A PROJECT REPORT ON

DESIGN AND IMPLEMENTATION OF MECCANUM WHEEL REMOTE-CONTROLLED VEHICLES USING ESP32 AND PEER TO PEER WIFI COMMUNICATION

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING(IOT)

BY

K.V.S.KARTHIK	(20MH1A4923)
D.KASI VISWANADH	(21MH5A4916)
PALAK RAJ MISHRA	(21MH5A4909)
D.VIJAY SANKAR	(20MH1A4912)
V.V.V.PAVAN ADITYA	(20MH1A4946)

Under the esteemed guidance of
DR.B.Kiran kumar ,M.Tech ,Ph.D
Professor & HOD

DEPARTMENT OF CSE(AIML & IOT)

ADITYA COLLEGE OF ENGINEERING(A)

(Affiliated to JNTU Kakinada and approved by AICTE, New Delhi)ADB road, Surampalem, East Godavari District, A.P.-533437

(2020-2024)

ADITYA COLLEGE OF ENGINEERING(A)

(Affiliated to Jawaharlal Nehru Technological University Kakinada)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(IoT)

CERTIFICATE

This is to certify that the project entitled "**Design and Implementation of Meccanum Wheel Remote Controlled Vehicles Using ESP32 And Peer to Peer Wifi Communication** " submitted by K**.V.S.Kathik (20MH1A4923), D. Kasi Viswanadh (21MH5A4916), Palak raj Mishra (21MH5A4909) , D.Vijay Sankar (20MH1A912) , V.V.V.Pavan Aditya (20MH1A4946)** in partial fulfillment of the degree of Bachelor of Technology in **Computer Science and Engineering(IoT)** to Aditya College of Engineering(A) affiliated to Jawaharlal Nehru Technological University, Kakinada is a record of the bonafide work carried out under my guidance and supervision. The results presented in this project report have not been submitted to any other university or institute for the award of any degree.

PROJECT GUIDE	HEAD OF THE DEPARTMENT
Dr.B.Kiran kumar M.Tech,Ph.D	**Dr.B.Kiran Kumar M.Tech,Ph.D**
Professor & HOD	Professor & HOD
Department of CSE(IOT)	Department of CSE(IOT)
Date :	Date:

External Examiner

DECLARATION

Here we declare that the project titled **DESIGN AND IMPLEMENTATION OF MECCANUM WHEEL REMOTE CONTROLLED VEHICLES USING ESP32 AND PEER TO PEER WIFI COMMUNICATION** has been undertaken by us. This work has been submitted to Aditya College of Engineering(A), Surampalem, in partial fulfillment for the award of Degree of Bachelor of Technology in Computer Science and Engineering(IoT).

We further declare that this project work has not been submitted in full or part for the award of any degree of this on any other educational institution.

Project Association

K.V.S.Karthik	(20MH1A4923)
D.Kasi Viswanadh	(21MH5A4916)
Palak Raj Mishra	(21MH5A4909)
D.Vijay Sankar	(20MH1A4912)
V.V.V.Pavan Aditya	(20MH1A4946)

ACKNOWLEDGEMENT

In the name of all mighty most beneficent and merciful, first of all, we want to praise the lord for helping us in all stages of this project work.

We express our gratefulness and a deep sense of gratitude to our project guide and Head of the Department of CSE (AIML & IOT) **Dr.B.Kiran Kumar** Professor of Aditya College of Engineering for his valuable suggestions in planning and successful completion of the project his constant encouragement, help and review of entire work during the course of the work is invaluable. we would like to express our gratitude for his cooperation in providing the necessary facilities to carry out our project work.

We extend our gratitude towards our principal **Dr. A. Ramesh** for providing us with excellent facilities to pursue our project effectively in the stipulated time.

We would also like to express our gratitude towards all the lab technicians for helping us a lot throughout our project and completing it on time.

Finally, we are thankful to one and all who contributed to our project directly or indirectly.

Project Association

K.V.S.Karthik (20MH1A4923)

D.Kasi Viswanadh (21MH5A4916)

Palak Rai Mishra (21MH5A4909)

D.Vijay Sankar (20MH1A4912)

V.V.V.Pavan Aditya (20MH1A4946)

ABSTRACT

This Project explores the cutting-edge design and implementation of mecanum wheel-equipped remote-controlled vehicles, harnessing the capabilities of the ESP32 microcontroller and establishing seamless communication through peer-to-peer WiFi networks.

Mecanum wheels, known for their omnidirectional movement, provide enhanced maneuverability. The ESP32 microcontroller acts as the central processing unit, orchestrating precise control of each mecanum wheel for unparalleled agility.

The study delves into hardware intricacies, including mecanum wheel assembly, motor control interfaces, and sensor integration. A peer-to-peer WiFi communication protocol is implemented, facilitating efficient data exchange between the remote control unit and the vehicle, enhancing reliability and range.

Experimental results highlight dynamic maneuverability and responsiveness, showcasing the system's suitability for applications like surveillance, exploration, and entertainment.

This research contributes to mecanum wheel robotics by introducing an innovative approach, leveraging ESP32 microcontrollers and peer-to-peer WiFi communication, opening possibilities for agile and versatile remote-controlled vehicles across diverse domains from industrial automation to educational platforms

INDEX

CONTENT **PAGE NO**

CONTENT	PAGE NO

LIST OF FIGURES

1.INTRODUCTION

1.1 HISTORY OF MECCANUM WHEELS:

Meccanum wheels, also known as Ilon wheels or Swedish wheels, were invented by Bengt Ilon in the 1970s. Ilon, a Swedish engineer, developed these wheels to create a more versatile and maneuverable platform for vehicles, particularly for industrial and robotic applications. The name "Meccanum" is derived from the Swedish word "mekanum," meaning mechanical. These unique wheels allow for omnidirectional movement, enabling a vehicle to move in any direction without changing its orientation.

The design of Meccanum wheels involves a series of rollers mounted at an angle around the circumference of the wheel. These rollers are typically set at a 45-degree angle to the wheel's axis, with each roller being independently driven by a motor. By controlling the speed and direction of rotation of each roller, the vehicle can achieve omnidirectional movement. When the rollers on one side of the vehicle rotate in one direction while those on the opposite side rotate in the opposite direction, the vehicle moves forward or backward. Similarly, if the rollers on one diagonal rotate in one direction while those on the other diagonal rotate in the opposite direction, the vehicle can move sideways. By varying the speeds and directions of the rollers, the vehicle can perform complex maneuvers such as rotating in place or moving in curved paths.

Mecanum wheels have found applications in various fields, including material handling, warehouse automation, mobile robotics, and even some special-purpose vehicles. They offer advantages in terms of maneuverability, precision, and versatility compared to traditional wheeled or tracked vehicles. However, they also have some limitations, such as reduced traction on certain surfaces and higher complexity in control systems compared to conventional wheels.

Despite these limitations, Mecanum wheels continue to be used and researched for their unique capabilities in enabling omnidirectional movement, making them an important innovation in the field of robotics and vehicle engineering.

1.2 NATURE INSPIRATION:

The design of Mecanum wheels is inspired by nature, specifically by the movement of certain creatures such as holonomic insects and animals with specialized appendages. While not directly modeled after any particular organism, the concept of omnidirectional movement found in Mecanum wheels draws inspiration from the natural world's diverse locomotion mechanisms.

1. **Insects and Arthropods**: Meccanum wheels mimic the movement of insects and arthropods that possess legs capable of moving in multiple directions. For example, insects like cockroaches and beetles can move swiftly in various directions by coordinating the movement of their legs. Mecanum wheels utilize a similar principle by independently controlling the rotation of each roller, allowing for omnidirectional movement.
2. **Mollusks and Cephalopods**: Certain mollusks, such as octopuses and squid, possess tentacles or arms that provide them with exceptional maneuverability in water. These creatures can move in any direction by adjusting the movement of their flexible appendages. Mecanum wheels, with their independently controlled rollers, replicate this flexibility on land, enabling vehicles to move with similar versatility.

1.3 OBJECTIVE OF WORK:

This Project explores the cutting-edge design and implementation of meccanum wheel-equipped remote-controlled vehicles, harnessing the capabilities of the ESP32 microcontroller and establishing seamless communication through peer-to-peer WiFi networks. Meccanum wheels, known for their omnidirectional movement, provide enhanced maneuverability. The ESP32 microcontroller acts as the central processing unit, orchestrating precise control of each meccanum wheel for unparalleled agility.

The study delves into hardware intricacies, including meccanum wheel assembly, motor control interfaces, and sensor integration. A peer-to-peer WiFi communication protocol is implemented, facilitating efficient data exchange between the remote control unit and the vehicle, enhancing reliability and range. Experimental results highlight dynamic maneuverability and responsiveness, showcasing the system's suitability for applications like surveillance, exploration, and entertainment. This research contributes to meccanum wheel robotics by introducing an innovative approach, leveraging ESP32 microcontrollers and peer-to-peer WiFi communication, opening possibilities for agile and versatile remote-controlled vehicles across diverse domains from industrial automation to educational platforms.

1. **Omnidirectional Movement**: The primary goal is to enable the vehicle to move in any direction with ease and precision. Meccanum wheels provide omnidirectional movement by independently controlling the speed and direction of each wheel's rollers. This allows the vehicle to navigate tight spaces, execute complex maneuvers, and change direction seamlessly.
2. **Remote Control via WiFi**: Utilizing ESP32 WiFi peer-to-peer communication enables remote control of the vehicle without the need for a direct internet connection or a separate router. This approach allows for real-time communication between the remote control device (such as a smartphone or a computer) and the vehicle, facilitating intuitive control and responsiveness.
3. **Stability and Maneuverability**: The design and implementation should prioritize stability and maneuverability. Meccanum wheels offer stability by distributing the vehicle's weight evenly and minimizing slippage during movement. Additionally, the vehicle should be capable of precise maneuvers, such as rotating in place and moving sideways, to navigate various environments effectively.
4. **Integration with ESP32**: The implementation should leverage the capabilities of the ESP32 microcontroller, including its built-in WiFi functionality and ample processing power. The ESP32 can serve as the central control unit of the vehicle, responsible for receiving commands from the remote control device, interpreting user inputs, and coordinating the movement of Meccanum wheels accordingly.
5. **User Interface**: Designing a user-friendly interface for the remote control device is essential. The interface should allow users to intuitively control the vehicle's movement, adjust speed and direction, and possibly access additional features or sensor data. This can be achieved through a custom mobile application, a web-based interface, or other interactive methods.
6. **Safety and Reliability**: Ensuring the safety and reliability of the remote-controlled vehicle is paramount. This includes implementing fail-safe mechanisms to prevent accidents or collisions, incorporating feedback sensors for position and orientation tracking, and conducting thorough testing to validate the robustness of the system.

7. **Customization and Expansion**: The design should allow for customization and expansion according to specific requirements or preferences. This may involve integrating additional sensors for environment perception, incorporating autonomous navigation capabilities, or interfacing with external devices for enhanced functionality.

1.4 MECCANUM WHEEL DEVELOPMENT HISTORY:

1. **Early Concepts and Development (1960s-1970s)**:
   The concept of omnidirectional wheels, including Meccanum wheels, was developed in the 1960s and 1970s by various engineers and researchers exploring novel wheel configurations for improved maneuverability.
2. **Invention by Bengt Ilon (1970s)**:
   Bengt Ilon, a Swedish engineer, invented the Meccanum wheel system in the 1970s. Ilon developed the concept to create a versatile and maneuverable wheel configuration for industrial and robotic applications.
3. **Patent and Commercialization (1970s-1980s)**:
   Bengt Ilon patented the Meccanum wheel design in the 1970s. The patent, titled "Wheel arrangement for a vehicle," described the unique arrangement of rollers mounted on wheels to achieve omnidirectional movement. Subsequently, the Meccanum wheel design was commercialized for various applications.
4. **Industrial and Robotics Applications (1980s-1990s)**:
   Throughout the 1980s and 1990s, Meccanum wheels gained popularity in industrial and robotics applications, particularly in material handling, warehouse automation, and mobile robotics. The wheels' ability to move in any direction without changing orientation made them valuable for tasks requiring precise maneuverability.
5. **Advancements in Control Systems (2000s-Present)**:
   In the 2000s and beyond, advancements in control systems and robotics technology further expanded the capabilities and applications of Meccanum-wheeled vehicles. Sophisticated control algorithms, including PID control and motion planning techniques, enabled precise and agile motion control in various environments.

2.MECHANISM OF MECCANUM WHEELS

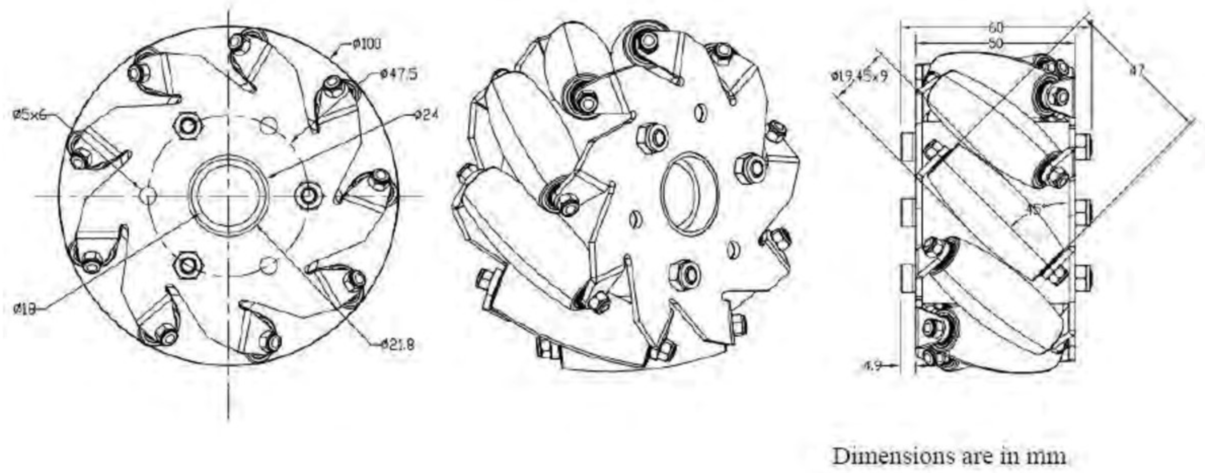

Fig 2.1.1

The mechanism of Meccanum wheels involves a set of specially designed wheels equipped with rollers mounted at an angle to the wheel's axis. These rollers are typically oriented at a 45-degree angle to the wheel's rotation axis and are free to rotate independently.

The key components and mechanisms of Meccanum wheels include:

1. **Wheel Structure**: Meccanum wheels consist of a central hub connected to a series of rollers arranged around the circumference of the wheel. The hub contains a motor or a drive mechanism that rotates the wheel.
2. **Rollers**: The rollers are mounted on the wheel hub at an angle, typically 45 degrees, to the wheel's axis of rotation. Each roller is free to rotate independently around its own axis. The orientation of the rollers enables omnidirectional movement when they are driven at different speeds and directions.
3. **Roller Arrangement**: The rollers are arranged in a specific pattern around the circumference of the wheel. Typically, four rollers are used, positioned symmetrically at 45-degree intervals. This arrangement ensures stability and smooth movement in all directions.
4. **Individual Roller Control**: The key feature of Meccanum wheels is the ability to control each roller independently. By adjusting the speed and direction of rotation of each roller, the vehicle can achieve omnidirectional movement. For example, to move forward, the rollers on each side of the vehicle rotate in the same direction, while to

move sideways, the rollers on one side rotate in the opposite direction to those on the other side.

5. **Drive System**: Meccanum wheels can be powered by various drive systems, including electric motors, servo motors, or hydraulic systems. These drive systems control the rotation of the wheel hub and, consequently, the movement of the vehicle.
6. **Control System**: The control system of a vehicle equipped with Meccanum wheels is responsible for coordinating the movement of the individual rollers to achieve the desired motion. This control system can be implemented using a variety of technologies, including microcontrollers, sensors, and feedback loops.

2.1 TYPES OF MECCANUM WHEELS:

Meccanum wheels come in various types, each offering unique features and advantages suited for different applications. Here are some common types of Meccanum wheels:

2.1.1 STANDARD MECCANUM WHEELS:

Fig 2.1.1.1

These are the traditional Meccanum wheels consisting of a central hub with multiple rollers mounted at an angle around the circumference. Standard Meccanum wheels are suitable for general-purpose applications and are widely used in robotics, material handling, and mobile platforms.

2.1.2 HEAVY-DUTY MECCANUM WHEELS:

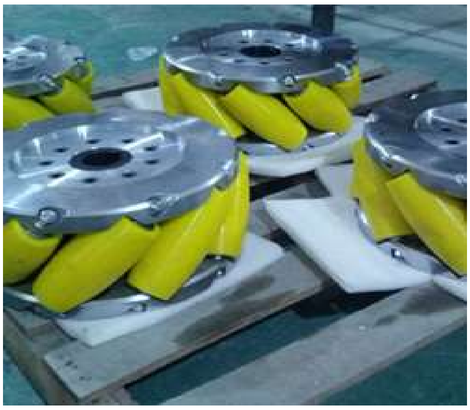

Fig 2.1.2.1

Fig 2.1.2.2

Designed for applications requiring higher load capacity and durability, heavy-duty Meccanum wheels feature reinforced construction and larger rollers. These wheels can withstand heavier loads and rougher terrain, making them suitable for industrial carts, AGVs (Automated Guided Vehicles), and specialized vehicles used in harsh environments.

2.1.3 MINIATURE MECCANUM WHEELS:

Fig 2.1.3.1

Miniature Meccanum wheels are smaller in size and lightweight, making them ideal for compact robotic platforms and applications where space is limited. They offer similar omnidirectional movement capabilities as standard Meccanum wheels but are designed to operate in confined spaces or on smaller-scale projects.

2.1.4 CUSTOMIZED MECCANUM WHEELS:

Fig 2.1.4.1

Manufacturers and engineers may offer customized Meccanum wheels tailored to specific requirements or preferences. These customized wheels may feature variations in roller size, angle, or material composition to optimize performance for particular applications. For example, specialized Meccanum wheels may be designed for high-speed operation, precise maneuverability, or compatibility with specific flooring surfaces.

2.1.5 MODULAR MECCANUM WHEELS:

Fig 2.1.5.1

Modular Meccanum wheels consist of interchangeable components that allow users to customize the wheel configuration according to their needs. These wheels may feature removable or adjustable rollers, hubs, or mounting brackets, enabling easy adaptation to different environments or vehicle designs. Modular Meccanum wheels offer flexibility and versatility, making them suitable for prototyping, experimentation, and rapid iteration in research and development projects.

2.1.6 POWERED MECCANUM WHEELS:

Fig 2.1.6.1

While all Meccanum wheels require external motors or actuators to drive their rollers, powered Meccanum wheels integrate motors directly into the wheel assembly. This design simplifies installation and reduces the overall footprint of the vehicle, making it more compact and efficient. Powered Meccanum wheels are commonly used in compact robots, drones, and other small-scale applications where space is limited.

2.2 ADVANTAGES OF MECCANUM WHEEL:

1. **Omnidirectional Movement**: Meccanum wheels enable vehicles to move in any direction without changing their orientation. This versatility allows for precise maneuvering in tight spaces and complex environments.
2. **Smooth and Agile Motion**: Vehicles equipped with Meccanum wheels can execute smooth, continuous motions, including rotating in place and moving diagonally. This agility is beneficial in applications requiring precise control and navigation.
3. **Reduced Turning Radius**: Meccanum wheels allow vehicles to turn on the spot, eliminating the need for wide turning arcs. This capability is advantageous in confined spaces where maneuverability is limited.
4. **Simple Mechanical Design**: Meccanum wheels have a relatively simple mechanical design compared to other omnidirectional drive systems like holonomic or

omnidirectional wheels. This simplicity can lead to easier maintenance and lower manufacturing costs.

5. **Improved Traction**: Meccanum wheels provide better traction on smooth surfaces compared to traditional omnidirectional wheels. The angled rollers create more contact points with the ground, enhancing stability and grip.
6. **Versatility**: Meccanum wheels are suitable for a wide range of applications, including robotics, material handling, AGVs (Automated Guided Vehicles), and entertainment systems. Their omnidirectional movement capabilities make them adaptable to various tasks and environments.

2.3 DISADVANTAGES OF MECCANUM WHEEL:

1. **Reduced Efficiency on Rough Terrain**: Meccanum wheels may experience reduced efficiency and performance on uneven or rough terrain. The rollers can get caught on obstacles or lose traction, limiting the vehicle's ability to traverse challenging surfaces.
2. **Complex Control Systems**: Achieving precise control of Meccanum-wheeled vehicles requires sophisticated control algorithms and feedback mechanisms. The independent control of each wheel adds complexity to the control system, increasing development and implementation challenges.
3. **Limited Load Capacity**: Meccanum wheels typically have lower load capacities compared to other wheel configurations. Vehicles carrying heavy loads may experience stability issues or reduced maneuverability when equipped with Meccanum wheels.
4. **Higher Energy Consumption**: The omnidirectional movement capabilities of Meccanum wheels often come at the cost of higher energy consumption. The complex motion patterns and increased friction can lead to greater power requirements compared to traditional wheeled systems.
5. **Specialized Training Required**: Operating Meccanum-wheeled vehicles effectively may require specialized training and expertise. Users must understand the principles of omnidirectional motion and be proficient in controlling vehicles equipped with Meccanum wheels.
6. **Maintenance Challenges**: While Meccanum wheels have a simple mechanical design, maintaining optimal performance may require regular inspection and adjustment of the rollers and drive mechanisms. Failure to maintain these components can lead to reduced efficiency and increased wear over time.

3.DESIGN CRITERIA

The designing and implementing a Meccanum wheel remote-controlled vehicle using ESP32 and peer-to-peer WiFi communication, several design criteria need to be considered to ensure the project's success.

1. **Mechanical Design**:
   - **Frame Structure**: Design a sturdy and lightweight frame that can accommodate the Meccanum wheels and other components while providing sufficient strength and durability.
   - **Wheel Mounting**: Ensure proper mounting of the Meccanum wheels to the frame to allow for omnidirectional movement.
   - **Chassis Layout**: Organize the layout of components such as motors, batteries, and electronics to optimize weight distribution, balance, and accessibility.
2. **Electronics and Control System:**
   - **ESP32 Integration**: Select and integrate ESP32 microcontrollers for wireless communication and control. Ensure compatibility with Meccanum wheel control algorithms and libraries.
   - **Motor Drivers**: Choose suitable motor drivers to control the Meccanum wheels' motors. Ensure compatibility with ESP32 and adequate power handling capabilities.
   - **Sensors**: Incorporate sensors such as encoders, gyroscopes, and accelerometers for feedback and control purposes. These sensors can provide data for motion sensing, navigation, and stability control.
   - **Power Management**: Implement an efficient power management system to regulate voltage levels, charge batteries, and maximize runtime.
3. **Wireless Communication**:
   - **Peer-to-Peer WiFi**: Utilize peer-to-peer WiFi communication between multiple ESP32 devices for remote control and data transmission. Ensure reliability, low latency, and adequate range for the intended application.
   - **Protocol Selection**: Choose appropriate communication protocols and standards (e.g., TCP/IP, UDP) for data exchange between the remote controller and the vehicle.

- **Security**: Implement security measures to prevent unauthorized access and ensure the integrity and confidentiality of communication.

4. **Control and Navigation**:
   - **Motion Control Algorithms**: Develop or utilize existing control algorithms for Meccanum wheel motion control, including forward, backward, sideways movement, and rotation.
   - **User Interface**: Design an intuitive user interface for the remote controller, possibly using a smartphone app or a custom-built controller with joystick or accelerometer input.
   - **Navigation Assistance**: Implement features such as obstacle detection, collision avoidance, and path planning to assist with navigation and maneuvering in complex environments.
5. **Testing and Iteration**:
   - **Prototyping**: Build and test prototypes of the vehicle to evaluate performance, identify potential issues, and refine the design.
   - **Simulation**: Use simulation software or tools to simulate vehicle behavior, test control algorithms, and optimize performance before physical implementation.
   - **Feedback and Improvement**: Gather feedback from testing and user experience to iteratively improve the design, functionality, and usability of the vehicle.

3.1 DESIGN CRITERIA FOR MECCANUM WHEEL:

1. **Roller Configuration**:
   - The orientation and arrangement of rollers around the wheel circumference are crucial. Typically, Meccanum wheels have four rollers mounted at 45-degree angles to the wheel's axis..
2. **Wheel Construction**:
   - The wheel's construction should be robust enough to withstand various loads and environments. Materials like rubber, polyurethane, or a combination of materials are often used for the wheel's outer surface to provide traction and durability.The hub or core of the wheel should be sturdy and lightweight, often made from aluminum, steel, or composite materials, to ensure proper mounting and structural integrity.
3. **Mounting System**:

- Meccanum wheels need to be securely mounted to the vehicle's chassis or frame to maintain stability and alignment.
- Mounting brackets or hubs should be designed to accommodate the specific dimensions and mounting pattern of the Meccanum wheels.

4. **Size and Dimensions**:
    - The size of Meccanum wheels should be selected based on the vehicle's size, weight, and intended use.
    - The diameter and width of the wheels affect the vehicle's ground clearance, stability, and load-bearing capacity.
5. **Load Capacity**:
    - Meccanum wheels should be rated for the anticipated load of the vehicle, including the weight of the chassis, payload, and any additional equipment.
    - Consideration should be given to factors such as dynamic loads, shock loading, and continuous operation to ensure the wheels can handle the required load without premature wear or failure.
6. **Friction and Traction**:
    - The surface material and tread pattern of the rollers should be optimized to provide adequate traction on various surfaces, including smooth floors, carpets, and outdoor terrain.
    - The coefficient of friction between the rollers and the surface affects the vehicle's maneuverability and efficiency.
7. **Alignment and Calibration**:
    - Proper alignment of the rollers is crucial for achieving smooth and precise omnidirectional movement.
    - Calibration procedures may be necessary to ensure uniform performance across all rollers and to compensate for any manufacturing tolerances or variations.
8. **Integration with Control System**:
    - Meccanum wheels should be compatible with the vehicle's control system, including motor drivers, sensors, and microcontrollers.
    - Consideration should be given to communication protocols, control algorithms, and feedback mechanisms to optimize performance and responsiveness.

3.2 BLOCK DIAGRAM INTERFACE

REMOTE CONTROLLER:

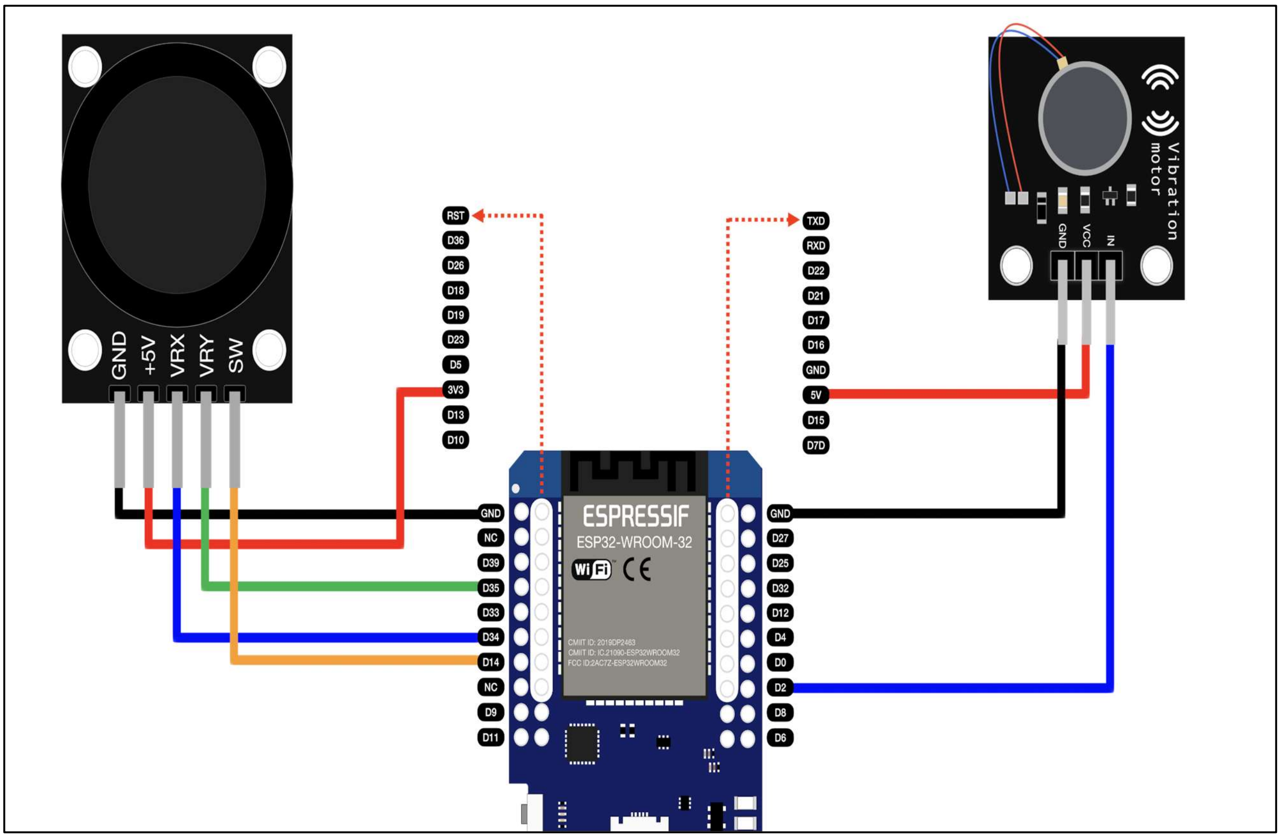

Fig 3.2.1

Here we used ESP32, Power Adaptor, Touch Sensor,Battery, LED Bulb, Jump Wires, PCB board prototype,Joystick by using wifi peer to peer communication in it.

- LED Bulb
- ESP32
- Touch Sensor
- Power Adaptor
- Jump Wires
- Joystick
- Battery
- Breadboard

MECCANUM WHEEL VEHICLE:

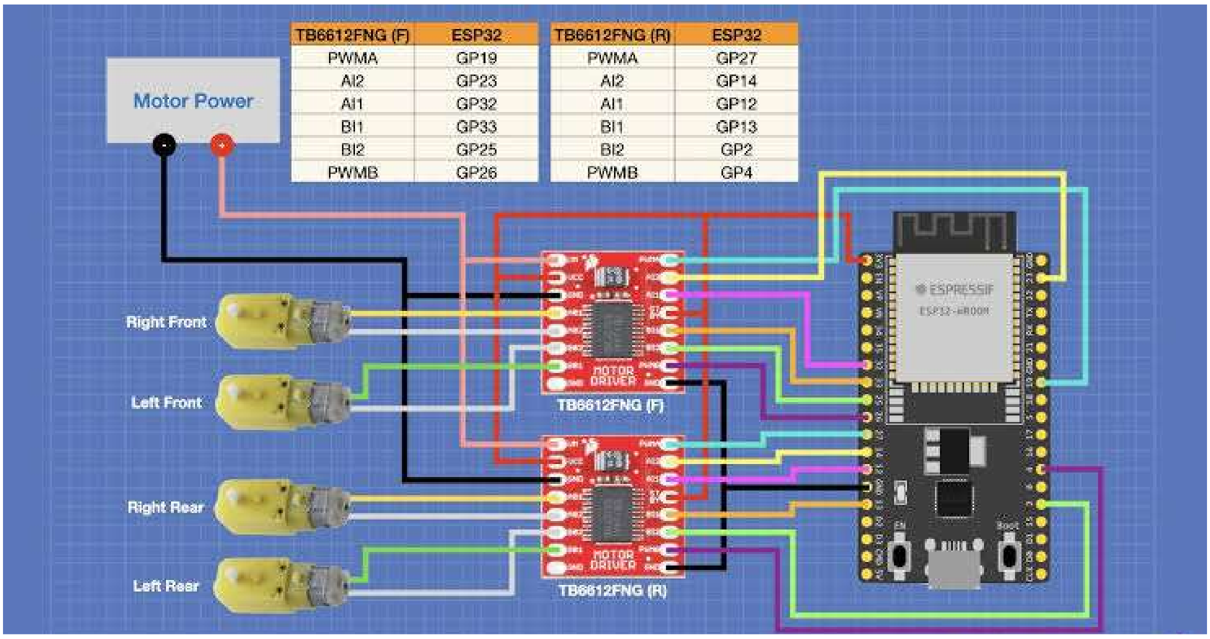

Fig 3.2.2

Here we used ESP32, FNG Motor, Meccanum Wheel, Buck Converter, Gear Motor, Switch, Battery and Jump wires in which dump the code by using software, and connect through wires and design in it.

- 2 FNG Motors
- ESP32
- 4 Gear Motors
- Buck Converter
- Switch
- Wires
- Power Supply
- Battery
- Breadboard

4. CONCEPT OF MECHANISM IN MECCANUM WHEELS AND ITS KINEMATICS

4.1 MECHANISM IN MECCANUM WHEELS:

The mechanism of Meccanum wheels involves understanding how the motion of individual rollers contributes to the overall movement of a vehicle equipped with Meccanum wheels. To illustrate this mechanism, let's consider a simplified scenario with two-dimensional motion (forward and sideways) and use basic trigonometric calculations.

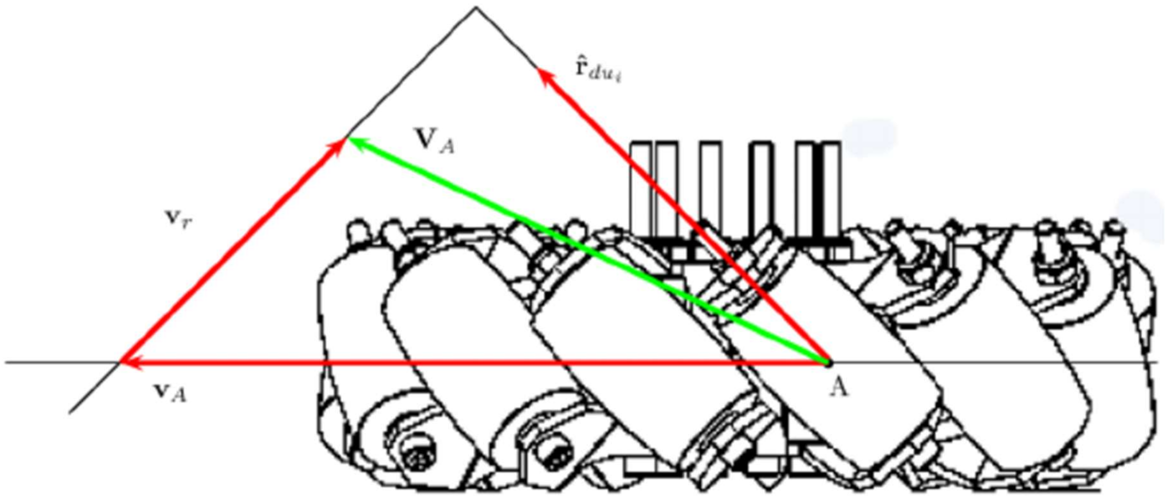

Fig 4.1.1

Components of Meccanum Wheel Mechanism:

1. **Rollers**: Meccanum wheels consist of several rollers mounted around the circumference of the wheel. Each roller is positioned at a 45-degree angle to the wheel's axis of rotation.
2. **Roller Velocity**: The velocity of each roller can be decomposed into two components: tangential velocity (Vt) along the direction of the roller's rotation and lateral velocity (Vl) perpendicular to the roller's rotation.

Calculations:

1. **Forward Velocity (Vx)**: The forward velocity of the vehicle is the sum of the tangential velocities of all four rollers.

   $V_x = V_{t1} + V_{t2} + V_{t3} + V_{t4}$

2. **Sideways Velocity (Vy)**: The sideways velocity of the vehicle is the sum of the lateral velocities of all four rollers.

   $V_y = V_{l1} + V_{l2} + V_{l3} + V_{l4}$

3. **Angular Velocity (W)**: The angular velocity of the vehicle is determined by the difference in tangential velocities between diagonally opposed rollers.

   $W = \frac{1}{2R}(V_{t2} + V_{t3} - V_{t1} - V_{t4})$

   Where:

- V_x = Forward velocity of the vehicle
- V_y = Sideways velocity of the vehicle
- W = Angular velocity of the vehicle
- V_{ti} = Tangential velocity of roller i
- V_{li} = Lateral velocity of roller i
- R = Distance from the center of the vehicle to the contact point of the Meccanum wheel (wheelbase)

Sample Calculations:

Let's assume:

- $V_{t1} = V_{t4} = 1 m/s$ (forward tangential velocities of rollers 1 and 4)
- $V_{t2} = V_{t3} = 0 m/s$ (forward tangential velocities of rollers 2 and 3)
- $V_{l1} = V_{l2} = 0 m/s$ (sideways lateral velocities of rollers 1 and 2)
- $V_{l3} = V_{l4} = 1 m/s$ (sideways lateral velocities of rollers 3 and 4)
- $R = 0.5m$ (distance from the center of the vehicle to the contact point of the Meccanum wheel)

1. Calculate Forward Velocity (V_x): $V_x = 1 + 0 + 0 + 1 = 2 m/s$

2. Calculate Sideways Velocity (V_y): $V_y=0+0+1+1=2$m/s
3. Calculate Angular Velocity (W): $W=2\times0.51(0+0-1-1)=-2$rad/s

These calculations demonstrate how the velocities of individual rollers contribute to the overall motion of a vehicle equipped with Meccanum wheels. The forward and sideways velocities determine the linear motion of the vehicle, while the angular velocity controls its rotation.

4.2 <u>KINEMATICS OF MECCANUM WHEELS:</u>

Meccanum wheel kinematics describe the relationship between the velocities of individual rollers and the resulting motion of a vehicle equipped with Meccanum wheels. The kinematics are based on vectorial analysis, considering the direction and magnitude of each roller's velocity. Here's an overview of Meccanum wheel kinematics.

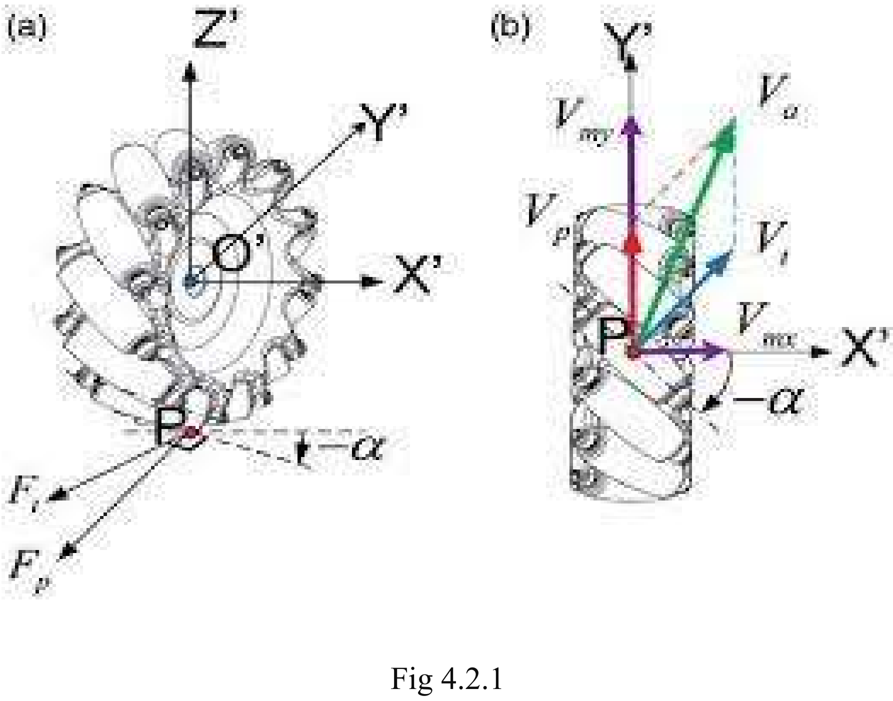

Fig 4.2.1

Basic Concepts:

1. **Roller Configuration**: Meccanum wheels typically consist of four rollers arranged symmetrically around the circumference of the wheel. Each roller is mounted at a 45-degree angle to the wheel's axis of rotation.

2. **Roller Velocity**: The velocity of each roller can be decomposed into two components: tangential velocity (Vt) along the direction of the roller's rotation and lateral velocity (Vl) perpendicular to the roller's rotation. By controlling the speed and direction of each roller's rotation, the vehicle can achieve omnidirectional movement.
3. **Vehicle Motion**: The vehicle's overall motion is determined by the combined effect of the velocities of all four rollers. By summing the tangential and lateral velocities of each roller, the resultant velocity vector represents the vehicle's motion in the horizontal plane.

Formulas:

1. **Forward Velocity (Vx):** $V_x = (v_x \cdot \cos(\theta)) + (v_y \cdot \sin(\theta))$
2. **Sideways Velocity (Vy)**: $V_y = (v_y \cdot \cos(\theta)) - (v_x \cdot \sin(\theta))$
3. **Angular Velocity (W)**: $W = {}_{Lr} \cdot (v_x - v_y)$

Where:

- V_x = Forward velocity of the vehicle
- V_y = Sideways velocity of the vehicle
- W = Angular velocity of the vehicle
- v_x = Desired forward velocity of the vehicle
- v_y = Desired sideways velocity of the vehicle
- θ = Angle of the desired motion (in radians)
- r = Radius of the Meccanum wheel
- L = Distance from the center of the vehicle to the contact point of the Meccanum wheel (wheelbase)

Control and Implementation:

1. **Inverse Kinematics**: Inverse kinematics involves determining the velocities of individual rollers required to achieve a desired motion of the vehicle. This calculation is based on the desired forward and sideways velocities of the vehicle.

2. **Feedback Control**: Feedback control algorithms adjust the velocities of individual rollers based on sensor feedback to maintain desired motion and stability. Proportional-Integral-Derivative (PID) control and state estimation techniques are commonly used for Meccanum wheel control.
3. **Trajectory Planning**: Trajectory planning algorithms generate smooth paths for the vehicle to follow while considering kinematic constraints and dynamic limitations. This ensures efficient and stable motion in various environments.
4. **Simulation and Testing**: Simulation tools such as ROS (Robot Operating System), Gazebo, or MATLAB/Simulink are used to model Meccanum-wheeled vehicles, simulate their behavior, and validate control algorithms before implementation on physical platforms.

By understanding Meccanum wheel kinematics and implementing appropriate control strategies, engineers can design vehicles capable of agile and precise omnidirectional movement for various applications, including robotics, material handling, and automation.

5. SAMPLE CALCULATION

SAMPLE 1:

Let's assume we want to calculate the velocities required for a vehicle to move forward

($vx=0.5m/s$), sideways ($vy=0.3m/s$), and rotate ($\theta=45\circ$).

Given:

- $r=0.1m$ (radius of Meccanum wheel)
- $L=0.2m$ (distance from the center of the vehicle to the contact point of the Meccanum wheel)

1. Calculate Forward Velocity (Vx):
2. $Vx=(0.5\cdot\cos(45\circ))+(0.3\cdot\sin(45\circ))$

   $Vx=(0.5\cdot 0.7071)+(0.3\cdot 0.7071)$

   $Vx=0.5656m/s$
3. Calculate Sideways Velocity (Vy):

   $Vy=(0.3\cdot\cos(45\circ))-(0.5\cdot\sin(45\circ))$

   $Vy=(0.3\cdot 0.7071)-(0.5\cdot 0.7071)$

   $Vy=0.2121-0.3535$

   $Vy=-0.1414m/s$
4. Calculate Angular Velocity (W):

   $W=0.20.1\cdot(0.5-0.3)$

   $W=0.5\cdot 0.2$

   $W=0.1rad/s$

SAMPLE 2:

Let's say we want the vehicle to move forward at 0.6 m/s$0.6m/s$ and sideways at 0.4 m/s$0.4m/s$. We'll also assume the vehicle's wheelbase (LL) is 0.3 m$0.3m$ and the radius of the Meccanum wheels (rr) is 0.05 m$0.05m$.

Calculations:

1. **Forward Velocity (Vx)**: Using the formula: $Vx=(vx \cdot \cos(\theta))+(vy \cdot \sin(\theta))$
   Given:
   - $vx=0.6m/s$ (desired forward velocity)
   - $vy=0.4m/s$ (desired sideways velocity)
   - $\theta=0\circ$ (direction of motion)

   $Vx=0.6+0$ $Vx=0.6+0$ $Vx=0.6$ m/s $Vx=0.6m/s$

2. **Sideways Velocity (Vy)**: Using the formula: $Vy=(vy \cdot \cos(\theta))-(vx \cdot \sin(\theta))$
   Given:
   - $vx=0.6m/s$ (desired forward velocity)
   - $vy=0.4m/s$ (desired sideways velocity)
   - $\theta=0\circ$ (direction of motion)

   $Vy=(0.4 \cdot \cos(0\circ))-(0.6 \cdot \sin(0\circ))$
   $Vy=0.4-0$
   $Vy=0.4m/s$

3. **Angular Velocity (W)**: Using the formula: $W=Lr \cdot (vx-vy)$
   Given:
   - $r=0.05m$ (radius of Meccanum wheels)
   - $L=0.3m$ (wheelbase)
   - $vx=0.6m/s$ (desired forward velocity)
   - $vy=0.4m/s$ (desired sideways velocity)

   $W=0.30.05 \cdot (0.6-0.4)$
   $W=0.30.05 \cdot 0.2$
   $W=0.0333rad/s$

Summary:

- Forward Velocity (Vx): 0.6 m/s
- Sideways Velocity (Vy): 0.4 m/s
- Angular Velocity (W): 0.0333 rad/s

These calculations provide the velocities needed for the vehicle to achieve the desired motion, taking into account the characteristics of Meccanum wheels and the vehicle's geometry. These velocities can then be used to control the rotation of the rollers and achieve the desired movement.

6. HARDWARE COMPONETS USED

6.1 <u>ESP32:</u>

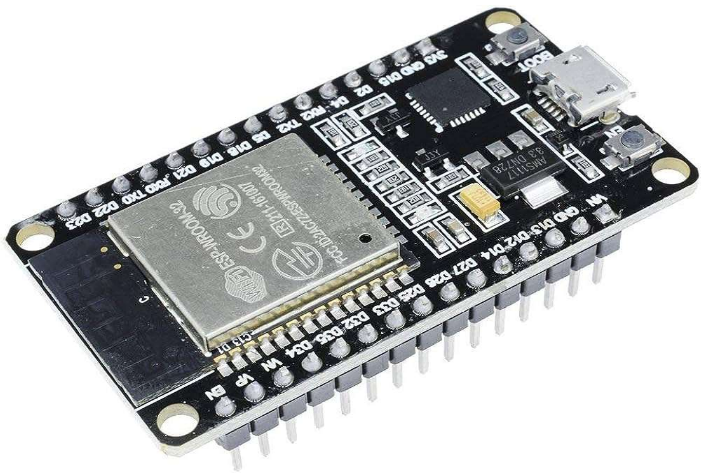

Fig 6.1.1

The ESP32 is a versatile system-on-chip (SoC) microcontroller developed by Espressif Systems, offering a wide range of features and configurations suitable for various IoT (Internet of Things) applications. Below are the specifications

1. **Microcontroller:**
   - Dual-core Tensilica Xtensa LX6 microprocessor
   - Clock frequency: up to 240 MHz
   - Architecture: 32-bit RISC-V Vector Extensions (RV32-V)
2. **Memory:**
   - Internal SRAM: 520 KB
   - Internal ROM: 448 KB
   - External flash memory support: up to 16 MB
3. **Wireless Connectivity:**
   - **Wi-Fi :** IEEE 802.11 b/g/n (2.4 GHz)
   - **Bluetooth:** Bluetooth Low Energy (BLE) 4.2 and 5.0
4. **Peripheral Interfaces:**
   - SPI (Serial Peripheral Interface)
   - I2C (Inter-Integrated Circuit)
   - UART (Universal Asynchronous Receiver-Transmitter)
   - I2S (Inter-IC Sound)

- PWM (Pulse Width Modulation)
- ADC (Analog-to-Digital Converter)
- DAC (Digital-to-Analog Converter)
- Touch sensor inputs
- GPIO (General Purpose Input/Output) pins

5. **Security Features:**
    - Secure boot
    - Flash encryption
    - Hardware-accelerated encryption (AES, SHA-2, RSA)
    - Secure storage
6. **Operating Voltage:** 2.2 V to 3.6 V
7. **Operating Temperature:** -40°C to 125°C
8. **Power Consumption:** Low power consumption design for battery-powered applications
9. **Integrated Hardware Accelerators:** Floating-point unit (FPU)
10. **Development Support:**
    - Arduino IDE
    - ESP-IDF (Espressif IoT Development Framework)
    - PlatformIO
    - MicroPython

6.2 BUCK CONVERTOR:

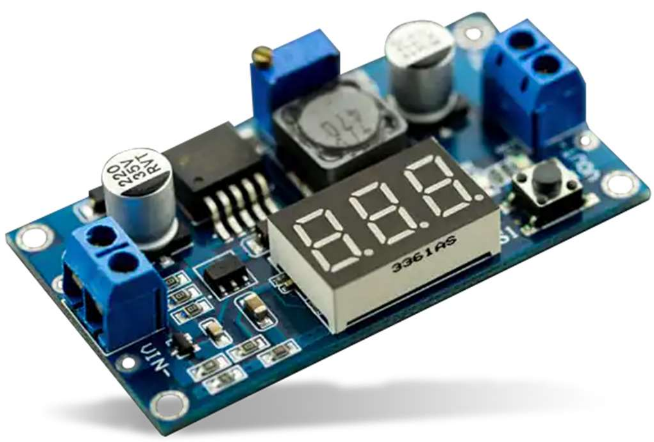

Fig 6.2.1

A buck converter, also known as a step-down converter, is a type of DC-DC converter that transforms a higher voltage DC input to a lower voltage DC output. It is widely used in electronic devices and power supply systems to efficiently regulate voltage levels.

Specifications of DC-DC Buck Converter Step Down Module LM2596 Power Supply :

- Conversion efficiency: 92%(highest)
- Switching frequency: 150KHz
- Output ripple: 30mA9maximum)
- Load Regulation: ± 0.5%
- Voltage Regulation: ± 0.5%
- Dynamic Response speed: 5% 200uS
- Input voltage:4.75-35V
- Output voltage:1.25-26V(Adjustable)
- Output current: Rated current is 2A,maximum 3A(Additional heat sink is required)
- Conversion Efficiency: Up to 92% (output voltage higher, the higher the efficiency)
- Switching Frequency: 150KHz
- Rectifier: Non-Synchronous Rectification
- Module Properties: Non-isolated step-down module (buck)
- Short Circuit Protection: Current limiting, since the recovery
- Operating Temperature: Industrial grade (-40 to +85) (output power 10W or less)

6.3 <u>FNG MOTOR DRIVE:</u>

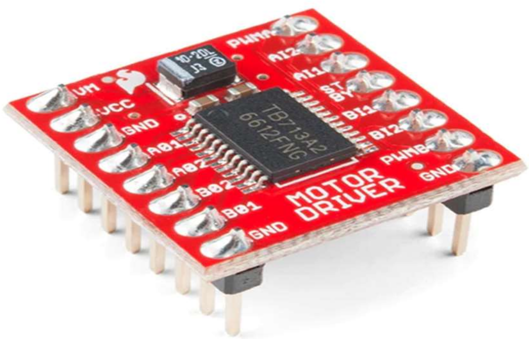

Fig 6.3.1

The TB6612FNG is a great dual motor driver that is perfect for interfacing two small DC motors such as our micro metal gearmotors to a microcontroller, and it can also be used to control a single bipolar stepper motor. The MOSFET-based H-bridges are much more efficient than the BJT-based H-bridges used in older drivers such as the L298N and Sanyo's LB1836M, which allows more current to be delivered to the motors and less to be drawn from the logic supply (the LB1836 still has the TB6612 beat for really low-voltage applications). Our little breakout board gives you direct access to all of the features of the TB6612FNG and adds power supply capacitors and reverse battery protection on the motor supply (note: there is no reverse protection on the Vcc connection).

- Dual-H-bridge motor driver: can drive two DC motors or one bipolar stepper motor
- Recommended motor voltage (VMOT): 4.5 V to 13.5 V (can operate down to 2.5 V with derated performance)
- Logic voltage (VCC): 2.7 V to 5.5 V
- Output current maximum: 3 A per channel
- Output current continuous: 1 A per channel (can be paralleled to deliver 2 A continuous)
- Maximum PWM frequency: 100 kHz
- Built-in thermal shutdown circuit
- Filtering capacitors on both supply lines
- Reverse-power protection on the motor supply

6.4 MECCANUM WHEEL:

Fig 6.4.1

The mecanum wheel is a form of tireless wheel, with a series of rubberized external rollers obliquely attached to the whole circumference of its rim. These rollers typically each have an axis of rotation at 45° to the wheel plane and at 45° to the axle line. Each Mecanum wheel is an independent non-steering drive wheel with its own powertrain, and when spinning generates a propelling force perpendicular to the roller axle, which can be vectored into a longitudinal and a transverse component in relation to the vehicle.

SPECIFICATION

- Diameter: 48mm/1.89”
- Thickness: 24.5mm/0.96”
- Hex Hole: 7*7*7mm
- Number of Rollers: 9
- Angle: 45°
- Color: yellow
- Material: plastic + silicone rubber

6.5 METAL GEAR MOTORS:

A geared motor is a component whose mechanism adjusts the speed of the motor, leading them to operate at a certain speed. geared motor have the ability to deliver high torque at low speeds, as the gearhead functions as a torque multiplier and can allow small motors to generate higher speeds.

A geared motor can also be defined as a gear reducer because essentially, it is a combination of a speed reducer with a motor typically functioning as a gearbox, to reduce speed making more torque available.

A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gearbox to a motor reduces the speed while increasing the torque output. The most important parameters in regard to gear motors are speed (rpm), torque (lb-in) and efficiency (%). In order to select the most suitable assembly for your application, you must first compute the load, speed and torque requirements for your application.

Gear ratio:	986.41:1
No-load speed @ 6V:	31 rpm[3]
No-load current @ 6V:	0.10 A[4]
Stall current @ 6V:	1.6 A[5]
Stall torque @ 6V:	12 kg·cm[5]
Extended motor shaft?:	N
Motor type:	1.6A stall @ 6V (HP 6V)

Fig 6.5.1

6.6 TOUCH SENSOR:

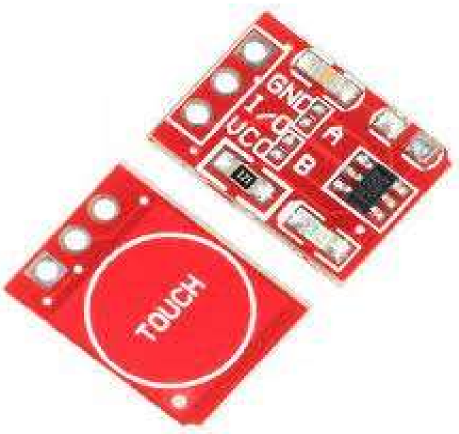

Fig 6.6.1

The TTP223 1-Channel Capacitive Touch Sensor Module Red Color is based on a touch-sensing IC (TTP223) capacitive touch switch module. A capacitive touch sensor module based on the dedicated TTP223 touch sensor IC. The module provides a single integrated touch sensing area of 11 x 10.5mm with a sensor range of ~5mm.

An onboard LED will give a visual indication of when the sensor is triggered. When triggered the output of the module will switch from its idle low state to high (default operation). Solder jumpers allow for reconfiguring its mode of operation to be either active low or toggle output.

TTP223 Capacitive Touch Sensor Module. The TTP223 is a touchpad detector IC replicating a single tactile button.

The TTP223 is a touchpad detector IC replicating a single tactile button. This touch detection IC is designed for replacing traditional direct button key with diverse pad size.

6.7 BATTERY:

Fig 6.7.1

Battery is a part of a circuit that provides the electricity. Battery can be said as the source to provide electricity to the circuit. So its main function is to supply electric power in order for electric items to work.

A battery is a device consisting of one or more electro chemical cells with external connection provided to power electric device. A battery have a positive terminal or cathode and a negative terminal or anode. The terminal marked positive is at a higher electrical potential energy then is the terminal marks negative terminal have electron that is flow external circuit and deliver energy here for battery are connected in series and it is use to power supply to BLDC motor, than motor is make forward and backward to wheel.

6.8 BATTERY CHARGER:

Fig 6.8.1

A battery charger or recharge is a device used to put energy into a secondary or rechargeable battery by forcing an electric current through it. The charging protocol depends on the size and tyre of the battery being charged. Some batteries have high tolerance for overcharging and can be charge by connection to constant voltage source or constant voltage source or constant voltage source. Microprocessor controller to adjust the charging current, determine the sate of charge and cut off at the end of charge.

6.9 LED BULB:

Fig 6.9.1

An LED bulb, also known as a light-emitting diode bulb, is a type of light bulb that uses light-emitting diodes (LEDs) as the source of illumination. LED bulbs are highly efficient, long-lasting, and environmentally friendly alternatives to traditional incandescent and compact fluorescent bulbs.

An LED bulb used for indication purposes serves as a visual indicator to convey specific information or status in various devices and systems.

6.10 JUMPER WIRES:

Fig 6.10.1

Connecting wire is a piece of wire used to attach two circuits or components together. The gauge or the size of the wire must be large enough to support the amount of current flow. Wires are used to join parts of a circuit. Electricity flows through wires.Its main function is to provide electrical items the power they need to work, Provided by battery.

Jumper wires are of 3 types. They are:
Male to Male (M2M)
Male to Female (M2F)
Female to Female (F2F)

6.11 <u>PCB BOARD PROTOTYPE CIRCUIT TINNED IN BREADBOARD:</u>

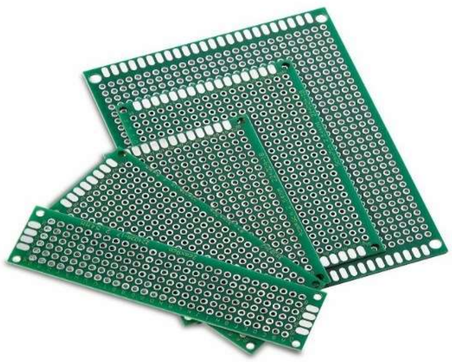

Fig 6.11.1

A breadboard, also known as a protoboard or solderless breadboard, is a device used for prototyping and testing electronic circuits. It provides a platform for temporarily connecting electronic components and wires without the need for soldering, allowing for quick and easy experimentation and modification of circuit designs.

6.12 <u>JOYSTICK:</u>

Fig 6.12.1

A joystick is a type of input device used primarily in electronic systems, such as computers, gaming consoles, remote-controlled vehicles, and industrial control systems, to control the movement of an on-screen cursor, character, or other graphical elements. It typically consists of a handheld lever or stick that can be tilted or moved in various directions, along with one or more buttons or triggers for additional functions.

6.13 METAL FRAME:

Fig 6.13.1

The body structure of this mecanum wheels robot car kit is made of high-quality aluminum plate and has been oxidized and sandblasted. It is not only exquisite in appearance, not easy to be scratched, but also solid and durable, can carry 0-5kg. The firm base design makes this robot chassis be able to run fast and stable. Therefore, this mecanum wheel chassis kit is very suitable for robot experiment and school eduactional science kits.

7. SOFTWARE TOOLS USED

For developing a mecanum wheel remote-controlled vehicle using ESP32 with peer-to-peer WiFi communication, you'll need to consider both the hardware and software aspects. Below are the software requirements for such a project:

1. **ESP32 Development Environment :**
    - Set up the ESP32 development environment using a platform like Arduino IDE or PlatformIO.
2. **WiFi Communication:**
    - Implement WiFi communication using the ESP32's capabilities. For peer-to-peer communication, you may consider setting up the ESP32 devices as both access points and stations simultaneously.
    - Utilize the ESPNow protocol for efficient peer-to-peer communication. ESPNow is a lightweight protocol designed for direct communication between ESP devices.

3. **Motor Control:**
    - Develop motor control algorithms to drive the mecanum wheels based on the control signals received over WiFi. Consider using Pulse Width Modulation (PWM) for controlling motor speeds.
4. **Mecanum Wheel Kinematics:**
    - Implement mecanum wheel kinematics to calculate the individual wheel speeds based on the desired robot motion. This is crucial for achieving the desired movement characteristics of a mecanum wheel robot.
5. **User Interface:**
    - Develop a user interface for remote control. This could be a mobile application, a web-based interface, or a physical remote control device. Ensure it can send control commands to the robot over the WiFi network.
6. **Control Commands:**
    - Define a protocol for sending control commands from the remote control to the robot. This protocol could include commands for forward, backward, sideways, and rotational movements.

7. **Safety Features:**
    - Implement safety features to prevent unintended movements. This might include emergency stop mechanisms or obstacle detection using sensors.
8. **Sensor Integration:**
    - Integrate sensors, such as gyroscope or accelerometer, for feedback control. These sensors can provide information about the robot's orientation, which can be useful for more advanced control.
9. **Error Handling:**
    - Implement error handling mechanisms to deal with communication errors, sensor malfunctions, or other unexpected situations.
10. **Testing and Debugging:**
    - Develop a comprehensive testing strategy to ensure the software functions correctly. Implement debugging features to facilitate troubleshooting and diagnosis.
11. **Documentation:**
    - Document the code thoroughly, including comments, README files, and any necessary documentation on how to use the software and set up the hardware.
12. **Compliance with Standards:**
    - If applicable, ensure that the software complies with any relevant standards or regulations for safety and performance.

7.1 ESP32 DEVELOPMENT ENVIRONMENT:

Set up the ESP32 development environment using a platform like Arduino IDE or PlatformIO ESP32 is a system on a chip that integrates the following features:

- Wi-Fi (2.4 GHz band)
- Bluetooth
- Dual high performance Xtensa® 32-bit LX6 CPU cores
- Ultra Low Power co-processor
- Multiple peripherals

Powered by 40 nm technology, ESP32 provides a robust, highly integrated platform, which helps meet the continuous demands for efficient power usage, compact design, security, high performance, and reliability.

Espress if provides basic hardware and software resources to help application developers realize their ideas using the ESP32 series hardware. The software development framework by Espressif is intended for development of Internet-of-Things (IoT) applications with Wi-Fi, Bluetooth, power management and several other system features.

7.2 <u>WIFI COMMUNICATION:</u>

Implement WiFi communication using the ESP32's capabilities. For peer-to-peer communication, you may consider setting up the ESP32 devices as both access points and stations simultaneously

Utilize the ESPNow protocol for efficient peer-to-peer communication. ESPNow is a lightweight protocol designed for direct communication between ESP devices

The following features are supported:

- 4 virtual Wi-Fi interfaces, which are STA, AP, Sniffer and reserved.
- Station-only mode, AP-only mode, station/AP-coexistence mode
- IEEE 802.11b, IEEE 802.11g, IEEE 802.11n, and APIs to configure the protocol mode
- WPA/WPA2/WPA3/WPA2-Enterprise/WPA3-Enterprise/WAPI/WPS and DPP
- AMSDU, AMPDU, HT40, QoS, and other key features

7.3 <u>MOTOR CONTROL:</u>

Motor control for a mecanum wheel robot involves translating desired movements into specific motor commands. Below is a simplified example of how you might implement motor control in the software for a mecanum wheel robot using ESP32 and Arduino. This example assumes that you have a basic understanding of mecanum wheel kinematics.

- **moveForward**, **moveBackward**, and **stopRobot** functions are used to control the direction of the mecanum wheel robot by setting the appropriate combinations of HIGH and LOW on the motor control pins.
- The **loop** function demonstrates moving forward, backward, and stopping, each for a specific duration.
- WiFi is used to connect the ESP32 to a network. Ensure to replace "YourWiFiSSID" and "YourWiFiPassword" with your actual WiFi credentials.

7.4 MECANUM WHEEL KINEMATICS:

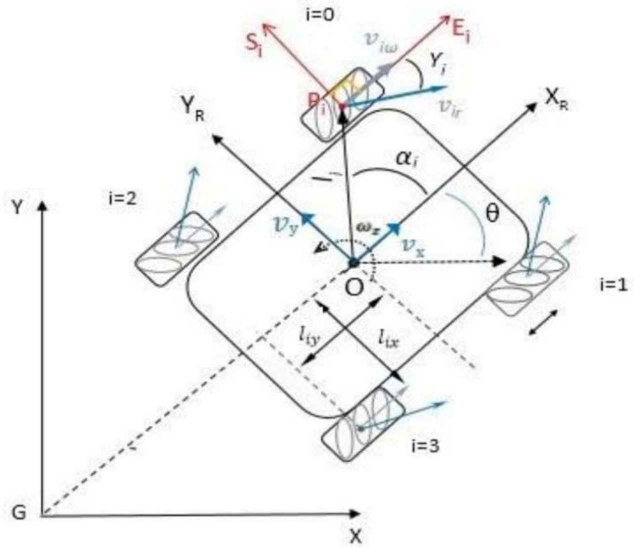

Fig 7.4.1

Mecanum wheel kinematics involves determining the relationship between the motion of each mecanum wheel and the resulting motion of the entire robot. The unique feature of mecanum wheels is their ability to achieve omnidirectional movement, allowing the robot to move forward, backward, sideways, and rotate without changing its orientation.

Each mecanum wheel is equipped with rollers set at a 45-degree angle to the axis of rotation. By independently controlling the speed and direction of rotation of each wheel, the robot can achieve various types of motion.

Here's a detailed explanation of the kinematics for mecanum wheels:

Mecanum Wheel Components:

1. **Roller Angle (α):**
   - The rollers on a mecanum wheel are typically set at a 45-degree angle to the axis of rotation. This angle is denoted as α.
2. **Wheel Axes:**
   - The wheel's axis, or centerline, is the line along which the wheel rotates.
3. **Wheel Velocity (V):**
   - The speed at which each wheel rotates. Each wheel can have a different velocity.

Kinematic Equations:

The kinematic equations for mecanum wheels are derived based on the geometry of the wheels. For each mecanum wheel, the resulting linear velocity (Vx, Vy) and angular velocity (ω) of the robot can be calculated:

1. **Linear Velocities (Vx, Vy):**
   - $Vx=V1\cos(\alpha)+V2\cos(\alpha)+V3\cos(\alpha)+V4\cos(\alpha)$
   - $Vy=V1\sin(\alpha)+V2\sin(\alpha)+V3\sin(\alpha)+V4\sin(\alpha)$

   where V1, V2, V3, and V4 are the linear velocities of each mecanum wheel.
2. **Angular Velocity (ω):**
   - $\omega=d1(V1-V2-V3+V4)$

   where d is the distance between the centers of two opposite mecanum wheels.

Control Inputs:

The control inputs to achieve desired robot motion are the linear velocities along the x-axis (Vx), the y-axis (Vy), and the angular velocity (ω).

Inverse Kinematics:

To achieve a desired motion (Vx, Vy, ω), inverse kinematics is used to calculate the required linear velocities for each mecanum wheel:

1. **Linear Velocity for Each Wheel:**
   - $V1=Vx+Vy+d\cdot\omega$
   - $V2=Vx-Vy-d\cdot\omega$
   - $V3=Vx-Vy+d\cdot\omega$
   - $V4=Vx+Vy-d\cdot\omega$
2. **Forward Velocity (Vx)**: The forward velocity of the vehicle is the sum of the tangential velocities of all four rollers.

   $V_x=V_{t1}+V_{t2}+V_{t3}+V_{t4}$
3. **Sideways Velocity (Vy)**: The sideways velocity of the vehicle is the sum of the lateral velocities of all four rollers.

   $V_y=V_{l1}+V_{l2}+V_{l3}+V_{l4}$

4. **Angular Velocity (W)**: The angular velocity of the vehicle is determined by the difference in tangential velocities between diagonally opposed rollers.

   $W = \frac{1}{2R}(V_{t2} + V_{t3} - V_{t1} - V_{t4})$

Implementation:

In a mecanum wheel robot, the control system adjusts the speeds of individual wheels (V1, V2, V3, V4) based on the desired linear and angular velocities. These velocities are then translated into motor control signals to achieve the desired robot motion.

7.5 MECHANUM WHEELS SENSOR INTEGRATION:

Integrating sensors into a mecanum wheel robot enhances its capabilities by providing feedback about the environment, position, and orientation. Common sensors for mecanum wheel robots include encoders, gyros, accelerometers, and distance sensors. Below is a detailed explanation of integrating these sensors into a mecanum wheel robot:

1. Encoder Integration:

Purpose:

- Encoders measure the rotation of each wheel, allowing the robot to determine its distance traveled and its current position.

Implementation:

- Attach encoders to the shaft of each motor.
- Count the encoder pulses to determine the number of rotations.
- Calculate linear displacement and use this information for odometry.

2. Gyroscope Integration:

Purpose:

- Gyroscopes measure the rate of rotation, providing information about the robot's angular velocity and changes in orientation.

Implementation:

- Attach a gyroscope to the robot's chassis.
- Read gyro data to estimate the robot's heading and angular velocity.
- Use gyro data to improve turning accuracy and maintain a consistent heading.

3. Accelerometer Integration:

Purpose:

- Accelerometers measure the acceleration of the robot, providing information about its movement in a straight line and its orientation relative to gravity.

Implementation:

- Attach an accelerometer to the robot's chassis.
- Read accelerometer data to estimate the robot's linear acceleration.
- Use accelerometer data for tilt compensation and to detect abrupt movements.

4. Distance Sensor Integration:

Purpose:

- Distance sensors (e.g., ultrasonic or infrared sensors) provide information about the robot's proximity to obstacles.

Implementation:

- Mount distance sensors in locations to cover the robot's surroundings.
- Use sensor readings to implement obstacle avoidance algorithms.
- Combine distance sensor data with encoder and gyro data for more robust navigation.

5. Sensor Fusion:

Purpose:

- Combining data from multiple sensors enhances accuracy and reliability.

Implementation:

- Implement sensor fusion algorithms (e.g., Kalman filters) to combine data from encoders, gyros, accelerometers, and distance sensors.
- Improve overall navigation and localization accuracy by integrating information from different sensors.

6. Control Algorithm Adjustments:

Purpose:

- Use sensor data to dynamically adjust control algorithms for better performance.

Implementation:

- Adjust motor speeds based on encoder feedback to maintain straight-line motion.
- Use gyro data to adjust turning rates and ensure accurate rotations.
- Implement closed-loop control using sensor feedback for precise movements.

7. Software Integration:

Purpose:

- Develop software routines to read and process sensor data.

Implementation:

- Write code to read data from each sensor.
- Implement algorithms to process sensor data and extract relevant information.
- Integrate sensor data into the control system for real-time adjustments.

Considerations:

- **Calibration:** Calibrate sensors to ensure accurate and consistent readings.
- **Noise Filtering:** Implement filtering techniques to reduce noise in sensor data.
- **Sensor Placement:** Carefully place sensors to cover the robot's critical areas and minimize interference.

By integrating these sensors and using their data effectively, you can enhance the capabilities of a mecanum wheel robot, enabling it to navigate more intelligently and respond to its environment.

7.6 MECCANUM WHEEL USER INTERFACE:

Creating a user interface (UI) for controlling a mecanum wheel robot involves designing a system that allows the user to interact with the robot, sending commands for different movements and actions. Below are some considerations and steps for creating a basic user interface for a mecanum wheel robot:

- Select a Platform
- Physical Remote Control (Optional)
- Web-Based Interface
- Mobile Application
- Control Elements
- Connection Setup
- Feedback Mechanism
- Responsive Design
- Programming Interface
- Testing
- Security
- Documentation

7.7 CONTROL COMMANDS IN MECHANUM WHEEL:

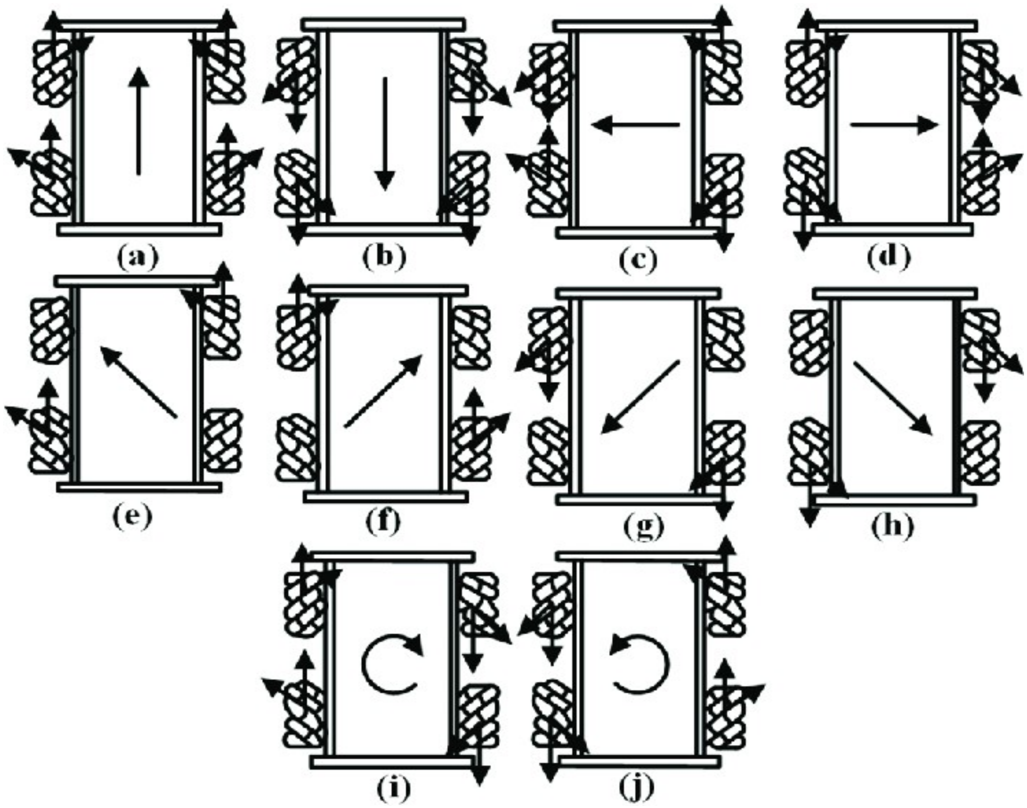

Fig 7.7.1

Controlling a mecanum wheel robot involves sending specific commands to adjust the speeds and directions of individual wheels. The following control commands are commonly used for mecanum wheel robots:

1. Basic Movement Commands:

- **Move Forward:**
  - Set all wheel speeds in the forward direction.
- **Move Backward:**
  - Set all wheel speeds in the reverse direction.
- **Move Left:**
  - Adjust individual wheel speeds to achieve leftward motion.
- **Move Right:**
  - Adjust individual wheel speeds to achieve rightward motion.

2. Rotation Commands:

- **Rotate Clockwise:**
    - Adjust wheel speeds to achieve a clockwise rotation.
- **Rotate Counterclockwise:**
    - Adjust wheel speeds to achieve a counterclockwise rotation.

3. Diagonal Movement Commands:

- **Move Diagonally Forward and Left:**
    - Adjust wheel speeds to achieve diagonal motion in the forward-left direction.
- **Move Diagonally Forward and Right:**
    - Adjust wheel speeds to achieve diagonal motion in the forward-right direction.
- **Move Diagonally Backward and Left:**
    - Adjust wheel speeds to achieve diagonal motion in the backward-left direction.
- **Move Diagonally Backward and Right:**
    - Adjust wheel speeds to achieve diagonal motion in the backward-right direction.

4. Variable Speed Commands:

- **Adjust Speed:**
    - Allow the user to control the robot's speed by adjusting the speed of all wheels simultaneously.

5. Stop Command:

- **Emergency Stop:**
    - Set all wheel speeds to zero for an emergency stop.

6. Custom Sequences (Optional):

- **Execute Custom Sequence:**
    - Allow the user to define and execute custom sequences of movements.

7.8 SAFETY FEATURES FOR MECANUM WHEEL:

Implementing safety features is crucial for ensuring the reliable and secure operation of a mecanum wheel robot. Here are some safety features that you can integrate into your mecanum wheel robot:

1. Emergency Stop (E-stop) Button:

- **Purpose:**
  - Provides a quick and immediate way to stop all robot movements in case of an emergency or unexpected behavior.
- **Implementation:**
  - Include a dedicated physical or software E-stop button that, when pressed, immediately halts all motor movements.

2. Obstacle Detection and Avoidance:

- **Purpose:**
  - Prevents collisions with obstacles in the robot's environment.
- **Implementation:**
  - Integrate sensors (e.g., ultrasonic, infrared) to detect obstacles.
  - Implement algorithms to analyze sensor data and adjust the robot's movements to avoid collisions.

3. Soft Limits:

- **Purpose:**
  - Defines operational boundaries to prevent the robot from moving beyond safe limits.
- **Implementation:**
  - Set limits on the allowable range of motion for each degree of freedom.
  - Program the robot to automatically reduce speed or stop when approaching these limits.

4. Battery Monitoring:

- **Purpose:**
  - Prevents unexpected shutdowns due to low battery voltage.
- **Implementation:**
  - Monitor the battery voltage in real-time.
  - Implement warnings or automatic actions when the battery voltage falls below a safe threshold.

5. Overcurrent Protection:

- **Purpose:**

    - Protects the motors and other electrical components from damage due to excessive current.
- **Implementation:**
    - Incorporate current sensors or fuses to monitor and limit current flow.
    - Automatically reduce motor power or trigger an E-stop if overcurrent is detected.

6. Secure Communication:

- **Purpose:**
    - Ensures that control commands are transmitted securely, preventing unauthorized access.
- **Implementation:**
    - Use encryption for wireless communication channels.
    - Implement authentication mechanisms to verify the legitimacy of control commands.

7. Redundant Safety Systems:

- **Purpose:**
    - Enhances reliability by having backup safety features.
- **Implementation:**
    - Integrate multiple sensors for obstacle detection to increase reliability.
    - Implement redundant E-stop mechanisms for added safety.

8. Fall Detection (Inclination Sensor):

- **Purpose:**
    - Detects and reacts to the robot tipping over or falling.
- **Implementation:**
    - Include an inclination sensor or gyroscope to monitor the robot's tilt.
    - Implement algorithms to respond to sudden tilts by stopping or adjusting the robot's orientation.

9. User Authentication and Access Control:

- **Purpose:**
    - Restricts access to control commands to authorized users.

- **Implementation:**
  - Implement user authentication mechanisms for remote control interfaces.
  - Define different user roles with varying levels of control access.

10. Fail-Safe States:

- **Purpose:**
  - Defines safe states for the robot in case of system failures.
- **Implementation:**
  - Program the robot to transition to predefined safe states when critical failures or malfunctions are detected.

11. Real-time Monitoring and Logging:

- **Purpose:**
  - Provides continuous monitoring of the robot's state and performance.
- **Implementation:**
  - Implement real-time monitoring systems that log sensor data, error messages, and critical events for post-analysis.

12. Regular Maintenance and Inspection:

- **Purpose:**
  - Ensures that the robot remains in good working condition.
- **Implementation:**
  - Establish a maintenance schedule for regular inspection and servicing of mechanical and electronic components.

13. Training and User Guidelines:

- **Purpose:**
  - Ensures that operators are trained to use the robot safely.
- **Implementation:**
  - Provide clear user guidelines and training programs for operators.

Implementing a combination of these safety features will contribute to the overall reliability and safety of your mecanum wheel robot. Always consider the specific requirements of your application and the environment in which the robot will operate.

8. APPLICATIONS AND LIMITATIONS

Fig 8.1

Fig 8.2

Meccanum wheels are versatile and find applications in various fields where omnidirectional movement is desired. Some common applications of mecanum wheels include:

1. **Mobile Robotics:**
    - Mecanum wheels are widely used in mobile robots and unmanned ground vehicles (UGVs) to achieve omnidirectional movement capabilities. They enable robots to move in any direction without changing their orientation, making them suitable for navigation in tight spaces, crowded environments, and complex terrains.
2. **Material Handling Systems:**
    - Mecanum wheels are utilized in material handling systems, such as conveyor belts, automated guided vehicles (AGVs), and warehouse robots, to transport goods and materials with precise control and maneuverability. They allow for efficient movement and sorting of items in logistics and distribution centers.
3. **Industrial Automation:**
    - Mecanum wheels are integrated into industrial automation systems and machinery to facilitate flexible and agile movement of robotic arms, manipulators, and assembly platforms. They enable precise positioning and manipulation of workpieces in manufacturing processes, assembly lines, and production facilities.
4. **Entertainment and Amusement:**
    - Mecanum wheels are used in entertainment and amusement applications, such as robotic toys, interactive exhibits, and amusement park rides, to provide

dynamic and interactive experiences for users. They enable smooth and agile movement of interactive devices and attractions.

5. **Research and Development:**
    - Mecanum wheels are employed in research and development projects, robotics competitions, and academic laboratories for prototyping, testing, and experimenting with omnidirectional motion control algorithms, navigation systems, and robotic platforms.
6. **Agricultural Robotics:**
    - Mecanum wheels are integrated into agricultural robots and automated farming equipment for tasks such as crop monitoring, precision spraying, and autonomous harvesting. They allow robots to navigate through fields and orchards with ease, optimizing efficiency and productivity in agricultural operations.
7. **Medical and Healthcare:**
    - Mecanum wheels are used in medical and healthcare applications, such as hospital logistics robots, patient transport systems, and rehabilitation devices. They enable smooth and precise movement of medical equipment and assistive devices in healthcare facilities and rehabilitation centers.
8. **Educational Robotics:**
    - Mecanum wheels are incorporated into educational robotics kits, STEM (Science, Technology, Engineering, and Mathematics) programs, and robotics workshops to teach students about omnidirectional motion, robotics principles, and programming concepts. They provide hands-on learning experiences and encourage creativity and innovation in robotics education.

Fig 8.3

LIMITATIONS:

While mecanum wheels offer many advantages in terms of omnidirectional movement and maneuverability, they also have several limitations that need to be considered:

1. **Limited Load Capacity:**
   - Mecanum wheels are generally not suitable for heavy-duty applications or transporting heavy loads. Their load capacity is typically lower compared to traditional wheels, making them less suitable for applications that require lifting or moving heavy objects.
2. **Reduced Traction:**
   - Mecanum wheels may have reduced traction compared to traditional wheels, especially on certain surfaces such as loose gravel, sand, or uneven terrain. This limitation can affect the performance of vehicles or robots equipped with mecanum wheels in outdoor environments or challenging terrains.
3. **Complex Control System:**
   - Achieving precise control and coordination of mecanum wheel-equipped vehicles or robots requires a complex control system and software algorithms. Implementing and tuning these control systems can be challenging and may require expertise in robotics and control theory.
4. **Increased Mechanical Complexity:**
   - Mecanum wheels are mechanically more complex compared to traditional wheels, as they require additional rollers or rollers with specific orientations to achieve omnidirectional movement. This complexity can result in higher manufacturing costs and maintenance requirements.
5. **Higher Cost:**
   - Mecanum wheels are typically more expensive than traditional wheels due to their specialized design and manufacturing process. The higher cost of mecanum wheels may limit their adoption in certain applications, especially those with strict budget constraints.
6. **Limited Speed and Efficiency:**
   - Mecanum wheels may have limitations in terms of maximum speed and efficiency compared to traditional wheels, especially at higher speeds. The

design of mecanum wheels and the friction between the rollers and the ground can lead to energy losses and reduced efficiency.

7. **Noise and Vibration:**
   - Mecanum wheels may produce more noise and vibration during operation compared to traditional wheels, especially at higher speeds or on certain surfaces. This noise and vibration can affect the comfort of passengers or operators in vehicles or robots equipped with mecanum wheels.
8. **Maintenance Requirements:**
   - Mecanum wheels may have specific maintenance requirements, such as regular inspection and lubrication of the rollers, to ensure optimal performance and longevity. Failure to maintain mecanum wheels properly can lead to premature wear and reduced reliability

.

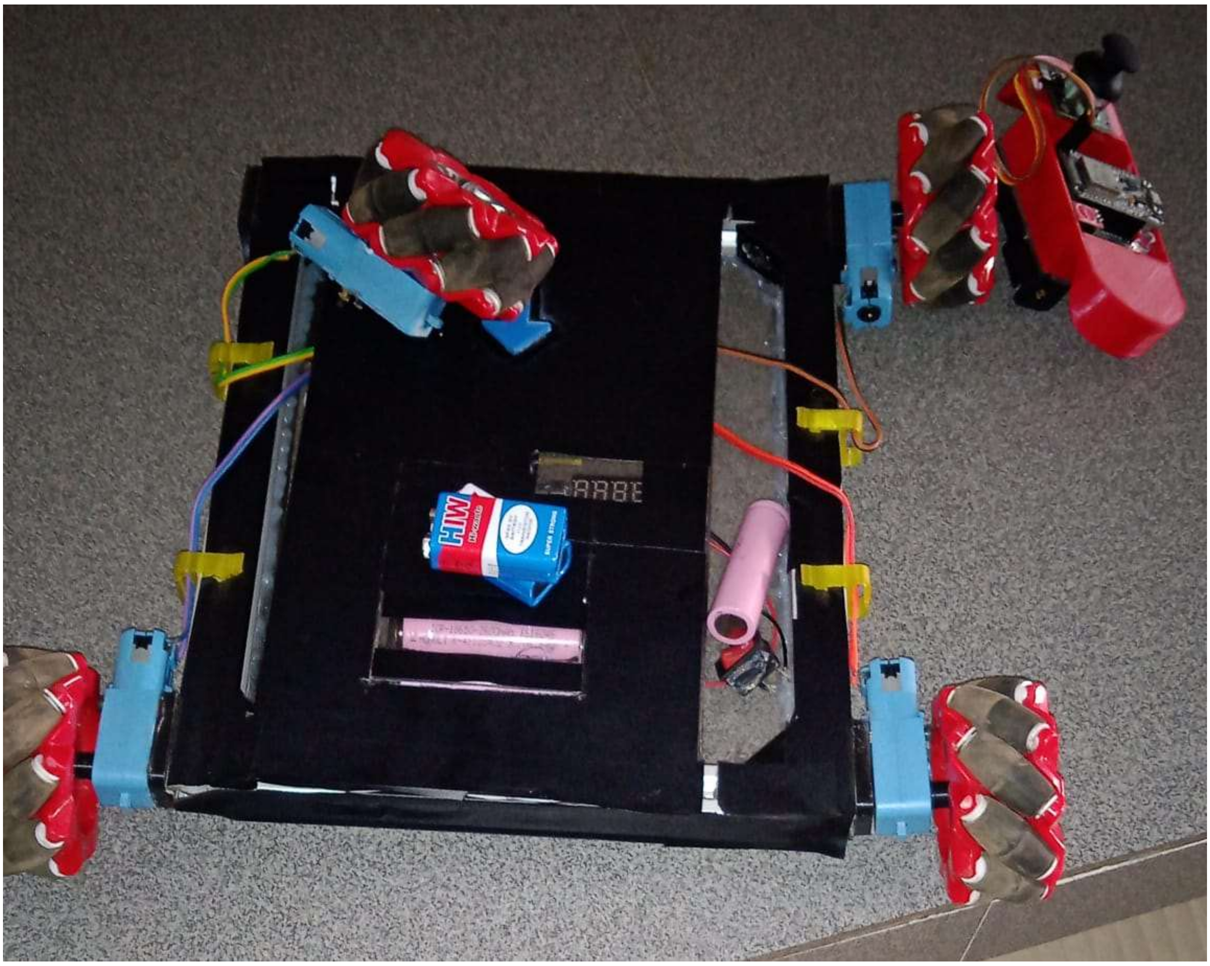

Fig 8.4

9. ALTERNATIVE OF MECCANUM WHEELS

9.1 <u>OMNIWHEELS:</u>

Omni wheels, also known as omni-directional wheels or polywheels, are specialized wheels designed to provide omnidirectional movement. They consist of a main wheel with multiple smaller rollers or wheels mounted around its circumference, perpendicular to the main wheel's axis. These rollers or wheels allow the omni wheel to move laterally and rotate around its own axis, providing the ability to move in any direction without changing the orientation of the vehicle or platform.

Fig 9.1.1

Design and Mechanism:

1. **Wheel Structure**: Omni wheels consist of multiple rollers or wheels arranged around the circumference of a larger central wheel. The rollers are typically mounted perpendicular to the central wheel's axis of rotation.
2. **Roller Arrangement**: Omni wheels can have different roller arrangements, including straight rollers or angled rollers. Angled rollers allow for smoother movement and better omnidirectional capabilities.
3. **Roller Types**: Omni wheels can have either passive or driven rollers. Passive rollers freely rotate and allow the wheel to move in any direction, while driven rollers are powered to provide additional control and propulsion.

9.2 DIFFERENCE BETWEEN MECANUM WHEEL AND OMNI WHEEL:

Fig 9.2.1

Mecanum wheels and omni wheels are both types of wheels designed for omnidirectional movement, but they differ in their design, operation, and applications. Here are the main differences between mecanum wheels and omni wheels:

1. **Design:**
   - Mecanum Wheels: Mecanum wheels have a unique design with rollers mounted at an angle around the circumference of the wheel. These rollers allow the wheel to move in any direction by varying the speed and direction of rotation of individual wheels.
   - Omni Wheels: Omni wheels have a series of freely rotating rollers or wheels around the circumference of the main wheel. These rollers allow the wheel to move laterally and rotate around its axis, providing omnidirectional movement.
2. **Roller Orientation:**
   - Mecanum Wheels: The rollers on mecanum wheels are mounted at an angle (typically 45 degrees) to the wheel's axis. This angled orientation allows the

wheels to generate lateral force components, enabling omnidirectional movement and maneuverability.

- Omni Wheels: The rollers on omni wheels are typically mounted perpendicular to the wheel's axis. This orientation allows the wheel to move laterally without resistance and rotate around its axis, providing omnidirectional movement.

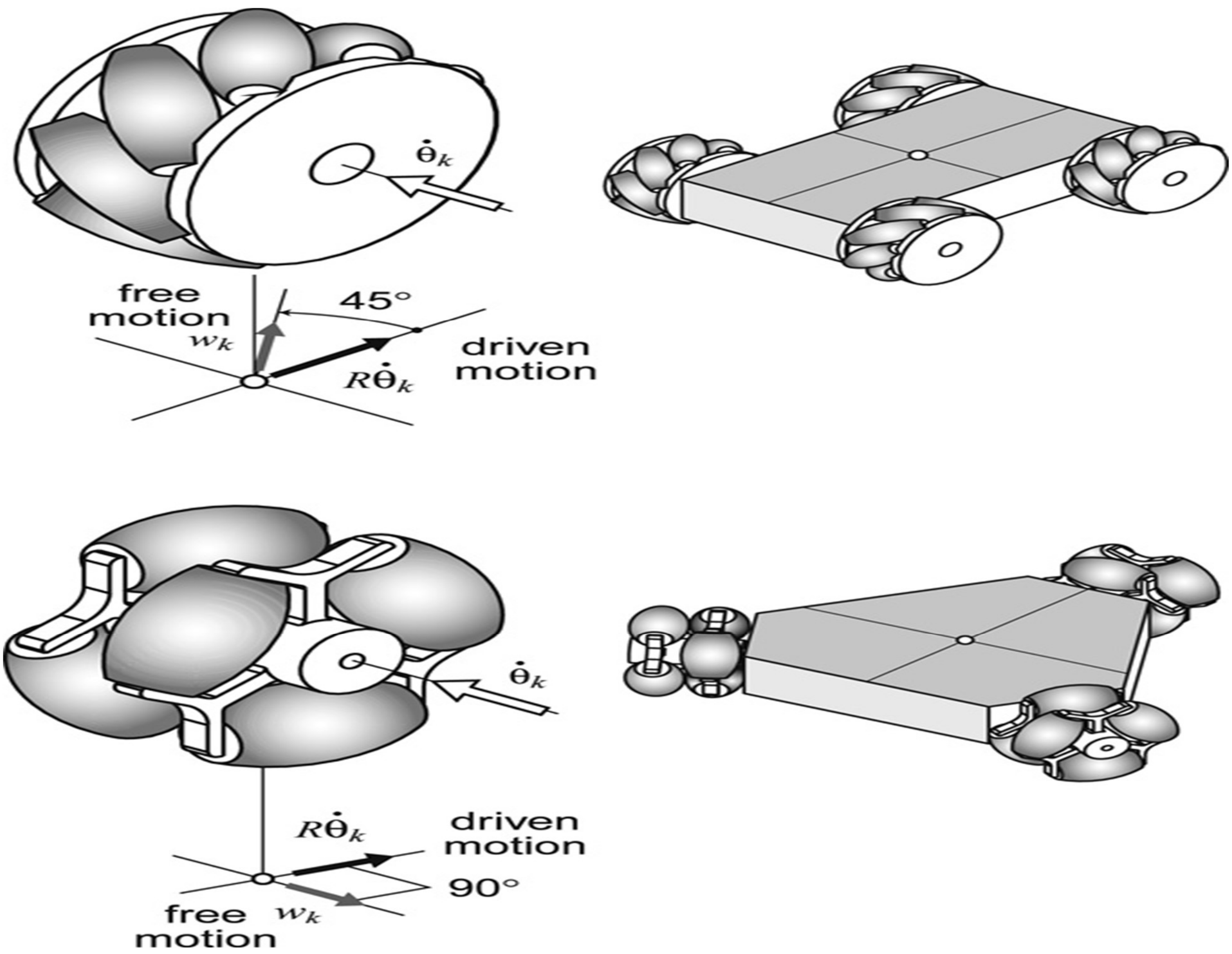

Fig 9.2.2

3. **Control and Maneuverability:**
   - Mecanum Wheels: Mecanum wheels offer more flexibility and control over movement compared to omni wheels. They can move in any direction, including sideways and diagonally, by independently controlling the speed and direction of rotation of each wheel.

- Omni Wheels: Omni wheels provide omnidirectional movement but may have limitations in terms of maneuverability compared to mecanum wheels. They are better suited for simple lateral movements and rotations.

4. **Load Capacity:**
    - Mecanum Wheels: Mecanum wheels typically have higher load capacities compared to omni wheels, especially in applications where heavy loads need to be transported or maneuvered.
    - Omni Wheels: Omni wheels may have lower load capacities compared to mecanum wheels due to their design and roller orientation.
5. **Applications:**
    - Mecanum Wheels: Mecanum wheels are commonly used in robotics, industrial automation, material handling, and mobile platforms where precise and agile omnidirectional movement is required.
    - Omni Wheels: Omni wheels are used in various applications such as conveyor systems, assembly lines, robotic platforms, and mobile robots where simple lateral movement and rotation are sufficient.

10.PROTOTYPE

Final assembly of product is shown in figures:

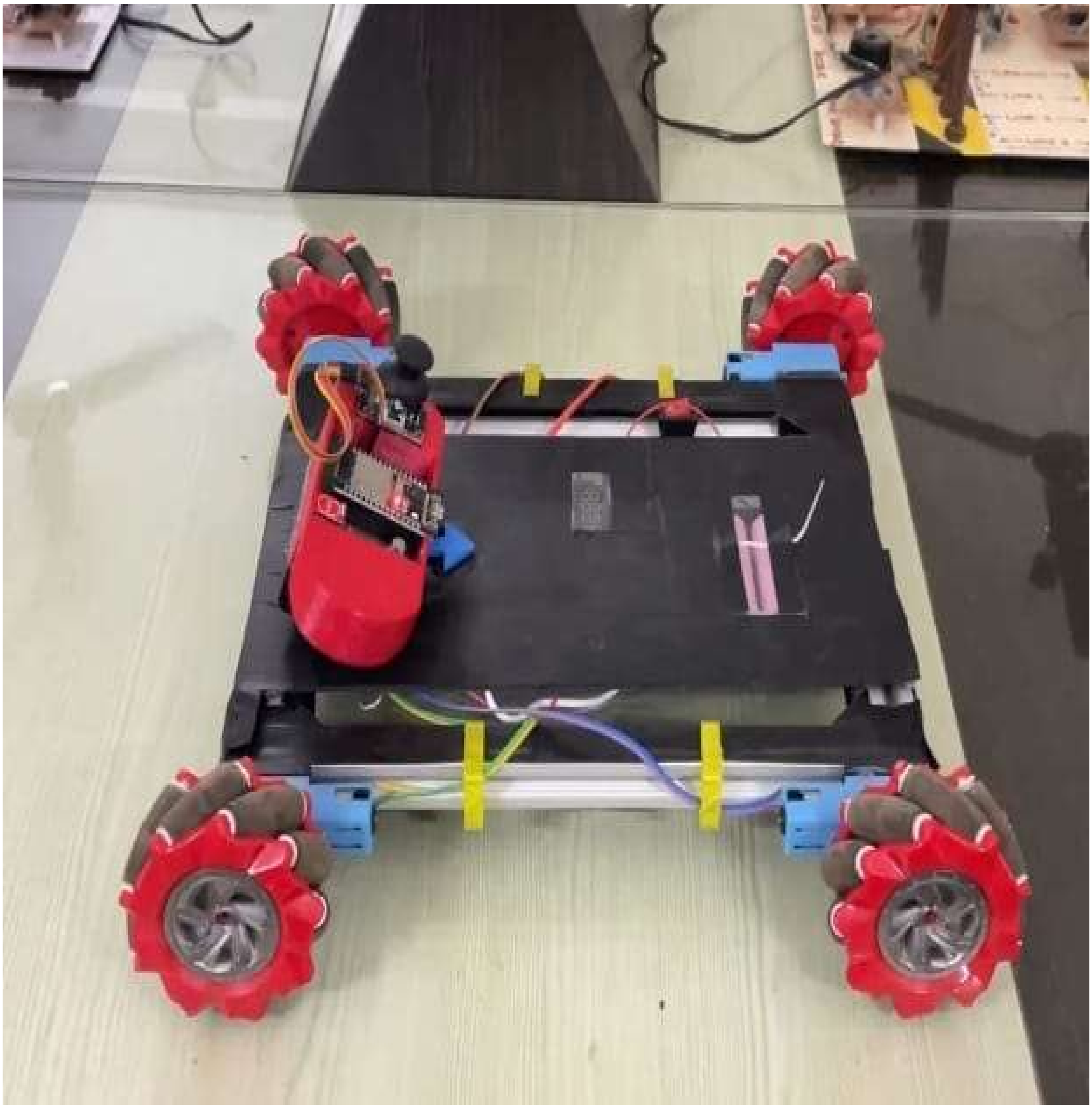

Fig 10.1

Fig 10.2

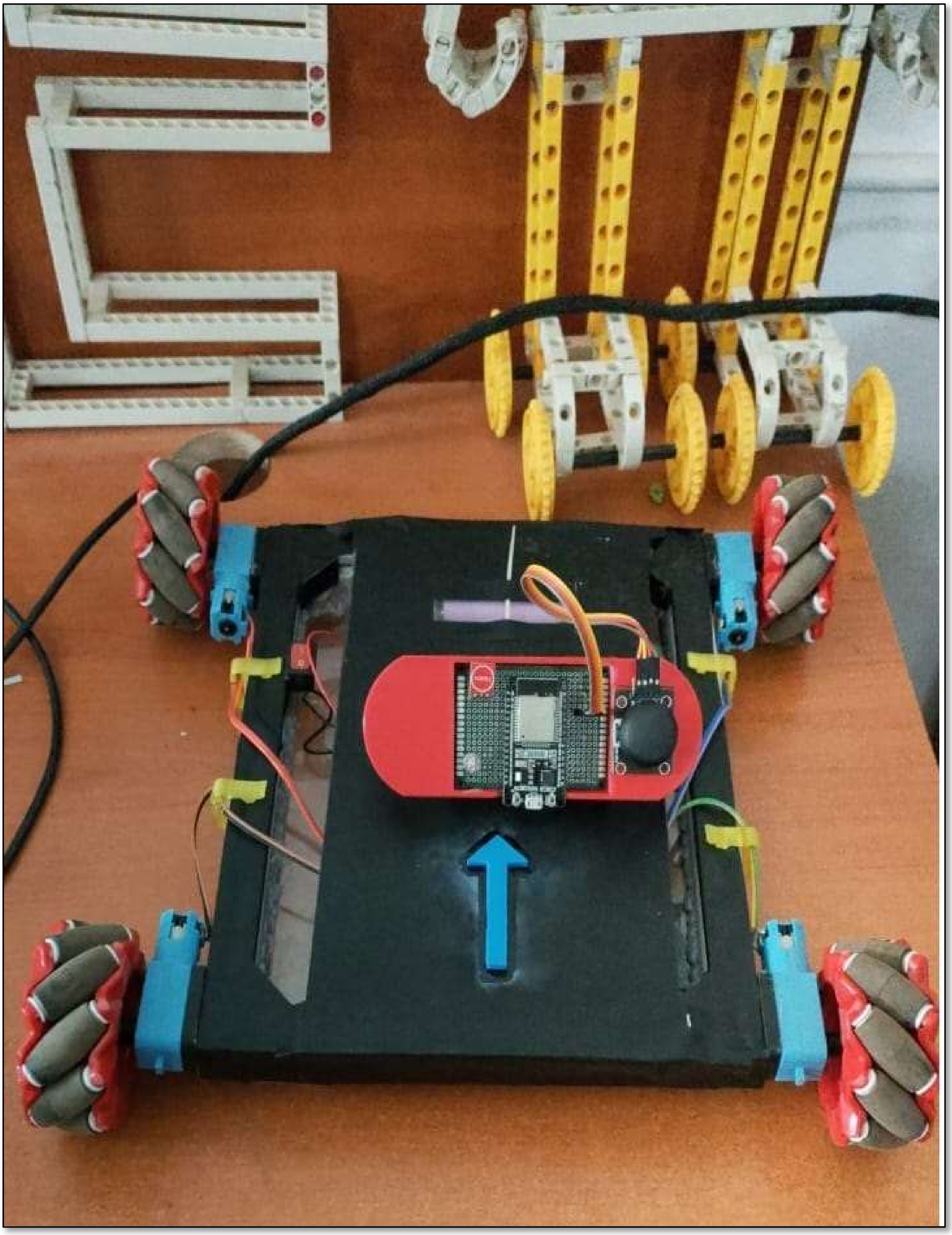

Fig 10.3

11.CONCLUSION

This Project explores the cutting-edge design and implementation of meccanum wheel-equipped remote-controlled vehicles, harnessing the capabilities of the ESP32 microcontroller and establishing seamless communication through peer-to-peer WiFi networks. Meccanum wheels, known for their omnidirectional movement, provide enhanced maneuverability. The ESP32 microcontroller acts as the central processing unit, orchestrating precise control of each meccanum wheel for unparalleled agility.

In conclusion, the design and implementation of Meccanum wheel remote-controlled vehicles using ESP32 and peer-to-peer WiFi communication present an opportunity to create agile and versatile platforms for various applications. By carefully considering factors such as mechanical design, electronics integration, wireless communication, control algorithms, and user interface, engineers can develop vehicles capable of omnidirectional movement with precision and responsiveness. The ESP32 microcontroller's capabilities, coupled with peer-to-peer WiFi communication, provide a robust platform for remote control and data exchange, enabling seamless interaction between the vehicle and the user. Iterative testing and refinement are essential to ensure the vehicle's reliability, performance, and user experience meet the project's objectives. Overall, the design and implementation of Meccanum wheel remote-controlled vehicles offer exciting possibilities for robotics, automation, and research, pushing the boundaries of mobility and control in diverse environments.

12.REFERNCE

1. International Research Journal of Engineering and Technology (IRJET) by Gajanan Chavan.
2. journal of field robotics and electronics. @google
3. Pictures from Wikimedia.org , en.wikibooks.org , google.
4. Videos From “Drone Bot Workshop” @Youtube.

13.APPENDIX

```
// Define Motor Connections
// Right Front Motor
#define MF_PWMA 19
#define MF_AI1 32
#define MF_AI2 23
// Left Front Motor
#define MF_PWMB 26
#define MF_BI1 33
#define MF_BI2 25
// Right Rear Motor
#define MR_PWMA 27
#define MR_AI1 12
#define MR_AI2 14
// Left Rear Motor
#define MR_PWMB 4
#define MR_BI1 13
#define MR_BI2 2
// Define some preset motor speeds (0 - 255, adjust as desired)
int speed_slow = 100;
int speed_fast = 250;
void setup() {
  // Set all connections as outputs
  pinMode(MF_PWMA, OUTPUT);
  pinMode(MF_AI1, OUTPUT);
  pinMode(MF_AI2, OUTPUT);
  pinMode(MF_PWMB, OUTPUT);
  pinMode(MF_BI1, OUTPUT);
  pinMode(MF_BI2, OUTPUT);
  pinMode(MR_PWMA, OUTPUT);
  pinMode(MR_AI1, OUTPUT);
```

```
  pinMode(MR_AI2, OUTPUT);
  pinMode(MR_PWMB, OUTPUT);
  pinMode(MR_BI1, OUTPUT);
  pinMode(MR_BI2, OUTPUT);
}
void loop() {
  // Front Right
Motor  **************************************************************
  // FR - Forward Slow Speed
  digitalWrite(MF_AI1, HIGH);
  digitalWrite(MF_AI2, LOW);
  analogWrite(MF_PWMA, speed_slow);
  delay(2000);
  // FR - Forward Fast Speed
  analogWrite(MF_PWMA, speed_fast);
  delay(2000);
  // FR - Stop 1 second
  analogWrite(MF_PWMA, 0);
  delay(1000);
  // FR - Reverse Fast Speed
  digitalWrite(MF_AI1, LOW);
  digitalWrite(MF_AI2, HIGH);
  analogWrite(MF_PWMA, speed_fast);
  delay(2000);
  // FR - Reverse Slow Speed
  analogWrite(MF_PWMA, speed_slow);
  delay(2000);
  // FR - Stop 1 second
  analogWrite(MF_PWMA, 0);
  delay(1000);
```

```
 // Front Left
Motor *************************************************************

 // FL - Forward Slow Speed

 digitalWrite(MF_BI1, HIGH);

 digitalWrite(MF_BI2, LOW);

 analogWrite(MF_PWMB, speed_slow);

 delay(2000);

 // FL - Forward Fast Speed

 analogWrite(MF_PWMB, speed_fast);

 delay(2000);

 // FL - Stop 1 second

 analogWrite(MF_PWMB, 0);

 delay(1000);

 // FL - Reverse Fast Speed

 digitalWrite(MF_BI1, LOW);

 digitalWrite(MF_BI2, HIGH);

 analogWrite(MF_PWMB, speed_fast);

 delay(2000);

 // FL - Reverse Slow Speed

 analogWrite(MF_PWMB, speed_slow);

 delay(2000);

 // FL - Stop 1 second

 analogWrite(MF_PWMB, 0);

 delay(1000);

 // Rear Right
Motor *************************************************************

 // RR - Forward Slow Speed

 digitalWrite(MR_AI1, HIGH);

 digitalWrite(MR_AI2, LOW);

 analogWrite(MR_PWMA, speed_slow);

 delay(2000);
```

```
// RR - Forward Fast Speed
analogWrite(MR_PWMA, speed_fast);
delay(2000);
// RR - Stop 1 second
analogWrite(MR_PWMA, 0);
delay(1000);
// RR - Reverse Fast Speed
digitalWrite(MR_AI1, LOW);
digitalWrite(MR_AI2, HIGH);
analogWrite(MR_PWMA, speed_fast);
delay(2000);

// RR - Reverse Slow Speed
analogWrite(MR_PWMA, speed_slow);
delay(2000);
// RR - Stop 1 second
analogWrite(MR_PWMA, 0);
delay(1000);
// Rear Left Motor ****************************************************************
// RL - Forward Slow Speed
digitalWrite(MR_BI1, HIGH);
digitalWrite(MR_BI2, LOW);
analogWrite(MR_PWMB, speed_slow);
delay(2000);
// RL - Forward Fast Speed
analogWrite(MR_PWMB, speed_fast);
delay(2000);
// RL - Stop 1 second
analogWrite(MR_PWMB, 0);
delay(1000);
// RL - Reverse Fast Speed
```

```
  digitalWrite(MR_BI1, LOW);
  digitalWrite(MR_BI2, HIGH);
  analogWrite(MR_PWMB, speed_fast);
  delay(2000);
  // RL - Reverse Slow Speed
  analogWrite(MR_PWMB, speed_slow);
  delay(2000);

  // RL - Stop 1 second
  analogWrite(MR_PWMB, 0);
  delay(1000);
}
```