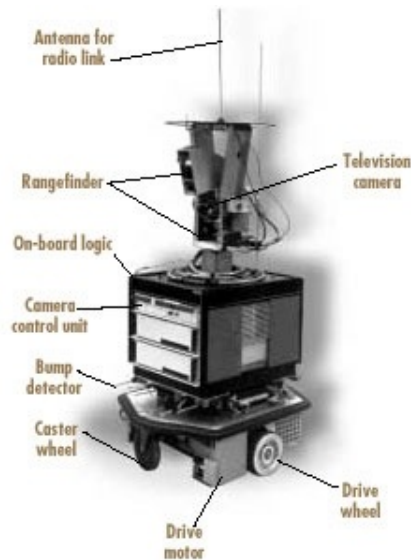


Figure I.2: The robot mobile "Shakey"

1974: the Silver Arm was invented. This was a robotic arm that could perform small-parts assembly by gathering feedback from touch and pressure sensors.

1979: the Stanford Cart crossed a room, using a TV camera to take pictures from multiple

angles and sent them to a computer, without human assistance.

1982: A first model from the series of commercial HERO robots became available, which were mainly targeted for home and educational use.

1980-1989: This decade is dominated by the development of advanced Japanese robots, especially walking robots (humanoids).

1986: The team of Ernst Dieter Dickmanns developed the robot car VaMoRs that was able to drive by itself on streets without traffic, reaching speeds up to 90 km/h.

1986: Stevo Bozinovski and Gjorgji Gruevski control a wheeled robot using speech commands.

1989: Stevo Bozinovski and his team control a mobile robot using EOG signals.

1990-1999: During this decade the emergence of “explorer robots” Dante I and Dante II were developed by Carnegie Mellon University. These robots went where the human did not visit before or it was considered too risky or inconvenient. Both were walking robots.

2005: Boston Dynamics creates a quadruped robot intended to carry heavy loads across terrain too rough for vehicles.

2008: Boston Dynamics released new generation quadruped robot able to walk on icy terrain and recover its balance when kicked from the side.

2012: NASA successfully landed the robotic rover Curiosity on Mars.



Figure I.3: robotic rover Curiosity

2014: Google unveiled their new prototype autonomous vehicle that has no steering wheel and no pedals.

2016: the fleet of Google self-driving vehicles had driven a total of 2,777,585 km in autonomous mode[KZBŠ17a].

I.4 CLASSIFICATIONS OF MOBILE ROBOTS

Mobile robots can be classified in two ways: by the environment in which they work and of the device they use to move[BAM18].

Examples of different environment mobile robots include:

- Polar robots that are designed to traverse icy, uneven environments.
- Aerial robots, also known as unmanned aerial vehicles (UAVs) or drones, which fly through the air.
- Land or home robots, or unmanned ground vehicles (UGVs), that navigate on dry land or within houses.
- Underwater robots, or autonomous underwater vehicles (AUVs), that can direct themselves and travel through water.
- Delivery and transportation mobile robots that are designed to move materials and supplies around a work environment.

A mobile robot can be classified by different devices:

- legs (human-like or animal-like legs).
- tracks.
- wheels.

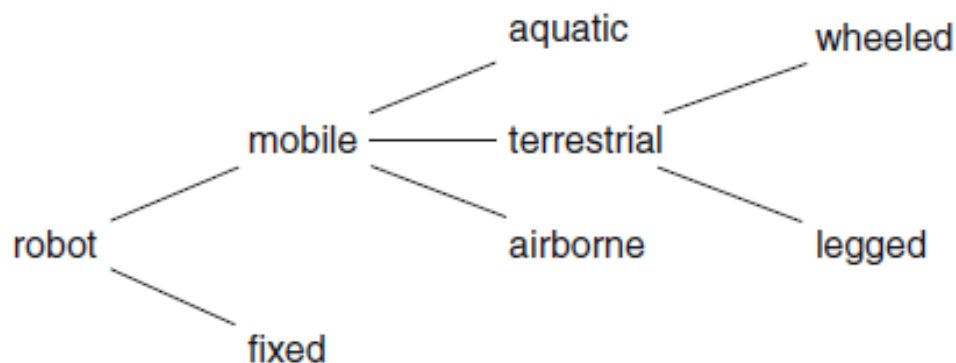


Figure I.4: Classification of robots

I.5 MOBILE ROBOTS TYPES

There are two main types of mobile robots: Automated Guided Vehicle and Autonomous Mobile Robot :

- **Automated Guided Vehicle (AGV)** it's robots that require assistance or some form of instruction or guidance system in order to move such as marking lines or wires on the floor, use radio waves, vision cameras, magnets, or lasers for navigation, Thus creating an inflexible system that is both expensive and difficult to adjust as needs change. They required depots for proper and efficient navigation so that safety can be maintained[SNS11] .
- **Autonomous Mobile Robot (AMR)** they are able to move and explore their surroundings without any sort of external direction using on-boarded sensors and processors AMRs are so smart that they can recognize and at the same time can react to people, cars, and many more. They perform all their jobs safely and are not dependent on how busy the surrounding environment[BAM18].

I.6 FUNCTIONS AND USES OF MOBILE ROBOTS

The basic functions of a mobile robot include the ability to move and explore its environment, transport payloads, or revenue producing cargo, and complete complex tasks using an onboard system, its functions can be used in the medical field like surgery, personal assistance and security like robotic arms. Also the use of mobile robots is popular in industrialization, especially in warehouses and distribution centers, Ocean and space exploration and navigation are also amongst the most common uses of mobile robots.

Mobile robots are being used to access areas where it's too dangerous for humans to inspect and monitor themselves. Such as nuclear power plants, where factors, like high radiation[Jou].

However, mobile robotics must be designed to be able to handle tolerate high radiation otherwise their electronic circuitry will be impacted. Attempts to invent mobile robots to deal specifically with these situations are currently being made.

Other uses of mobile robots include:

- shoreline exploration of mines.
- repairing ships.
- a robotic pack dog or exoskeleton to carry heavy loads for military troopers.

- painting and stripping machines or other structures.
- arms to assist doctors in surgery.
- manufacturing automated prosthetics that imitate the body's natural functions , such as surveilling thermal and other environmental conditions

I.7 ADVANTAGES OF MOBILE ROBOTS

One major advantage of mobile robots is their computer vision capabilities. The complex array of sensors used by mobile robots to detect their surroundings allows them to accurately observe their environment in real time. This is valuable, especially in industrial settings that are constantly changing and shifting.

The onboard intelligence system and AI used by AMRs creates another advantage. The autonomy provided by the mobile robots' ability to learn their surroundings through either an uploaded blueprint or by driving around and developing a map, enables the quick adapting to new environments and assists in the continued pursuit of industrial productivity.

Furthermore, mobile robots are flexible and quick to implement; since they can create their own pathways and easily adapt, possible to break up the implementation into different installations with a modular deployment system and capable of removing the potential for human error by performing easily repeatable tasks, thus improving the safety of a facility or area. The downside of mobile robot is the continued challenges with wireless connections between the robot and information endpoint.

I.8 CONCLUSION

In this chapter, we had the opportunity to see general information about mobile robot, its development through the history, and discover its classification and types also knowing the principle of its work, its advantages, Next in the following chapter we will see the mathematical background of the mobile robot kinematic.

II.1 INTRODUCTION

In this chapter, we introduce the mathematical background to model mobile wheeled robots motion. In wheeled mobile robotics, usually kinematic models are sufficient to design locomotion strategies, while for the other systems such as robots in space, air, or walking robots, dynamic modeling is also needed [KZBS17b]. Later, we will focus on one model for the kinematics of mobile robots with four omnidirectional wheels.

II.2 MECANUM WHEELED MOBILE ROBOT

The key aspect to the omnidirectional robot Mobil is the special wheel and omnidirectional motion.

II.2.1 Omnidirectional motion

Omnidirectional motion is obtained by adding forces from different Mecanum wheels. Because of the angled rollers, the mechanical design is much more difficult, but due to the smoother transfer of contact surfaces a higher loads can be supported [vH05].

II.2.2 Mecanum wheel

Mecanum wheel (Figure II.1b) also called Ilon wheel and Swedish wheel is a more common omnidirectional wheel designs, invented in 1973 by Bengt Ilon a Swedish engineer [DBB⁺02]. In this design, similar to the Omni wheel Figure (II.1a), There are a series of free moving rollers attached to the hub but with an 45° of angle about the hub's circumference but still the overall side profile of the wheel is circular .

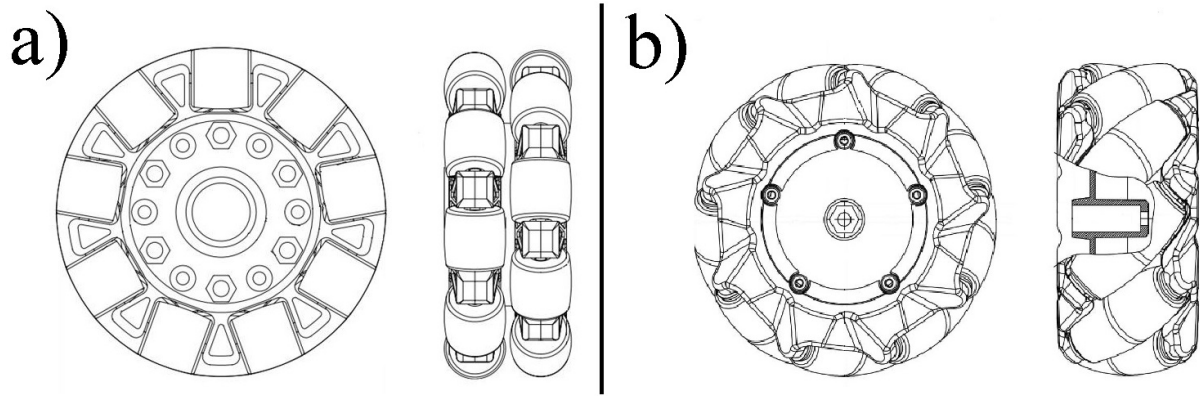


Figure II.1: omni wheel mecanum wheel

II.3 KINEMATICS

The figure II.2 shows different parameters which define the configuration and the position of our four mecanum wheels mobile robot.

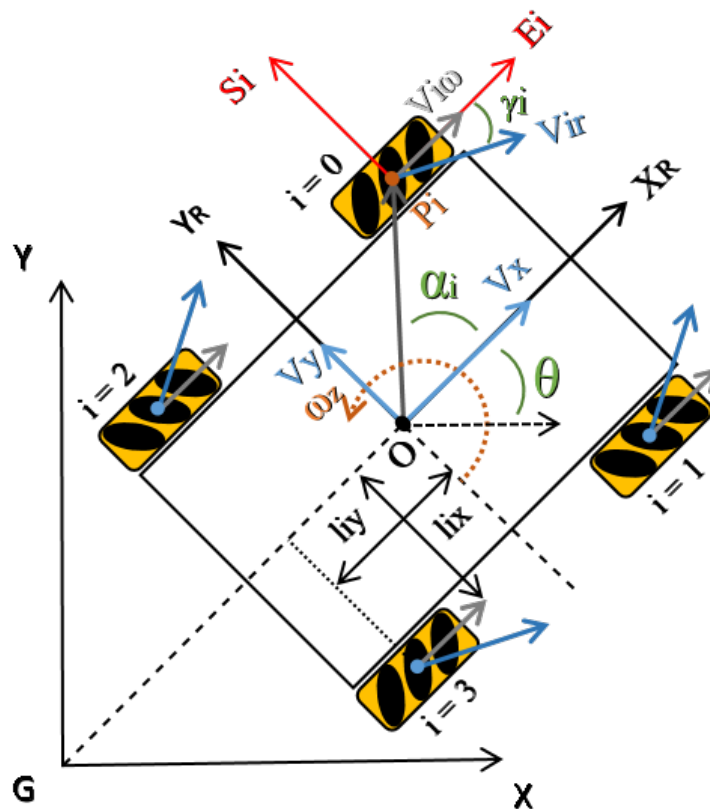


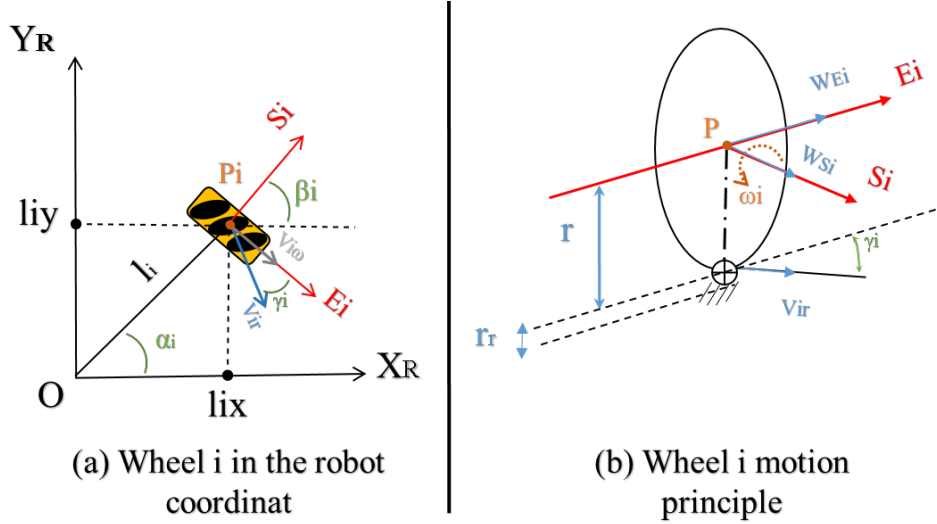
Figure II.2: Wheels Configuration and Posture definition

These parameters are described as follow :

- x, y, θ , robot's position (x, y) and its orientation angle θ (The angle between X and

X_R).

- XGY , inertial frame; x, y are the coordinates of the reference point O in the inertial basis.
- $X_R O Y_R$, robot's base frame; Cartesian coordinate system associated with the movement of the body center.
- $S_i P_i E_i$ coordinate system of i^{th} wheel in the wheel's center point P_i .
- O, P_i the inertial basis of the Robot in Robot's frame and $P_i = \{X_{P_i}, Y_{P_i}\}$ the center of the rotation axis of the wheel i .
- $\overrightarrow{OP_i}$ is a vector which indicates the distance from the centre of the robot to the centre of the wheel. i^{th} .
- l_{ix}, l_{iy}, l_{ix} is the half distance between the front wheels and l_{iy} half the distance between the front wheel and the back wheels.
- l_i distance between wheels and the base (center of the robot O).
- r_i shows the radius of the wheel i (distance between the center of the wheel and the center of the roller).
- r_r denotes the radius of the rollers on the wheels.
- α_i the angle between O, P_i and X_R .
- β_i the angle between S_i and X_R .
- γ_i the angle between v_{ir} and E_i .
- ω_i [rad/s] wheels angular velocity.
- $v_{i\omega}$ [m/s] ($i=0,1,2,3$) $\in R$, Is the speed vector corresponding to the wheel rotation.
- v_{ir} the velocity of the passive roller in the wheel i .
- $\left[W_{si} \quad W_{Ei} \quad \omega_i \right]^T$, Generalized velocity of point P_i in the fram $S_i P_i E_i$.
- $\left[V_{si} \quad V_{Ei} \quad \omega_i \right]^T$, Generalized velocity of point P_i in the fram $X_R O Y_R$.
- $v_x v_y$ [m/s], Robot linear velocity.
- ω_z [rad/s], Robot angular velocity.

Figure II.3: Parameters of the i th wheel

Depending on II.3(b), we can calculate the speed of wheel i and the tangential speed of the freeroll attached to the wheel in contact with the ground:

$$V_{ir} = \frac{1}{\cos\gamma_i} r_i \omega_i, W_{Ei} = r_i \omega_i, i = 0, 1, 2, 3. \quad (\text{II.1})$$

Based on the figure II.3(b) and considering the equations (eq2.1), the velocity of the wheel i in the frame $S_i P_i E_i$ derived by:

$$\begin{aligned} V_{si} &= V_{ir} \sin\gamma_i \\ V_{Ei} &= \omega_i r_i \\ \begin{bmatrix} V_{si} \\ V_{Ei} \end{bmatrix} &= \begin{bmatrix} 0 & \sin\gamma_i \\ r_i & \cos\gamma_i \end{bmatrix} \begin{bmatrix} \omega_i \\ V_{ir} \end{bmatrix} = {}^{\omega_i} T_{Pi} \begin{bmatrix} \omega_i \\ V_{ir} \end{bmatrix}. \end{aligned} \quad (\text{II.2})$$

The matrix of transformation from velocities of the i th wheel to its center:

$${}^{\omega_i} T_{Pi} = \begin{bmatrix} 0 & \sin\gamma_i \\ r_i & \cos\gamma_i \end{bmatrix} \quad (\text{II.3})$$

According to Figure II.3(a) and Figure II.2, the velocity of the wheel's center translated to the $X_R O Y_R$ coordinate system can be achieved by eq 5.

$$\begin{bmatrix} V_{iXR} \\ V_{iYR} \end{bmatrix} = \begin{bmatrix} \cos\beta_i & -\sin\beta_i \\ \sin\beta_i & \cos\beta_i \end{bmatrix} \begin{bmatrix} V_{si} \\ V_{Ei} \end{bmatrix} = {}^{\omega_i} T_{Pi} {}^{Pi} T_R \begin{bmatrix} \omega_i \\ V_{ir} \end{bmatrix} \quad (\text{II.4})$$

$$T' = \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix} \quad (\text{II.5})$$

Then, the transformation matrix from the i th wheel's center to the robot coordinate's system can be obtained from equation 6.

$${}^{P_i}T_R = \begin{bmatrix} \cos\beta_i & -\sin\beta_i \\ \sin\beta_i & \cos\beta_i \end{bmatrix} \quad (\text{II.6})$$

Since the robot's motion is planar, we also have:

$$\begin{bmatrix} V_{iXR} \\ V_{iYR} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = T' \begin{bmatrix} V_{xR} \\ V_{yR} \\ \omega_R \end{bmatrix} \quad (\text{II.7})$$

From (eq.3) and (eq.6), the inverse kinematic model can be obtained:

$$\omega_i T_{P_i} {}^{P_i}T_R \begin{bmatrix} \omega_i \\ V_{ir} \end{bmatrix} = T' \begin{bmatrix} V_{xR} \\ V_{yR} \\ \omega_R \end{bmatrix}, i = 0, 1, 2, 3. \quad (\text{II.8})$$

As $r_i \neq 0, 0 < \gamma_i < \frac{\pi}{2}, \det({}^{P_i}T_R) \neq 0, \det(\omega_i T_{P_i})$ hence, by merging eq 4 and 7 the robot's base velocity (at point O) related to the rotational velocity of the i th wheel can be obtained from eq. 9.

$$\begin{bmatrix} \omega_i \\ V_{ir} \end{bmatrix} = {}^{\omega_i} T_{P_i}^{-1} \cdot {}^{P_i}T_R^{-1} \cdot T' \begin{bmatrix} V_{xR} \\ V_{yR} \\ \omega_z \end{bmatrix}, i = 0, 1, 2, 3. \quad (\text{II.9})$$

Depending to eq.3 and eq.4 there is a relationship between variables in each robot's wheels frames and its center. And with the inverse kinematic, the velocity of the system can be obtained by implementing V_{ir} the linear velocity and ω_i the rotational speed of wheel i th in eq.10 and the contrary in eq.11.

$$\begin{bmatrix} V_{xR} \\ V_{yR} \\ \omega_z \end{bmatrix} = T^+ \begin{bmatrix} \omega_i \\ V_{ir} \end{bmatrix} \quad (\text{II.10})$$

$$\begin{bmatrix} \omega_i \\ V_{ir} \end{bmatrix} = T \begin{bmatrix} V_{xR} \\ V_{yR} \\ \omega_z \end{bmatrix} \quad (\text{II.11})$$

Where : $T = {}^{\omega_i} T_{P_i}^{-1} \cdot P_i T_R^{-1} \cdot T'$, $T^+ = (T^T T)^{-1} T^T$

$$T = \begin{bmatrix} \cos\beta_i & -\sin\beta_i \\ \sin\beta_i & \cos\beta_i \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 & \sin\gamma_i \\ r_i & \cos\gamma_i \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix}$$

Considering the fact that $l_{ix} = l_i \cos\alpha_i$ and $l_{iy} = l_i$, and assuming that the wheels are in a same size, the transformation matrix is:

$$T = \frac{1}{-r} \begin{bmatrix} \frac{\cos(\beta_i - y_i)}{\sin(y_i)} & \frac{\sin(\beta_i - y_i)}{\sin(y_i)} & \frac{l_i \sin(-\alpha_i + \beta_i - y_i)}{\sin(y_i)} \\ -r \cos(\beta_i) & -r \sin(\beta_i) & l_i \sin(-\alpha_i + \beta_i) \\ \frac{\cos(\beta_i - y_i)}{\sin(y_i)} & \frac{\sin(\beta_i - y_i)}{\sin(y_i)} & \frac{l_i \sin(-\alpha_i + \beta_i - y_i)}{\sin(y_i)} \end{bmatrix} \quad (\text{II.12})$$

$$T^+ = \frac{1}{l_i^2 + 1} \begin{bmatrix} -\frac{1}{2}(l_i^2 \sin(\beta_i) - l_i^2 \sin(-\beta_i + 2\alpha_i) + 2\sin(\beta_i))r & \frac{1}{2}l_i^2 \sin(y_i - \beta_i + 2\alpha_i) - \frac{1}{2} \sin(-y_i + \beta_i)l_i^2 - \sin(-y_i + \beta_i) \\ \frac{1}{2}r(l_i^2 \cos(\beta_i) - l_i^2 \cos(-\beta_i + 2\alpha_i) + 2\cos(\beta_i)) & -\frac{1}{2}l_i^2 \cos(y_i - \beta_i + 2\alpha_i) + \frac{1}{2} \cos(-y_i + \beta_i)l_i^2 - \cos(-y_i + \beta_i) \\ \cos(\alpha_i - \beta_i)l_i r & \cos(\alpha_i - \beta_i + y_i)l_i \end{bmatrix} \quad (\text{II.13})$$

Since there is a relation between independent variables V_{ir} and ω_i in each joint and the systems angular and linear velocity, assuming that there is no wheel slipping on the ground, the system inverse kinematic can be obtained by eq.14

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{-1}{r} \begin{bmatrix} \frac{\cos(\beta_1 - \gamma_1)}{\sin\gamma_1} & \frac{\sin(\beta_1 - \gamma_1)}{\sin\gamma_1} & \frac{l_1 \sin(\beta_1 - \gamma_1 - \alpha_1)}{\sin\gamma_1} \\ \frac{\cos(\beta_2 - \gamma_2)}{\sin\gamma_2} & \frac{\sin(\beta_2 - \gamma_2)}{\sin\gamma_2} & \frac{l_2 \sin(\beta_2 - \gamma_2 - \alpha_2)}{\sin\gamma_2} \\ \frac{\cos(\beta_3 - \gamma_3)}{\sin\gamma_3} & \frac{\sin(\beta_3 - \gamma_3)}{\sin\gamma_3} & \frac{l_3 \sin(\beta_3 - \gamma_3 - \alpha_3)}{\sin\gamma_3} \\ \frac{\cos(\beta_4 - \gamma_4)}{\sin\gamma_4} & \frac{\sin(\beta_4 - \gamma_4)}{\sin\gamma_4} & \frac{l_4 \sin(\beta_4 - \gamma_4 - \alpha_4)}{\sin\gamma_4} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} \quad (\text{II.14})$$

eq.15 shows the Jacobian matrix for the system's inverse kinematic:

$$T = \frac{-1}{r} \begin{bmatrix} \frac{\cos(\beta_1 - \gamma_1)}{\sin\gamma_1} & \frac{\sin(\beta_1 - \gamma_1)}{\sin\gamma_1} & \frac{l_1 \sin(\beta_1 - \gamma_1 - \alpha_1)}{\sin\gamma_1} \\ \frac{\cos(\beta_2 - \gamma_2)}{\sin\gamma_2} & \frac{\sin(\beta_2 - \gamma_2)}{\sin\gamma_2} & \frac{l_2 \sin(\beta_2 - \gamma_2 - \alpha_2)}{\sin\gamma_2} \\ \frac{\cos(\beta_3 - \gamma_3)}{\sin\gamma_3} & \frac{\sin(\beta_3 - \gamma_3)}{\sin\gamma_3} & \frac{l_3 \sin(\beta_3 - \gamma_3 - \alpha_3)}{\sin\gamma_3} \\ \frac{\cos(\beta_4 - \gamma_4)}{\sin\gamma_4} & \frac{\sin(\beta_4 - \gamma_4)}{\sin\gamma_4} & \frac{l_4 \sin(\beta_4 - \gamma_4 - \alpha_4)}{\sin\gamma_4} \end{bmatrix} \quad (\text{II.15})$$

And for the forward kinematic according to the eq.10, we have:

$$\begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} T^+ = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (\text{II.16})$$

II.4 THE RELATION BETWEEN MOTIONS AND THE TRANSLATION MATRIX

Analyzing the motion of a four Mecanum wheeled robot brings out the following conclusion: According to the inverse kinematic, there is a relationship between velocities in each joint and the robot's center velocity, thus, the velocity of the robot's center is reflected by and obtained from an individual wheels velocity.

II.5 FOUR MECANUM OMNIDIRECTIONAL SOLUTION

Typical Mecanum four system shown in Figure II.2; the parameters of this configuration are shown in table II.1. In this configuration wheels sizes are the same.

By replacing the parameters of Table II.1 in matrix (eq. 15) and considering eq.14 we

i	α_i	β_i	γ_i	l_i	l_{ix}	l_{iy}
0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$-\frac{\pi}{4}$	1	l_x	l_y
1	$-\frac{\pi}{4}$	$-\frac{\pi}{2}$	$\frac{\pi}{4}$	1	l_x	l_y
2	$\frac{3\pi}{4}$	$\frac{\pi}{2}$	$\frac{\pi}{4}$	1	l_x	l_y
3	$-\frac{3\pi}{4}$	$-\frac{\pi}{2}$	$-\frac{\pi}{4}$	1	l_x	l_y

Table II.1: Robot Parameters

have come up with:

$$T = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \quad (\text{II.17})$$

$$T^+ = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} & -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} \end{bmatrix} \quad (\text{II.18})$$

According to equations (10) and (11) for Forward and Inverse kinematics there is:

$$\begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} \quad (\text{II.19})$$

$$\begin{cases} \omega_0 = \frac{1}{4}(V_x - V_y - (l_x + l_y)\omega) \\ \omega_1 = \frac{1}{4}(V_x + V_y + (l_x + l_y)\omega) \\ \omega_2 = \frac{1}{4}(V_x + V_y - (l_x + l_y)\omega) \\ \omega_3 = \frac{1}{4}(V_x - V_y + (l_x + l_y)\omega) \end{cases} \quad (\text{II.20})$$

and

$$\begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} & -\frac{1}{(l_x+l_y)} & \frac{1}{(l_x+l_y)} \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (\text{II.21})$$

Longitudinal Velocity:

$$V_x(t) = (\omega_0 + \omega_1 + \omega_2 + \omega_3) \cdot \frac{r}{4} \quad (\text{II.22})$$

Transversal Velocity:

$$V_y(t) = (-\omega_0 + \omega_1 + \omega_2 - \omega_3) \cdot \frac{r}{4} \quad (\text{II.23})$$

Angular velocity:

$$\omega_z(t) = (-\omega_0 + \omega_1 - \omega_2 + \omega_3) \cdot \frac{r}{4(l_x + l_y)} \quad (\text{II.24})$$

The resultant velocity and its direction in the stationery coordinate axis (x, y, z) can be achieved by the following equations (eq. 25, 26):

$$\rho = \tan^{-1} \frac{V_y}{V_x} \quad (\text{II.25})$$

$$V_R = \sqrt{V_x^2 + V_y^2} \quad (\text{II.26})$$

II.6 THEORETICAL RESULTS

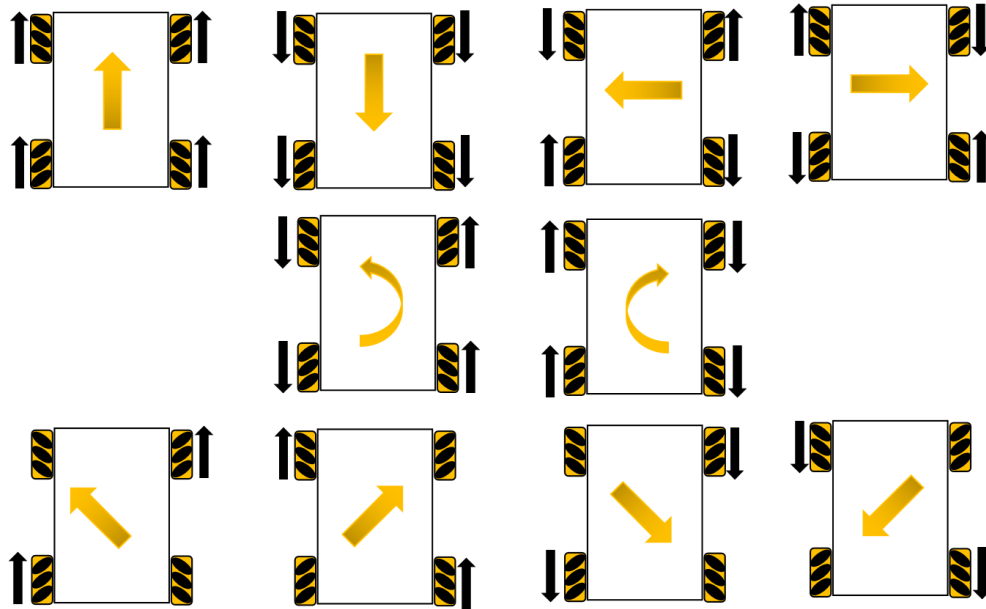


Figure II.4: Motions of Omnidirectional platform

In our test We used four Mecanum wheels and we assumed that there was no wheel slip, Wheel topology was identical to that shown in Figure II.2. The direction and the velocity of the diagonal wheels were set independently. By using the same velocity in each wheel at the same time during the operation we obtained robot movement in eight directions the without changing its orientation, and By changing the velocities of the diagonal wheels we achieved a motion between 0° to 360° . For example, to accomplish a transversal motion to the right, the right wheels were rotated against each other inward, while the left wheels were rotated against each other outward . By using the same technique we achieved all ten different motions shown in Figure II.4 [TQG15].

II.7 CONCLUSION

A mobile platform with 4 omnidirectional wheels was presented in this chapter, The results were systematically obtained by using kinematic equations, The results show that the platform is fully omnidirectional and the robot can reach anywhere between 0° and 360° without changing its orientation.

CHAPTER III

ROBOT PROTOTYPE CONSTRUCTION

III.1 INTRODUCTION

In this chapter, we will see the prototype 3D design ,then we will move on the electronic and mechanic equipment used to construct our project, lastly the interfacing between the electronics component together and the circuit design.

III.2 PROTOTYPE DESIGN

To begin with, we designed this Mecanum Wheels robot using a 3D modeling software. The design is very simple to match the postur show early in CHAPTER II Figure II.2.

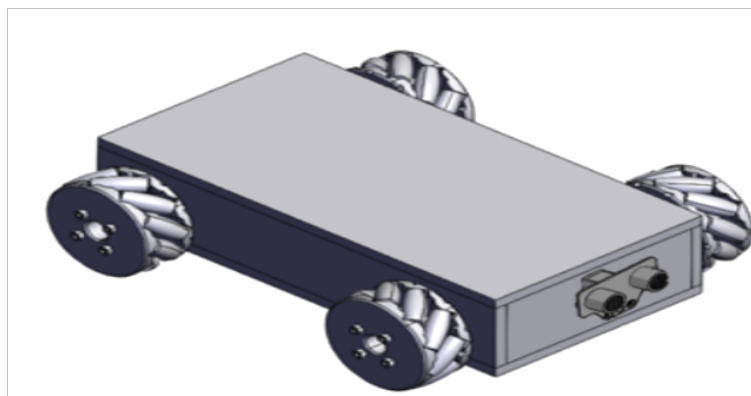


Figure III.1: prototype 3D design

The base platform of this robot is a simple box which we will make out of MDF boards .



Figure III.2: robot chassis

The robot parameters are as follows :

- $\alpha_i = 45^\circ$;
- $l_{iy} = 9.3$ cm ;
- $l_{ix} = 13.5$ cm ;
- $l_i = 16.3$ cm ;
- $r_i = 3.5$ cm ;
- $r_r = 1.5$ cm;

III.3 EQUIPMENT USED

III.3.1 HC-SR04 sensor

The HC-SR04 is an ultrasonic sensor uses SONAR to measure the distance between the sensor and an object. It provide non-contact detection with high accuracy and stable reading range from 2 cm to 400 cm, It emits an ultrasound at 40kHz that travels through the air and if there is an object or obstacle on its path the emitted wave will bounce back to the module, The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module [HC-].

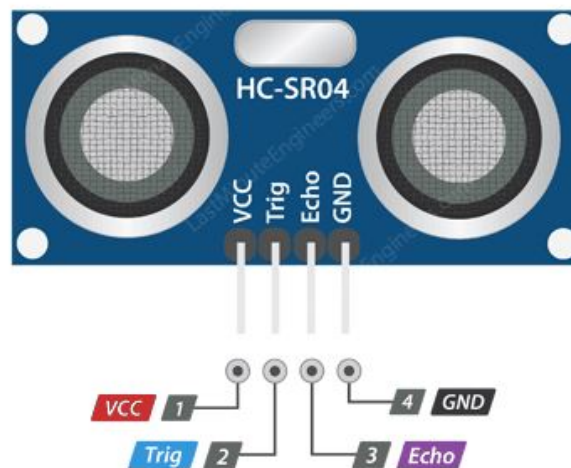


Figure III.3: HC-SR04 sensor

Considering the travel time and the speed of the sound we can calculate the distance using the following equation:

$$D = \frac{V_s}{2} * t$$

Where:

- D: distance between the module and the obstacle.
- Vs: Speed of Sound = 340 m/s.
- t: travel time .

The sound wave travel forward and bounce backward. So in order to get the distance we divide it by 2.

Note: this equation for a module fixed in position, it a case where the module is moving ,the change in distance need to be taken in consider .

III.3.1.1 Sensor pinout

the talbe III.1 show the number , name and the description of HC-SR04 pins

Pin number	Pin name	Description
1	V_{cc}	Connects to 5V of positive voltage for power
2	TRIG	A pulse is sent here for the sensor to go into ranging mode for object detection
3	ECHO	The echo sends a signal is returned, an object has been detected.
4	GND	Connected to ground of system

Table III.1: HC-SR04 sensor pins

III.3.1.2 Sensor features

the following are some important features HC-SR04 offers :

- Effectual Angle : $<15^\circ$
- Ranging Distance : 2cm – 400 cm
- Resolution : 0.3 cm
- Measuring Angle : 30°

III.3.2 MPU-6050

The MPU-6050 is a Gyroscope module designed for Motion Tracking, this module can be used to measure velocity, orientation, acceleration, displacement and other motion like features. It has Auxiliary I2C bus that can be used for communicate with the MCUs. The

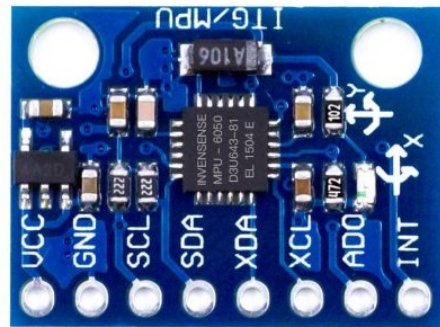


Figure III.4: MPU-6050 sensor

MPU-6050 is a 6-axis Motion Tracking device that combines a 3-axis accelerometer and 3-axis gyroscope both use Micro-electromechanical systems (MEMS)[MPU].

MEMS technology: it is a process technology used to create tiny integrated devices or systems that combine mechanical and electrical components.

3-axis accelerometer:using MEMS technology it detect Acceleration along the axes deflects the movable mass as shown in figureIII.5.

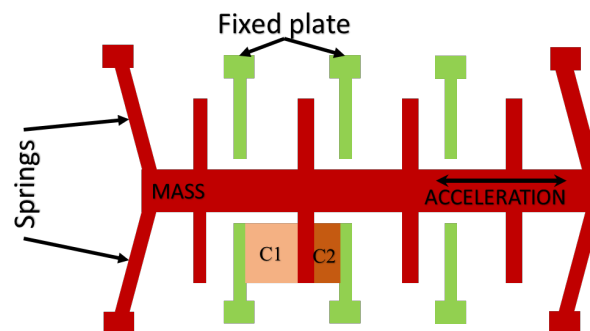


Figure III.5: acceleromete MEMS technology

It measures acceleration through the measurement of capacity change. With mass attached to a spring it move along one direction and fixed outer plates. So the mass will move when an acceleration applied and the capacitance between the plates and the mass will change. This change in capacitance will be measured, processed and it will correspond to a particular acceleration value.

3-axis gyroscope: When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEMS inside MPU-6050. It detect rotational velocity along the 3 axes as shown in figure III.6.

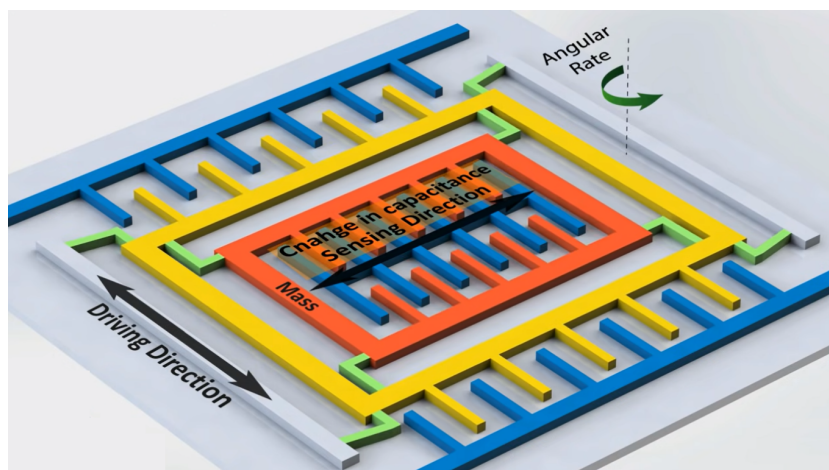


Figure III.6: gyroscope MEMS technology

Similar to the accelerometer, this displacement will cause change in capacitance which will be measured, processed and it will correspond to a particular angular rate [Inv15].

III.3.2.1 Sensor Pinout

In the table III.4 will explain the use of each pin in order to use the component in an efficient and safe method:

Pin number	Pin name	Description
1	V_{cc}	Connects to 5V of positive voltage for power.
2	GND	Connected to ground of system.
3	SCL	Serial clock, Used for providing clock pulse for I2C communication.
4	SDA	Serial data ,used for data transferring through I2C communication.
5	XDA	Auxiliary serial data, can be used to interface other I2C module with MPU-6050.
6	XCL	Auxiliary. Can be used to interface other I2C module with MPU-6050.
7	AD0	This pin used to vary the address of the MPU-6050 if more than one are being used .
8	INT	Interrupt pin , to indicate that data is available for MCU to read .

Table III.2: MPU-6050 sensor pins

III.3.2.2 Sensor Features

The gyroscope in the MPU-6050 includes a wide range of features, the following are some important features:

- Digital-output X, Y, and Z Axis angular rate sensors .
- Improved low-frequency noise performance .
- Digitally-programmable low-pass filter .
- Orientation detection and signaling .
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes[Inv15].

III.3.3 Mecanum wheels

A Mecanum wheel III.7 is a wheel with rollers fixed on its outer rim. The axis of each of the rollers forms the same angle γ ($0 < \gamma < 90$) with the plane of the wheel. As a rule, the angle γ is equal to 45° . Each roller may rotate freely about its axis, while the wheel may roll on the roller. This makes the wheel exert force in diagonal direction when moving forward or backward.

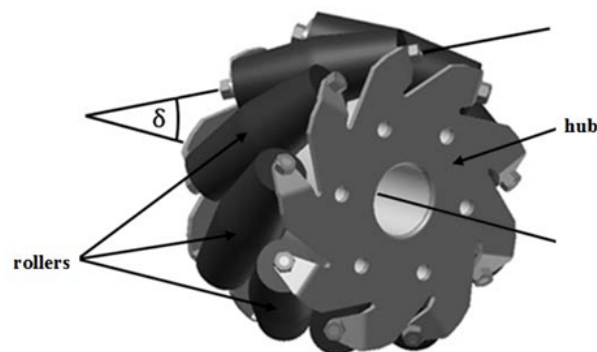


Figure III.7: mecanum wheel

The advantage of using Mecanum wheels is in the platform the robot can achieve omnidirectional motion without changing its orientation.

III.3.4 Neurosky mind-wave mobile 2

MindWave Mobile 2 headset III.8 is an EEG device, it's the culmination of decades of EEG biosensor technology research all in one easy-to-control, wearable package. The MindWave

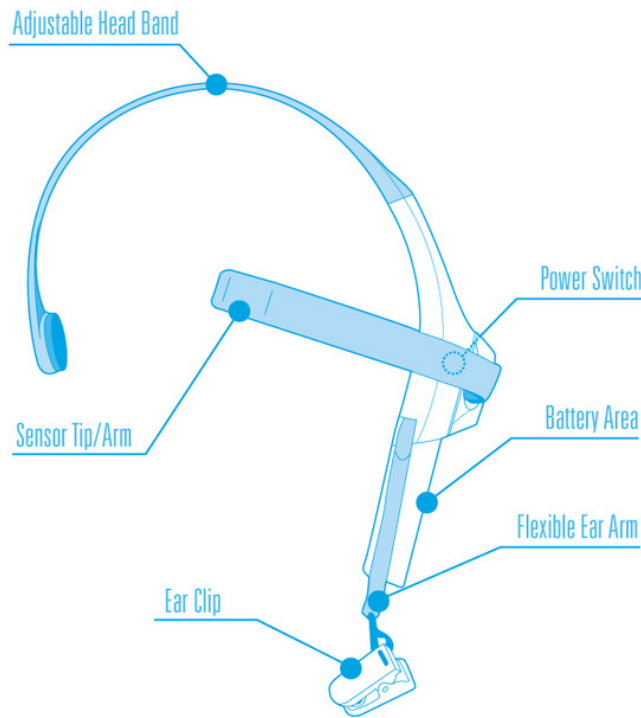


Figure III.8: MindWave Mobile 2 diagram

Mobile 2 safely measures and outputs the EEG power spectrums (alpha waves, beta waves, etc.), NeuroSky sense meters (attention and meditation) and eye blinks. The device consists of a headset, an ear-clip, and a sensor arm. The headset's reference and ground electrodes are on the ear clip and the EEG electrode is on the sensor arm, resting on the forehead above the eye (FP1 position). It uses a single AAA battery with 8 hours of battery life[Min].

III.3.4.1 Device features

- Automatic wireless pairing.
- 8 hours battery run time.
- BT/BLE dual mode module 10m range.
- Static Headset ID (headsets have a unique ID for pairing purposes).

III.3.5 HC-05 module

HC-05 Is a Bluetooth card designed for two-way (full-duplex) wireless communication it has a great data transfer rate and very easy to implement, the transfer done through a wireless serial connection which facilitates the interface with the PC or the controller.

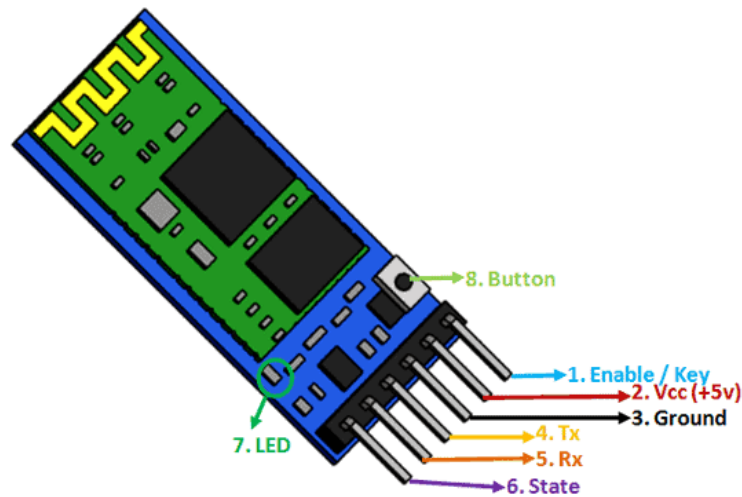


Figure III.9: HC-05 Bluetooth Module

III.3.5.1 Module features

- Range: <100m
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Serial Bluetooth module for Arduino and other MCUs.
- Works with Serial communication (USART) and TTL compatible.
- Can operate in Master, Slave or Master/Slave mode [ITe10] .

III.3.5.2 Module pinout

Pin number	Pin name	Description
1	Key/Enable	Mode switch input, This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode.
2	V_{cc}	Connects to 5V of positive voltage for power.
3	GND	Connected to ground of system.
4	TX	Transmits Serial Data, Everything received via Bluetooth will be given out by this pin as serial data.
5	RX	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth.
6	STATE	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module .
8	Buttons	Used to control the Key/Enable pin to toggle between Data and command Mode .

Table III.3: HC-05 module pins

III.3.6 Microcontroller

III.3.6.1 Arduino uno

The Arduino UNO is an open-source MCU board, The board is equipped with sets of digital and analog input/output (I/O) pins that can be interfaced to various expansion boards (shields), electronic module, sensors, etc... .

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from a PC or a power supply (the power supply must be less than 20V otherwise it will destroy the board). The figure III.10 labeled all of the board components and I/O pins:

- (1) the USB connection is labeled
- (2) and the barrel jack is labeled ,
- (3) is the GND pin,

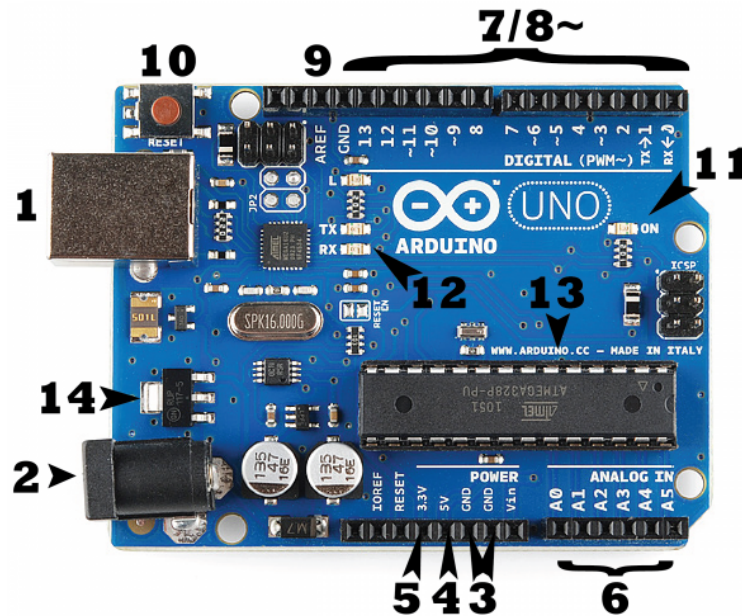


Figure III.10: Arduino uno board

- (4)(5) is the 5V pin and the 3.3V pin both used to supply components,
- (6) are the analog pins A0 through A5, These pins can read the signal from an analog sensor and convert it into a digital value .
- (7) are the digital pins, D0 through D13 ,These pins can be used for both digital input and digital output
- (8) PWM These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM).
- (9) AREF Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.
- (10) Reset Button Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino.
- (11) is Power LED Indicator ,
- (12) TX RX LEDs ,
- (14) is the Main IC Voltage Regulator [Ard],

III.3.6.2 ESP32

The ESP32 is a series of Espressif System (SoC) system-on-chip microcontrollers, it has intergrated WI-FI and dual-mode Bluetooth, This microcontroller is a programmable circuit compatible with the ARDUINO environment. It accepts components offered for ARDUINO. The ESP32 unit is programmed with the ARDUINO IDE interface.



Figure III.11: ESP32 board

On the input and output side, the ESP32 uses GPIO (General Purpose Input/Output) ports which are the pins to serve either input or output depending on the uploaded program.

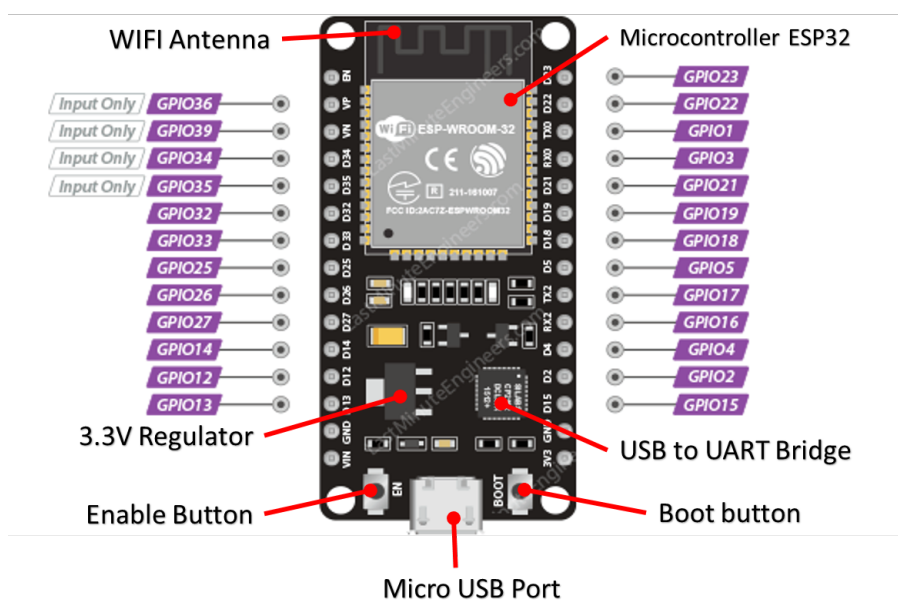


Figure III.12: description of the ESP32

The ESP32 offer many features, the following are the most important:

- **Networking:** The module's Wi-Fi Antenna and dual-core enables embedded devices to connect to routers and transmit data.
- **Data Processing:** Includes processing basic inputs from analog and digital sensors to far more complex calculations.
- **P2P Connectivity:** Creates direct communication between different ESPs. Web Server: Access pages written in HTML or development languages.
- **Security:** IEEE 802.11 standard security features are all supported, including WPA, WPA/WPA2 and WAPI. Moreover, ESP32 has a secure boot and flash encryption[ESp].

III.3.7 A4988 driver

A4988 is a stepper motor driver receives low-level signals from a controller/indexer and converts them into electrical winding currents to run a stepper motor, The A4988 is a steppers motor drive. It is intended to operate bipolar stepper motors from full step (1.8 degree) to sixteenth-step modes (12.5 degree), with an output drive capacity of up to 35V and $\pm 2A$. The A4988 includes build in current regulator. It can control both speed and spinning direction of a bipolar stepper motor like NEMA 17[All14].

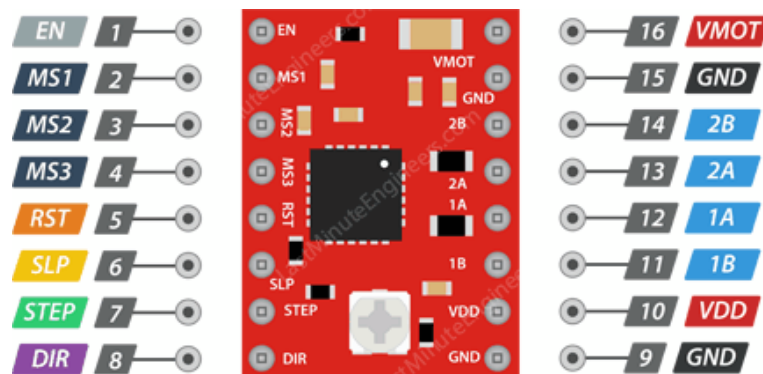


Figure III.13: A4988 Stepper Motor Driver

Pin number	Pin name	Description
1	EN	When this signal is activated, the driver outputs are disabled.
2	MS_1	Step resolution pin 1.
3	MS_2	Step resolution pin 2.
4	MS_3	Step resolution pin 3.
5	RST	Reset signal – Active Low.
6	SLP	This signal is used when the motors are not in use – Active Low.
7	STEP	Digital signal to control rotation steps .
8	DIR	Digital signal to control the direction of motor movement .
9	GND	Connected to ground of system .
10	VDD	Module power supply .
11	1B	Pin 1 for motor coil 2 .
12	1A	Pin 1 for motor coil 1 .
13	2A	Pin 2 for motor coil 1 .
14	2B	Pin 2 for motor coil 2.
15	GND	Connected to ground of system.
16	VMOT	Motor power supply.

Table III.4: A4988 pins

III.3.7.1 Driver Pinout

This Module has the following pins[A49]:

By changing the Step resolution Selection Pins, we can change the step from full step to step 1/16, in the tablesIII.5 High (H) indicates that a jumper is insert and Low (L) indicates that no jumper is inserted[All14]:

MS_1	MS_2	MS_3	Microstep resolution
L	L	L	Full step.
H	L	L	Half step.
L	H	L	Quarter Step.
H	H	L	Eighth Step.
H	H	H	Sixteenth Step.

Table III.5: Microstepping Resolution Truth Table

III.3.7.2 Driver Features and Benefits

- This driver has 200 steps per revolution in full step (1.8 degrees per step).
- Allowable continuous current per phase without cooling: 1A.
- Maximum current per phase with cooling: 2A.
- 3.3 and 5 V compatible logic supply.
- Thermal shutdown circuitry.
- Short-to-ground protection.
- Shorted load protection.
- Five selectable step modes: full, 1/2, 1/4, 1/8, and 1/16.
- Crossover-current protection.

III.3.7.3 A4988 Current Limiting

Before we connect the motor we should adjust the current limiting of the driver so that we are sure that the current is within the current limits of the motor.

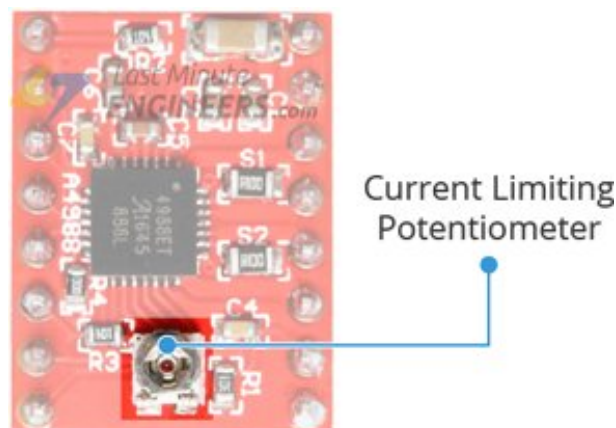


Figure III.14: A4988 Current Limiting Potentiometer

In this method we are going to set the current limit by measuring the current running through the coil.

- We put the driver into full-step mode by leaving the three microstep selection pins disconnected.
- Place the ammeter in series with one of the coils on our stepper motor and measure the actual current flowing.
- Take a small screwdriver and adjust the current limit potentiometer until we reach rated current[A49].

III.3.8 Steppers motor

For this project we need very precise movement for that we chose to use stepper motor over DC motor's, As their name implies, A stepper motor can divide a full rotation into an expansive number of equal steps, unlike DC motor the stepper motor position can be controlled accurately without any feedback mechanism (open loop control), as long as the motor is carefully sized to the application. The stepper motor uses the theory of operation for magnets to make the motor shaft turn a precise distance when a pulse of electricity is provided[NEM].

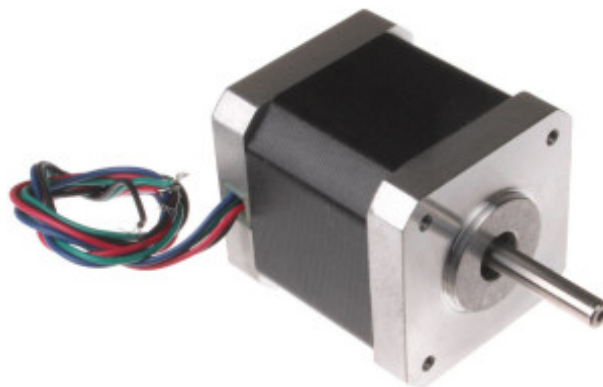


Figure III.15: NEMA17 stepper motor

III.3.8.1 Construction and Working Principle

Stepper motors have a stationary part (the stator) and a moving part (the rotor). On the stator, there are teeth on which coils are wired, while the rotor is either a permanent

magnet or a variable reluctance iron core[NEM].

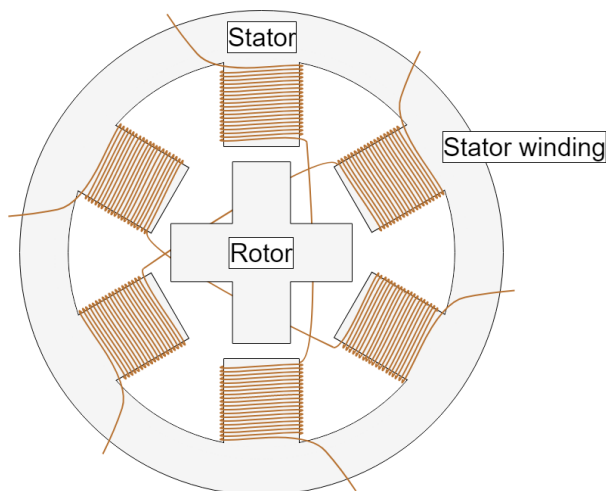


Figure III.16: stepper motor construction

The basic working principle of the stepper motor is the following: By energizing one or more of the stator phases, a magnetic field is generated by the current flowing in the coil and the rotor aligns with this field. By supplying different phases in sequence, the rotor can be rotated by a specific amount to reach the desired final position, in our case we have NEMA 17 48hs42 stepper motor which is bipolar with 4 leads and each phase output pair is independent see figureIII.17.

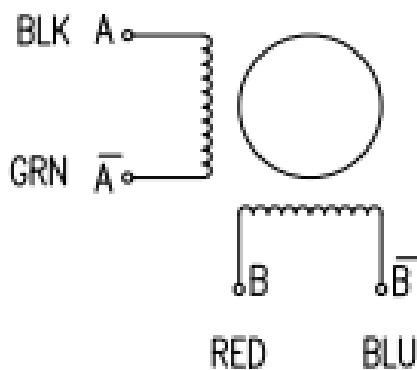


Figure III.17: four lead bipolar stepper motor

III.3.8.2 Motor Features and Benefits

Nema 17 stepper motor specifications

- Step Angle: 1.8° (full step mode).
- Rated current (single phase): 1.7 A.
- DC Rated Voltage: 3.06 V.
- Winding dc resistance (25 C): $1.8 \Omega \pm \%$.
- Winding inductance: $3.8 \text{ mH} \pm 20\%$.
- Holding torque: 560 mN.m ($I=1.7 \text{ A}$).
- Shaft diameter: 5 mm.

To achieve full rated torque, the coils in a stepper motor must reach their full rated current during each step.

III.3.9 Arduino CNC Shield v3.0

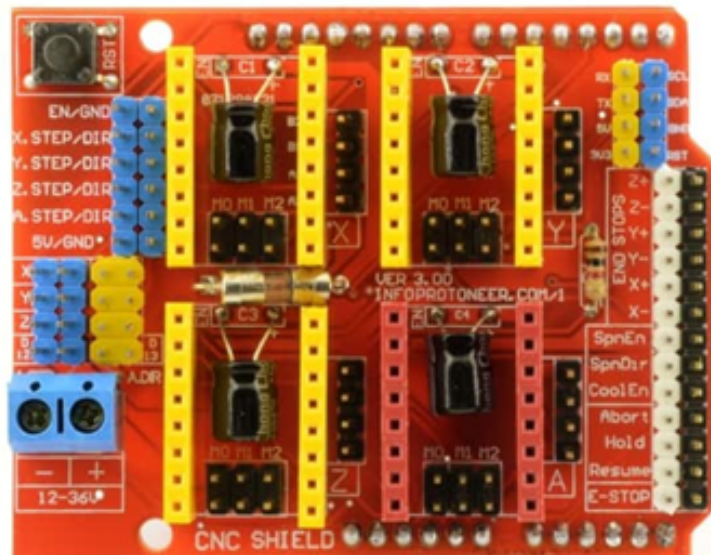


Figure III.18: CNC shield V3.0

The Arduino CNC Shield makes it easy to setup our prototype. It uses open source firmware on Arduino to control 4 stepper motors using 4 pieces of A4988 Stepper Motor

driver without any external jumper wires. also shown this CNC sit nicely on top of Arduino Uno board, without any external jumper wires.

Each steppers motor has 3 jumpers that can be set to select Step resolution for the A4988 plug-in driver board. After setting the micro step jumper, in our case we will leave it without any jumper since we want to use it at full step mode, the location of the micro stepping is shown in the III.19. [CNC18].

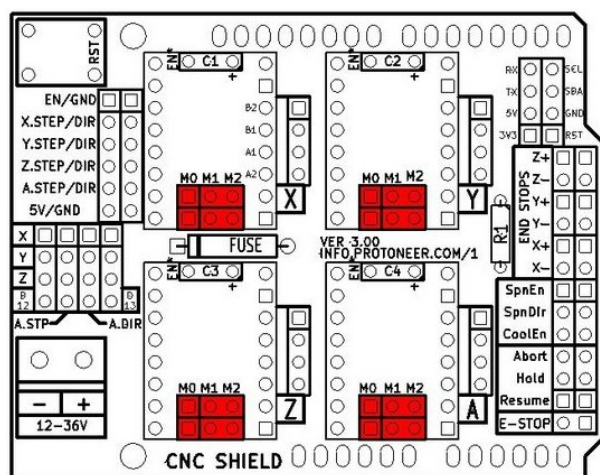


Figure III.19: CNC v3 Micro-stepping jumper location

III.3.10 Power supply

- **Steppers motor power supply** : requires a power supply capable to deliver current flow up 6A, we chose to use LiPo battery with 11.1v voltage output, 3300mAh battery capacity and 35C discharge capacity within one hour , meaning $3300 \times 35 = 115.5 \text{ Ah}$.



Figure III.20: li-po battery

- **Arduino UNO and ESP32 (receiver) Power supply:** don't require high current flow, but must have a voltage output between 5-24, but require high battery capacity for that we chose to use power bank with 3300mAh battery capacity and 5V voltage output.
- **ESP32 (sender) Power supply:** a 9v battery will work fine since it doesn't need neither high current or battery capacity .

III.4 CIRCUIT DIAGRAM

The last step left for as now is to group all the elements together, using application fritzing allowed as to design the electric circuit of the robot, our project consists of 3 separate circuit:

The first circuit contain ESP32 Board with MPU-6050 module.

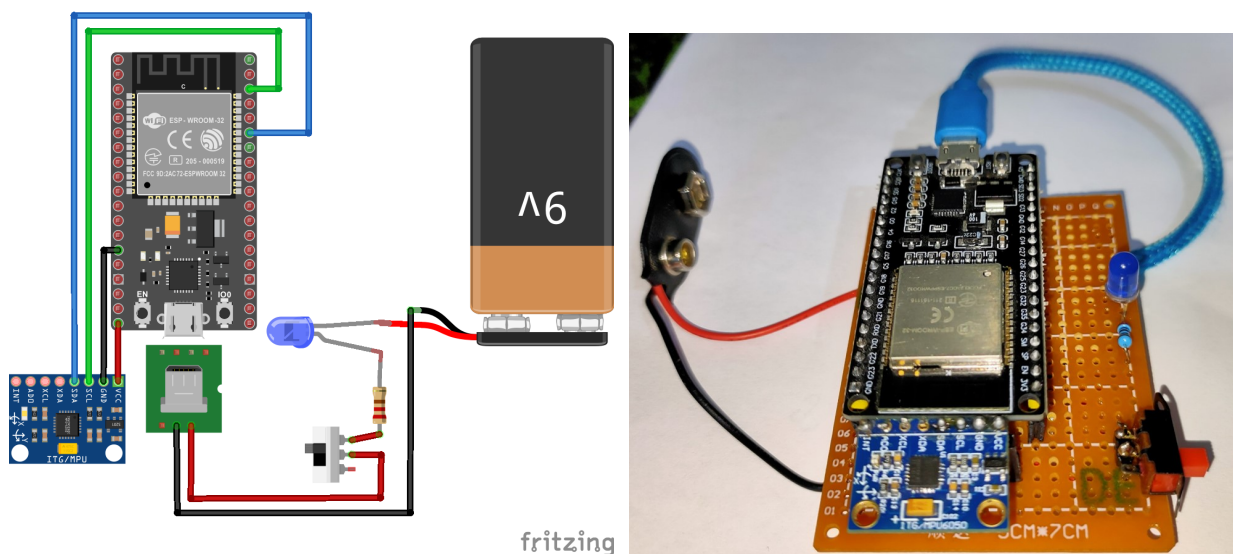


Figure III.21: Circuit one

The second circuit consist of ESP32 board with ultrasonic and Bluetooth module to receive data from MindWave headset.

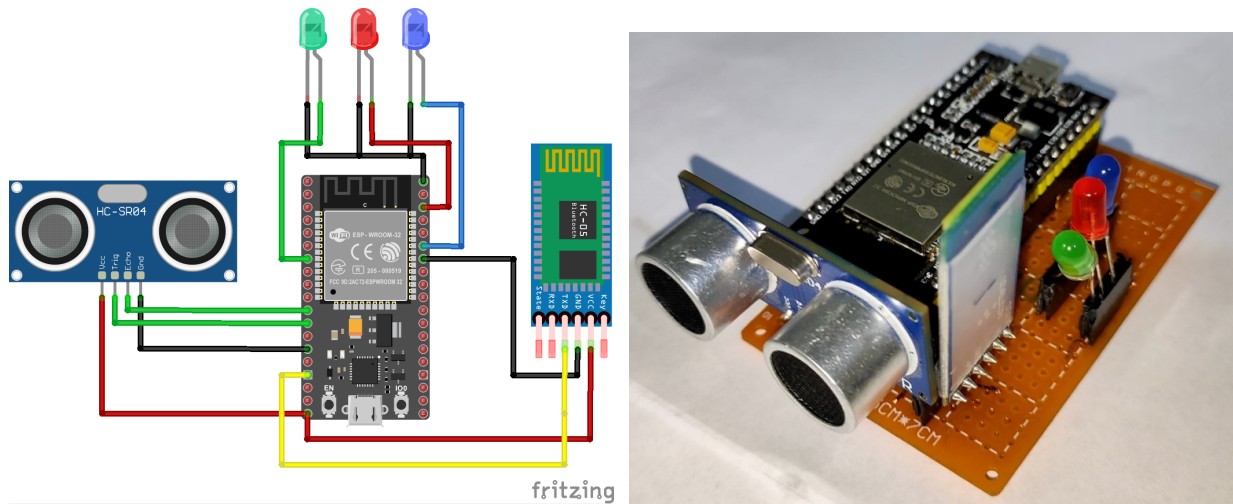


Figure III.22: Circuit two

The third circuit consist of Arduino uno and CNC shield connected with stepper motor driver.

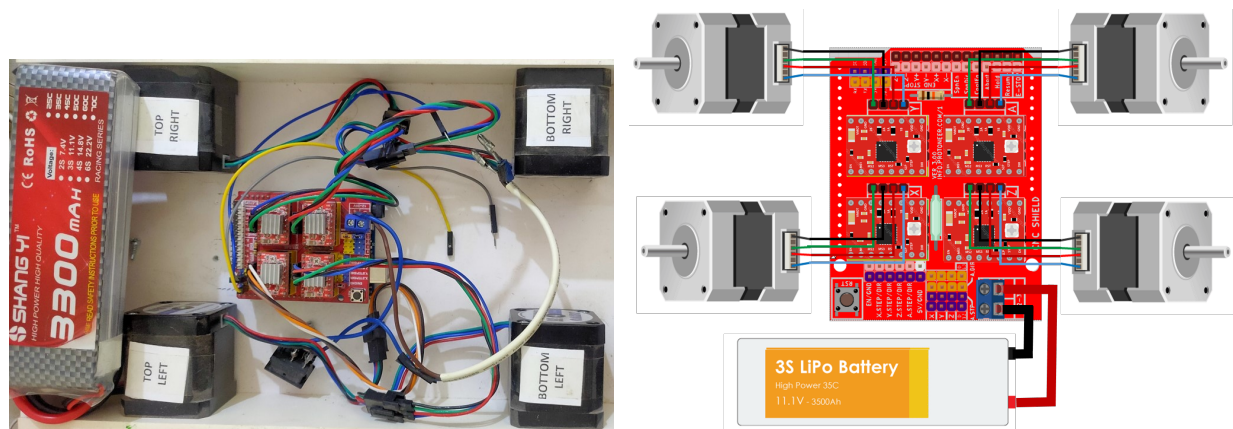


Figure III.23: Circuit three

III.5 CONCLUSION

In this chapter, we started with constructing the robot chassis, next we moved on to the material used to build the prototype and detailed all the information needed to be know to

use those equipment properly, then we ended with the project electric circuit where we saw the interfacing between the electronic component and the MCUs, In the next chapter we will program the robot MCUs to function as we need and use the parameter to calculate it's velocity.

IV.1 Introduction

In This chapter will walk through the programming part, where we will explain the electronic components and communications protocol and use them to write the code for our project, then finish with results interpretation .

IV.2 Component Code

Arduino Integrated Development Environment (IDE) is the application we chose to write the code since it suit the project we will control. In addition, we are going to use some available free libraries and combine them with Arduino IDE for an addition command or control methods.

IV.2.1 Gyroscope code

The Gyroscope will be installed at the top of the user's head along with the MindWave headset, it's going to be used to get the angle of the human head motion to form of yaw, pitch roll.

The following sketch can be used to get the gyroscope reads in form of yaw, pitch roll value.

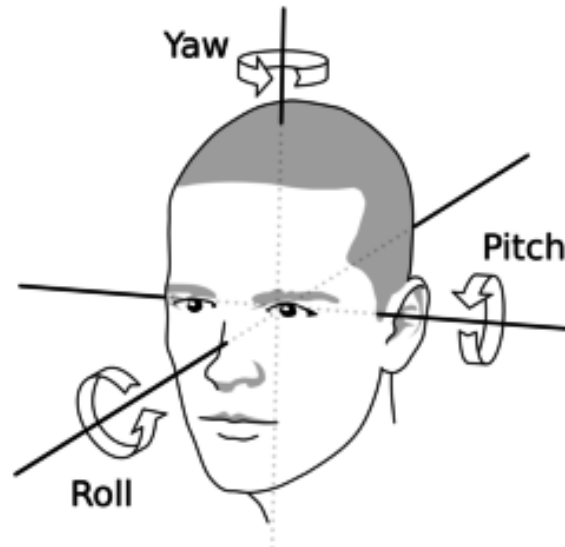


Figure IV.1: The yaw pitch and roll angles in the human head motion

To start with we need to include "Wire. h" library used for I2C communication, then defines the variables used for sorting the data.

```

1 #include <Wire.h>
2 const int MPU = 0x68; // MPU6050 I2C address
3 float AccX, AccY, AccZ;
4 float GyroX, GyroY, GyroZ;
5 float accAngleX, accAngleY, gyroAngleX, gyroAngleY, gyroAngleZ;
6 float ypr [3];
7 float roll , pitch , yaw;
8 float AccErrorX, AccErrorY, GyroErrorX, GyroErrorY, GyroErrorZ;
9 float elapsedTime, currentTime, previousTime;

```

Next on the setup section ; we set the the wire library and reset the sensor.

```

1 void setup() {
2   Serial.begin(19200);
3   Wire.begin();
4   Wire.beginTransmission(MPU);

```

```

5 Wire.write(0x6B);
6 Wire.write(0x00);
7 Wire.endTransmission(true);
8 }

```

Next to the loop section; we read the accelerometer data from the respective registers using function "requestFrom()" we request to read all 6 registers for the X, Y and Z axes. Then we read the data from each register, then we divide the output with the selected sensitivity, which is by default +- 2g for the accelerometer and 250 degrees/s range for the gyroscope so we divide by 16384 and 131 respectively according to the MPU-6050 register map[Inv13].

```

1 void loop() {
2 Wire.beginTransmission(MPU);
3 Wire.write(0x3B);
4 Wire.endTransmission(false);
5 Wire.requestFrom(MPU, 6, true);
6 AccX = (Wire.read() << 8 | Wire.read()) / 16384.0; // X value
7 AccY = (Wire.read() << 8 | Wire.read()) / 16384.0; // Y value
8 AccZ = (Wire.read() << 8 | Wire.read()) / 16384.0; // Z value

```

After that we calculate the roll and pitch angles from the accelerometer data.

```

1 accAngleX = (atan(AccY / sqrt(pow(AccX, 2) + pow(AccZ, 2))) *
   180 / PI) ;
2 accAngleY = (atan(-1 * AccX / sqrt(pow(AccY, 2) + pow(AccZ, 2)))
   ) * 180 / PI) ;

```

Using the same method, we read the gyroscope data from the respective six registers. The output we have right now are in degrees per second (deg/s), multiplying them with time (sec) to receive just degrees (deg).

```

1  previousTime = currentTime; // Previous time is stored before
   the actual time read
2  currentTime = millis(); // Current time actual time read
3  elapsedTime = (currentTime - previousTime) / 1000; // Divide
   by 1000 to get seconds
4  Wire.beginTransmission(MPU);
5  Wire.write(0x43);
6  Wire.endTransmission(false);
7  Wire.requestFrom(MPU, 6, true);
8  GyroX = (Wire.read() << 8 | Wire.read()) / 131.0;
9  GyroY = (Wire.read() << 8 | Wire.read()) / 131.0;
10 GyroZ = (Wire.read() << 8 | Wire.read()) / 131.0;
11 gyroAngleX = gyroAngleX + GyroX * elapsedTime;
12 gyroAngleY = gyroAngleY + GyroY * elapsedTime;

```

Finally, we use a complementary filter, as a downside of the gyroscope, it introduces errors in the output on long term, so we use the data from the accelerometer, 4% in this case, enough to limit the gyroscope drift error over time.

```

1  ypr[0] = yaw + GyroZ * elapsedTime;
2  ypr[1] = 0.96 * gyroAngleY + 0.04 * accAngleY;
3  ypr[2] = 0.96 * gyroAngleX + 0.04 * accAngleX;

```

Lastly, print the value on the serial monitor.

```

1  Serial.print("ypr\t");
2      Serial.print(ypr[0]);
3      Serial.print("\t");
4      Serial.print(ypr[1]);
5      Serial.print("\t");
6      Serial.println(ypr[2]);}

```

Through the serial monitor IV.2. We can read the current value updated very rapidly.

```

COM5
20:41:11.115 -> ypr    6.00   -2.00  -15.00
20:41:11.162 -> ypr    6.00   -2.00  -15.00
20:41:11.162 -> ypr    5.00   -3.00  -16.00
20:41:11.162 -> ypr    5.00   -3.00  -16.00
20:41:11.162 -> ypr    5.00   -4.00  -17.00
20:41:11.162 -> ypr    5.00   -4.00  -18.00
20:41:11.209 -> ypr    5.00   -4.00  -18.00
20:41:11.209 -> ypr    5.00   -4.00  -18.00
20:41:11.209 -> ypr    5.00   -4.00  -17.00
20:41:11.209 -> ypr    5.00   -5.00  -15.00
20:41:11.256 -> ypr    5.00   -6.00  -13.00
20:41:11.256 -> ypr    5.00   -7.00  -11.00
20:41:11.256 -> ypr    6.00   -9.00  -9.00
20:41:11.256 -> ypr    6.00  -10.00  -7.00
20:41:11.256 -> ypr    5.00  -12.00  -6.00
20:41:11.303 -> ypr    5.00  -12.00  -5.00

```

Figure IV.2: gyroscope reading through the serial monitor

IV.2.2 Mindwave

The sketch is simple, using the Bluetooth communication the MindWave device sends the data to the ESP32, then we can display the received data through the serial monitor. Using the following sketch:

```

1 #include <Mindwave.h>
2 Mindwave mindwave;
3 void setup() {
4   Serial.begin(MINDWAVE_BAUDRATE);
5 }
6 void onMindwaveData() {
7   Serial.print("\tquality: ");
8   Serial.print(mindwave.quality());
9   Serial.print("\tattention: ");
10  Serial.print(mindwave.attention());
11  Serial.print("\tmeditation: ");
12  Serial.print(mindwave.meditation());

```

```
13 Serial.print("\tlast update: ");
14 Serial.print(mindwave.lastUpdate());
15 Serial.print("ms ago");
16 Serial.println();
17 }
18 void loop() {
19   mindwave.update(Serial, onMindwaveData);
20 }
```

As the sketch show , this EEG device can track different parameters,from that we chose to use track the quality of signal over the attention and meditation,A major advantage of using this parameter is that it doesn't require time to learn how to use, unlike the other two where it's a must to go through a learning phase to be able to control and use this device properly.

The reason is the quality of signal depend on the positioning of EEG electrode on the forehead, which leads us to the blink detection, an intended blink will move forehead resulting a change in the quality of signal which can be used as a controllable input.



Figure IV.3: the positioning of EEG electrode on the forehead

IV.2.3 Ultrasonic

In order to generate the ultrasound we need to set the Trigger Pin on a High State for 10 μ s to emits a high-frequency sonic waves. which will travel through the air at the speed sound and it will be received in the Echo Pin, the Echo Pin will output the time in μ s the sound wave traveled.

The following sketch use the ultrasonic sensor to measure the distance between the sensor and an object if found :

```
1 #define echoPin 2
2 #define trigPin 3
3 long duration;
4 int distance;
```

At the start we define where the module pin are attached with the MCU then defines the variables we going to use in this case we have "duration" for the duration of sound wave travel and "distance" for the distance measurement, next on the setup section ; we set the trig pin as an output and the echo pin as an input and chose baud rate speed for the serial communication:

```
1 void setup() {
2   pinMode(trigPin , OUTPUT);
3   pinMode(echoPin , INPUT);
4   Serial.begin(9600);
5 }
```

Next on the void loop ; here we start by clearing the tring last condition and set it on LOW for 2 μ s then for 10 μ s as HIGH, after that Reads the echoPin, returns the sound wave travel time in microseconds ,next Calculating the distance and Displays the distance on the Serial Monitor on cm .

```
1 void loop() {
2   digitalWrite(trigPin , LOW);
3   delayMicroseconds(2);
```

```
4  digitalWrite(trigPin, HIGH);
5  delayMicroseconds(10);
6  digitalWrite(trigPin, LOW);
7  duration = pulseIn(echoPin, HIGH);
8  distance = duration * 0.034 / 2;
9  Serial.print("Distance:");
10 Serial.print(distance);
11 Serial.println("cm");
12 }
```

IV.2.4 Steppers motor

In the following sketch, the motor will make one full cycle rotation in a particular direction, As the driver is set on Full Step Mode and our Stepper Motor has 1.8 degrees step angle, or 200 steps, we need to send 200 pulses into the Step Pin to make one full cycle rotation. After this full cycle rotation we will make one second delay, then change the direction of rotation and make full cycle rotations. The direction of the motor can be selected with by setting the dirPin.

```
1  const int stepPin = 3;
2  const int dirPin = 4;
3  void setup() {
4    pinMode(stepPin, OUTPUT);
5    pinMode(dirPin, OUTPUT);
6  }
7  void loop() {
8    digitalWrite(dirPin, HIGH);
9    for(int x = 0; x < 200; x++) {
10     digitalWrite(stepPin, HIGH);
11     delayMicroseconds(500);
```

```

12     digitalWrite(stepPin,LOW);
13     delayMicroseconds(500);
14 }
15 delay(1000); // One second delay
16 digitalWrite(dirPin,LOW);
17 for(int x = 0; x < 200; x++) {
18     digitalWrite(stepPin,HIGH);
19     delayMicroseconds(500);
20     digitalWrite(stepPin,LOW);
21     delayMicroseconds(500);
22 }
23 }

```

IV.2.5 Communication code

The communication is very important in our situation where we have multiple devices that need to communicate together in order to run this project properly, different device need certain method for communication, for that we have to use different methods of communication from which we chose to use Bluetooth, ESP-NOW, and serial communication.

IV.2.6 Bluetooth

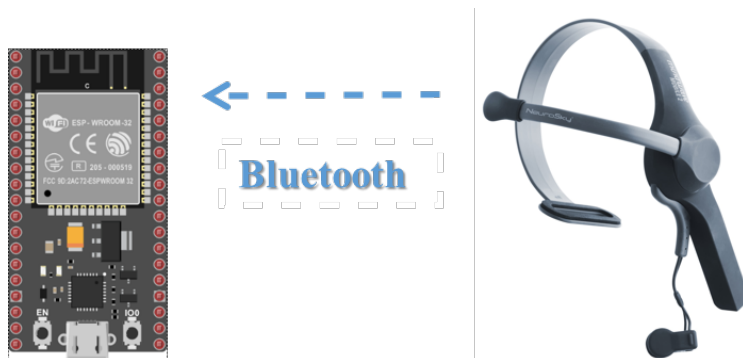


Figure IV.4: Bluetooth communication

The Bluetooth is needed for the communication between the EEG device and the ESP32,

the ESP32 have built in Bluetooth but we are going to use HC-05 Bluetooth module since its way easier to pair with the EEG device compare to the built in Bluetooth, the sketch below is used to pair HC-05 with the mindwave device using AT command :

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial Bluetooth(2,3);
3 void setup()
4 {
5     Serial.begin(9600);
6     Serial.println("ready");
7     Bluetooth.begin(38400);
8 }
9 void loop()
10 {
11     if( Bluetooth.available() ) Serial.write( Bluetooth.read() );
12     if( Serial.available() ) Bluetooth.write( Serial.read() );
13 }
```

In order to pair the two device we need to follow the next steps:

- While pressing the small button over the EN pin on the HC-05, connect HC-05's VCC to ESP32 +5v and keep the button pressed for a few seconds until the red light on the HC-05 begins to blink on for 2 seconds and off for 2 seconds.
- Open the Arduino's Serial Monitor on the computer and set the options in the lower right corner to "Both NL CR" and "38400" baud rate.
- Power-on the MindWave.
- Click inside the input line of the Serial Monitor window. Type AT.
- The response should be "OK" after each command.
- AT+UART=57600,0,0 Defines the baud rate used by the MindWave.

- AT+ROLE=1 Sets the HC-05 as Master rather than Slave.
- AT+PSWD=0000 Sets the password used by the MindWave.
- AT+CMODE=0 To connect the HC-05 to a specific device.
- AT+CLASS=0 Specifies the Class of Device.
- AT+INQM=1,9,48 Sets parameters for pairing.
- AT+INQ to see if the HC-05 recognizes the MindWave.
- AT+PAIR=addr,30 (Strangely, there was no “OK” response here.)
- AT+BIND=addr (The red light on HC-05 still blinks on for 2 seconds and off for 2.)
- AT+LINK=addr
- The red light on the HC-05 should make 2 quick blinks every 3-4 seconds.
- From now and on each time we power the MindWave it connect automatically with the HC-05 module [ITe10].

IV.2.7 ESP-NOW



Figure IV.5: ESP-NOW communication

ESP-Now is a method used for wireless communication between ESP32 boards without using Wi-Fi Access Point, this method require the MAC address for the receiver board ,therefor we going to use this simple sketch to get it done:

Then we create the message structure which will contains the data we want to send, also we create variable of type *struct_mmessage* called "*send_Data*" to store the variables values.

```

1 float send_message_val;
2 String success;
3 typedef struct struct_message {
4     float message;
5 } struct_message;
6 struct_message send_Data;

```

Next , define "onDataSent()" function ,this function will be executed when a message is sent , simply it will prints weather the message was successfully sent or not.

```

1 void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t
    status) {
2     Serial.print("\r\nLast Packet Send Status:\t");
3     Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery
        Success" : "Delivery Fail");
4     if (status == 0) {
5         success = "Delivery Success";
6     else {
7         success = "Delivery Fail";

```

In the steup section ,starting with the serial baud ,then set the device as a WI-FI station and initialize ESP-NOW.

```

1 void setup() {
2     Serial.begin(115200);
3     WiFi.mode(WIFI_STA);
4 Next on the steup section ,
5     if (esp_now_init() != ESP_OK) {
6         Serial.println("Error initializing ESP-NOW");
7         return;

```

Once ESP-Now is successfully initialize, we will register the callback function that will be called when a message is sent. Then, pair with another ESP-NOW device to send data.

```

1  esp_now_register_send_cb(OnDataSent);
2  esp_now_peer_info_t peerInfo;
3  memcpy(peerInfo.peer_addr, broadcastAddress, 6);
4  peerInfo.channel = 0;
5  peerInfo.encrypt = false;
6  if (esp_now_add_peer(&peerInfo) != ESP_OK) {
7      Serial.println("Failed to add peer");
8      return;

```

Finally on the loop section , we send the message via ESP-NOW , the serial monitor will indicate if it Sent with success or there is an Error sending the data .

```

1  void loop() {
2      send_message_val = message;
3      send_Data.message = send_message_val;
4      Serial.println("Send data");
5      esp_err_t result = esp_now_send(broadcastAddress, (
6          uint8_t *) &send_Data, sizeof(send_Data));
7      if (result == ESP_OK) {
8          Serial.println("Sent with success");
9      }
10     else {
11         Serial.println("Error sending the data");

```

Next is the receiver sketch :

it's similar to the transmitters sketch , one thing is the structure massge received Must match the Transmitter structure.

```

1 #include <esp_now.h>
2 #include <WiFi.h>

```



```
3 float receive_message_val;
4 typedef struct struct_message {
5     float message;
6 } struct_message;
7 struct_message receive_Data;
8 void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData
9     , int len) {
10     memcpy(&receive_Data, incomingData, sizeof(receive_Data));
11     receive_message_val = receive_Data.message ;
12 }
13 void setup() {
14     Serial.begin(9600);
15     WiFi.mode(WIFI_STA);
16     if (esp_now_init() != ESP_OK) {
17         Serial.println("Error initializing ESP-NOW");
18         return;
19     }
20     esp_now_register_recv_cb(OnDataRecv);
```

After the pairing is done the connection is safe, if suddenly one of the ESP32 resets or loses power, when it restarts, it will automatically restart the communication with its peers[ESp]

IV.2.8 Serial communication

Serial communication is a wired communication method that requires the use of the RX and TX pin on the MCU board, though it doesn't require the use of the both pins as long as the board use only one of them, so if one of the board is only listening (receiver) it only needs RX pin and same if the board is only talker (Transmitter) it only uses TX pin, the method consists of two simple sketches, one for the receiver and another for the Transmitter.

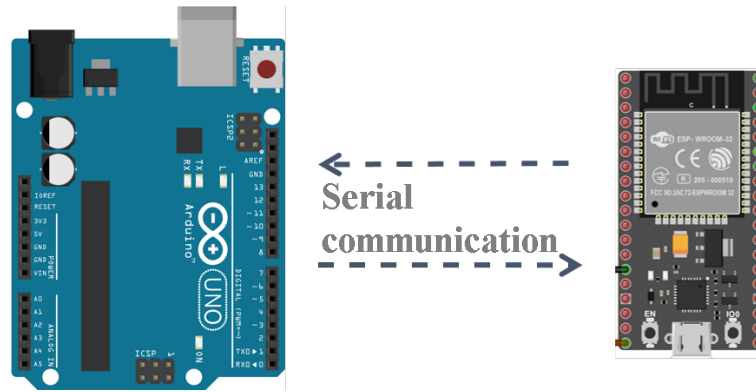


Figure IV.7: Serial communication

starting with the Transmitter sketch .

```

1 void setup() {
2   Serial.begin(9600);
3 }
4 void loop() {
5   Serial.write();
6   delay(1000);
7 }

```

Receiver sketch Must used the same baud rate :

```

1 char message ;
2 void setup() {
3   Serial.begin(9600);
4 }
5 void loop() {
6   if (serial.available()){
7     message = serial.read();
8   }

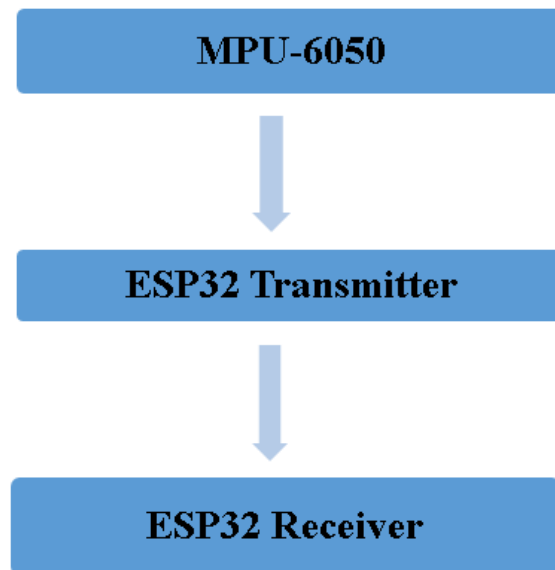
```

IV.3 Main code

Now, after we showcased all the sketch needed for each component and communication method, we will combine them to obtain 3 main code, each code goes to specific MCU.

IV.3.1 ESP32 Transmitter

The first MCU is ESP32, here we are using ESP-NOW and Gyroscope sketch , we start by reading the data coming from the gyroscope Then, using the ESP-NOW we send that data to the receiver board.

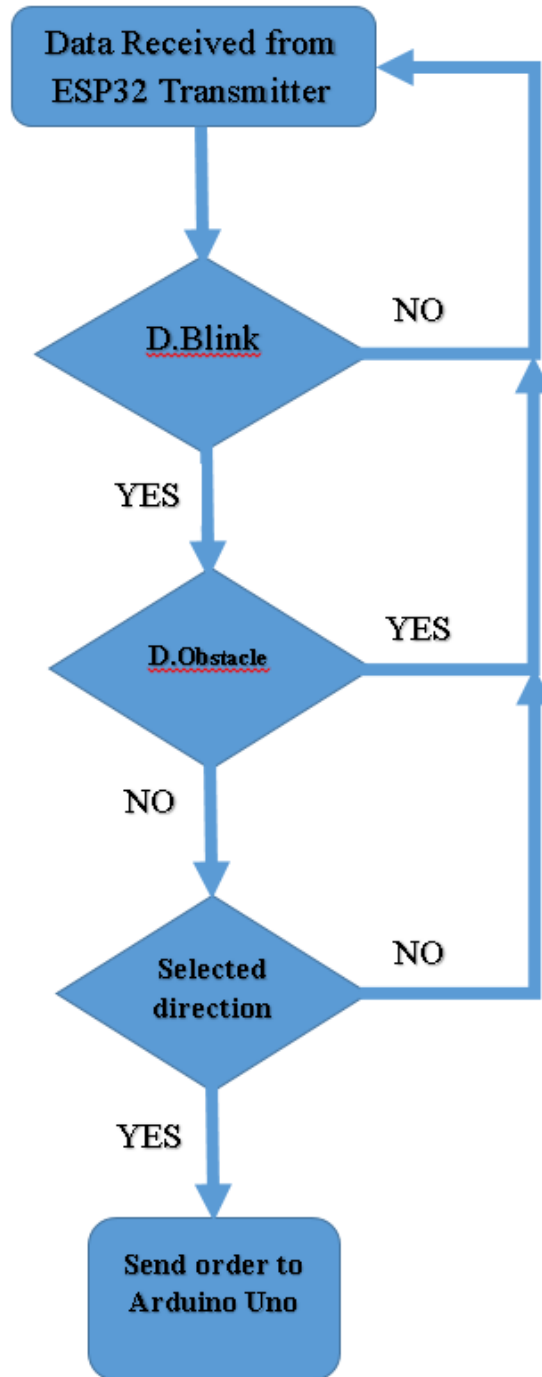


IV.3.2 ESP32 Receiver

The second ESP32 board will receive the Data coming with ESP-NOW and the MindWave data through the Bluetooth communication and check the ultrasonic state .

This board work as follows: firstly we check if there is no obstacle detected, secondly, we check if we have blink detection, then depend if we have direction selected or not, if we have a direction selected, using serial communication this board will send the order to move on the selected direction. Otherwise, if 3 second past and the user didn't select direction,

then the blink detection will be inactive and he has to perform new blink if he decides to move.



For example, if we have blink detection, no obstacle detection and the user moved his head forward, in such a case will send orders to move forward to the Arduino board.

IV.3.3 Arduino Uno

This MCU will receive the data through the serial communication, the data received is a character from which the Arduino board will control the stepper motors direction through the CNC shield and the drivers installed on the shield.

IV.4 EXPERIMENTAL AND ANALYTICAL RESULTS

Using the robot parameters from chapter III , and using the stepper motor driver at full step mode(1.8° per step) and 8 msec delay between each step we achieved wheels angular velocity $\omega_i = 4.45$ rad/s .

Using the equation 22,23,24,25,26 from CHAPTER II and according to Figure II.4 we can calculate the robot velocity,the obtained results are shown in tableIV.1

Direction	$v_x[Cm/s]$	$v_y[Cm/s]$	$\omega_z[Rad/s]$	ρ	$v_R[Cm/s]$
Forward	15.575	0	0	0	15.575
Backward	-15.575	0	0	0	15.575
Left	0	15.575	0	90	15.575
Right	0	-15.575	0	-90	15.575
Right rotation	0	0	0.68	0	0
left rotation	0	0	-0.68	0	0

Table IV.1: Experimental and Analytical Results

IV.5 DISCUSSION

After building our robot, the result was satisfying . But there are some remarks which must be noted:

- The robot will stop receiving information from a headset if it is removed in operating mode, it must be restarted.
- The blinking detection is not accurate.

- There is a time lag of one to five seconds for response time, which makes steering selection not smooth.
- If an obstruction is detected while the robot is moving, it will not stop until it finishes the loop.

IV.6 CONCLUSION

Throughout this chapter, we have exposed the main software parts which control the behavior of our mobile robot. These parts are divided into three categories:

1. Communication Code: three main protocols were exploited :
 - (a) Serial Communication
 - (b) WiFi ESP-Now
 - (c) Bluetooth using AT command
2. Motors command code
3. Control Code : which coordinate between all other parts

Many experiments have been realized. The obtained observations are consistent with the theoretical results given in chapter 2.

CONCLUSION AND FURTHER WORK

The main purpose of this thesis was the development of a prototype of an electrical wheelchair controlled by human MindWave with an accurate and efficient way to assist patients with reduced mobility to enhance their self-care. The reached result was very positive compared to previous works using conventional direct current motors and normal wheels .

As with all scientific research, this work is achieved with some discussions allowing us to highlight our contribution and in the same time to locate the weak point which we will continue to work on in future time in order to enhance the global behavior of our Robot.

Our robot offers the next advantages :

- The chassis is made in a generic way. Therefore, it is adaptable to be reused in any other project requiring mobile wheeled robot.
- Control separately each wheel with an independent stepper motor gives a high degree of control and ensure a very fluid motion.
- It is no more matter to make lateral motion with macanum wheels.
- The Control the motion of our robot is very simple and not requiring to move heavy body organs such as hands or feet. This makes it very convenient for stroke patients.
- Multi modal channels (WiFi, Bluetooth and Serial) and the advanced used protocols ensure a high reliability in communication between the chassis and the remote control parts.

In the other hand, our robot presents the next drawbacks

- The down side is the speed, using steppers motor meant to sacrifice speed. However, conventional DC motor are faster but not accurate and need feedback to eliminate errors unlike stepper motors which are very accurate.
- During the construction of this project, we noticed that the blink detection isn't accurate because it's based on the quality of signal, meaning that some of the forehead movement will be detected as an intended blink, this can be fixed by switching to a better EEG device.
- Despite the capability of the robot to move in any direction (360 degree). In our case, we used only four direction (forward, backward and left, right slide) plus rotation on both direction.

What we have presented in this thesis is only a very small part in this vast technology. Mobile wheeled robots and the control of their motion thought intelligent methods like EEG and gyroscopes steel a new research field. The final step is to switch from prototype to a real electrical wheelchair exploiting the obtained results. We hoped to go in this step, unfortunately, we was constrained by the the thesis limited time.

REFERENCES

- [A49] A4988 stepper motor driver. <https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/>. Accessed: 2022-04-10.
- [All14] Allegro MicroSystems, Inc. *DMOS Microstepping Driver with Translator A4988 and Overcurrent Protection*, 5 2014. Rev. 5.0.
- [Ard] Arduino uno board. <https://learn.sparkfun.com/tutorials/what-is-an-arduino>. Accessed: 2022-04-3.
- [BAM18] Mordechai Ben-Ari and Francesco Mondada. Robots and their applications. In *Elements of robotics*, pages 1–20. Springer, 2018.
- [CNC18] CNC shield V3.0 user manual. *3-Axis CNC/Stepper Motor Shield for Arduino*, 12 2018. Rev. 1.0.
- [DBB⁺02] Olaf Diegel, Aparna Badve, Glen Bright, Johan Potgieter, and Sylvester Tlale. Improved mecanum wheel design for omni-directional robots. In *Proc. 2002 Australasian conference on robotics and automation, Auckland*, pages 117–121, 2002.
- [ESp] Esp32 board. <https://www.nabto.com/guide-to-iot-esp-32>. Accessed: 2022-04-3.
- [HC-] ultrasonic sensor. https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm. Accessed: 2022-03-25.
- [Inv13] InvenSense, Inc. *MPU-6000 and MPU-6050 Register Map and Descriptions*, 3 2013. Rev. 4.2.
- [Inv15] InvenSense, Inc. *MPU-6000/MPU-6050 Product Specification*, 5 2015. Rev. 3.3.

- [ITe10] ITead Studio. *HC-05 Bluetooth module Product Specification*, 6 2010. Rev. 1.0.
- [Jou] A review of mobile robots. <https://journals.sagepub.com/doi/full/10.1177/1729881419839596>. Accessed: 2022-03-25.
- [KZBS17a] Gregor Klančar, Andrej Zdešar, S Blažič, and I Škrjanc. Wheeled mobile robotics. *From Fundamentals Towards Autonomous Systems*. Butterworth-Heinemann, 2017.
- [KZBS17b] Gregor Klančar, Andrej Zdesar, Saso Blazic, and Igor Skrjanc. *Wheeled mobile robotics: from fundamentals towards autonomous systems*. Butterworth-Heinemann, 2017.
- [Min] Neurosky. <https://store.neurosky.com/pages/mindwave>. Accessed: 2022-04-1.
- [MPU] Gyroscope. https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm. Accessed: 2022-03-30.
- [NEM] Stepper motor. <https://www.elprocus.com/stepper-motor-types-advantages-applications/>. Accessed: 2022-04-13.
- [SNS11] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.
- [TQG15] Hamid Taheri, Bing Qiao, and Nurallah Ghaeminezhad. Kinematic model of a four mecanum wheeled mobile robot. *International journal of computer applications*, 113(3):6–9, 2015.
- [Tza13] Spyros G Tzafestas. *Introduction to mobile robot control*. Elsevier, 2013.
- [vH05] Rob PA van Haendel. Design of an omnidirectional universal mobile platform. 2005.