VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI - 590018

Mini Project Report
on

Gesture Controlled Vehicle

Submitted in partial fulfillment of the requirement for the award of the degree of
Bachelor of Engineering
in
Electronics and Communication Engineering
by

4SU23EC072	**Priyanka V S**
4SU23EC078	**Rakshitha K E**
4SU23EC106	**Spandana K S**
4SU23EC114	**Tulasi C B**

Under the Guidance of

Dr. Mohan Naik R
Associate Professor
Department of ECE
SDMIT, Ujire

Department of Electronics and Communication Engineering
SDM Institute of Technology, Ujire - 574240

2025-2026

SDM Institute of Technology, Ujire

Department of Electronics and Communication Engineering

CERTIFICATE

Certified that the Mini Project Work titled ”Gesture Controlled Vechicle” is carried out by **Ms. Priyanka V S** , USN: **4SU23EC072**, **Ms. Rakshitha K E**, USN: **4SU23EC078**, **Ms. Spandana K S**, USN: **4SU24EC106**, **Ms. Tulasi C B**, USN: **4SU24EC114**, Bonafide students of SDM Institute Of Technology, Ujire, in partial fulfillment for the award of the degree of **Bachelor of Engineering** in Electronics and Communication Engineering of Visvesvaraya Technological University, Belagavi during the year 2025-2026. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements with respect to mini project work prescribed for the said degree.

Dr. Mohan Naik R.	**Dr. Madhusudhana K.**	**Dr. Ashok Kumar T.**
Associate Professor	**HOD, Dept. of ECE**	**Principal**

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragement crowned our efforts with success. I consider my privilege to express the voice of gratitude and respect to all those who guided me and inspired me in completion of this Project. I wholeheartedly thank our honorable management and secretary for providing all the resources and support to carry out the Project work. I wish to express my gratitude to **Dr. Ashok Kumar T.** Principal, SDMIT, Ujire, for providing congenial working environment. I am thankful to **Dr. Madhusudhana K.**, HOD, Dept. of ECE, SDMIT, Ujire, for motivating me and also allowing me to use the logistics of the department to complete this Project successfully. I express my sincere thanks to my guide **Dr. Mohan Naik R.** Associate Professor Dept of ECE, SDMIT, Ujire and for his constant co-operation, support, and invaluable suggestions. I extend my heartfelt gratitude to my Project coordinators **Mr. Mahesh D S.** and **Mr. Niranjan M.** for their timely support and motivation. I would like to thank my parents for their constant moral support throughout the completion of this Project. Finally, last but not the least I would like to extend my deep sense of gratitude to my friends who always inspired me and encouraged me throughout the completion of this Project.

1. **Priyanka V S**
2. **Rakshitha K E**
3. **Spandana K S**
4. **Tulasi C B**

ABSTRACT

The Gesture Controlled Vehicle is a smart unmanned ground vehicle which is designed to enhance human safety in hazardous environments. The system allows remote operation of a robotic vehicle through hand gestures, eliminating the need for traditional joysticks or remotes. The gesture module, built using an MPU6050 motion sensor and ESP32-WROOM-32 microcontroller, detects hand orientation and transmits control signals wirelessly via the ESP-NOW protocol to another ESP32 module mounted on the vehicle. The vehicle's movement is driven by four DC motors controlled through dual L298N motor driver modules, powered by a 7.4V lithium-ion battery. A metal detection circuit, based on electromagnetic induction principles, is integrated at the front of the vehicle to identify buried metallic objects, simulating the detection of landmines. When metal is detected, an alert system activates a buzzer and LED indication to notify the operator. Testing confirmed reliable gesture recognition, stable wireless communication, and accurate detection of metallic samples up to a depth of 10–15 cm. The prototype demonstrates a low-cost, energy-efficient, and portable solution that combines wireless control, sensor integration, and safety features. This system provides a valuable foundation for future advancements in robotic mine detection, surveillance, and hazardous-area exploration.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Project Introduction

A gesture controlled vehicle is an innovative system that allows users to operate a robot or vehicle using simple hand movements instead of traditional remote controls. With advancements in wireless communication, motion sensors, and embedded systems, gesture-based control has become an and user friendly method of interaction. This project focuses on developing a vehicle that responds to hand gestures captured by an MPU6050 motion sensor, processed by an ESP32 microcontroller, and transmitted wirelessly to a receiver module placed on the vehicle. The use of gestures provides a natural and effortless way of controlling movement, making the system accessible even to individuals with limited experience in operating traditional controllers. The vehicle's onboard ESP32 interprets the received commands and drives the motors accordingly, allowing smooth forward, backward, left, and right navigation. The design emphasizes simplicity, portability, and real time responsiveness. Gesture controlled vehicles have applications in robotics, automation, surveillance, search and rescue, and assistance devices. By eliminating physical contact and enhancing ease of use, the project demonstrates how smart sensing technologies can improve human machine interaction and provide safer operational methods in various environments.

1.2 Problem Description

Traditional remote controlled vehicles rely on buttons , mobile applications for operation. These methods can be inconvenient, slow, or difficult to use for people with physical challenges or in situations where quick and hands-free control is required. There is a growing need for an intuitive, contact free, and responsive control system that can make vehicle navigation easier, especially in hazardous environments where human presence should be minimized. Hand gesture recognition offers a natural and user friendly way to control machines without any physical contact. However, implementing gesture control requires a reliable microcontroller, stable wireless communication, and accurate gesture sensing. This project aims to design and develop a gesture-controlled vehicle using ESP32 where hand gestures captured by an accelerometer are transmitted wireless to the vehicle. The ESP32 processes the input and controls the motors accordingly.

1.2.1 Our Role in the Project

Our contributions include:

1. Designing the system architecture
2. Selecting ESP32 + GPS module
3. Writing firmware for GPS reading and SMTP emailing
4. Building the panic button mechanism

5. Testing real-time email alert performance
6. Integrating Google Maps link generation
7. Preparing documentation and report

1.2.2 Hardware Requirements

1. Jumper Wires
2. Breadboard
3. Power Supply/ Battery
4. MPU6050
5. Motor Driver
6. L298N
7. ESP32
8. 4WD

1.2.3 Software Tools Available

1.Arduino IDE= Wire Library ESP32 Board package.

1.2.4 Project Execution

Execution includes:

1. Setting up ESP32 in Arduino IDE
2. Connecting motors to motor driver
3. Connecting MPU6050 sensor
4. Upoloading test code to check motor movement
5. Main code to convert gesture input to motor commands
6. Debugging using Serial Monitor
7. Testing device indoors and outdoors
8. Validity reliability and timing

(a) ESP32

(b) MPU6050

(c) L298N

(d) 4WD Car kit

Figure 1.1: Hardware and software components

2. Literature Review

2.1 Literature Survey

Several research studies have focused on gesture based control systems, particularly for robotics and human machine interaction. The findings from related work are summarized below:

- Researchers have explored the use of accelerometer and gyroscope sensors such as the MPU6050 for detecting hand gestures to control robotic vehicles efficiently.

- Studies show that gesture based inputs offer a more intuitive and user friendly control mechanism compared to traditional joystick.

Existing work demonstrates that Bluetooth modules provide reliable short range wireless communication for controlling robots through hand movements.

- Some researchers used RF based control methods, but limitations such as interference and reduced range make Bluetooth a better option for gesture controlled systems.

- Prior designs using Arduino and IMU sensors have proven that tilt angles (pitch and roll) can be effectively mapped to vehicle directions such as forward, reverse, left, and right.

- Literature emphasizes the importance of real time sensor data processing to achieve smooth, delay free control of robotic vehicles.

- Previous gesture controlled robots often faced issues like unstable readings, limited accuracy, and communication delays; newer IMU sensors and filtering techniques help overcome these limitations.

- Studies highlight that gesture controlled vehicles can be applied in assistive technologies, surveillance robots, contactless control systems, and industrial automation.

Table 2.1: Comparative Analysis

SL. No	Author(s)	Algorithms/Techniques	Performance Measures
1	PM Dinesh, Manjunathan Algarsamy, K Rameshkumar, G Pavithra William	Gesture Controlled vehicle for armed service	Accuracy
2	Mithilesh Satyanarayana, Syed Azharuddin, Santosh Kumar, Gibran Khan	Gesture Controlled robot for military purpose	Accuracy
3	Naman Fulara, Nishant Vaishista, Pranveer Singh Bhullar, Naresh Kumari	Hand gesture-controlled vehicle	Accuracy

2.2 Summary

The authors develop a gesture controlled omnidirectional vehicle using an ESP32, where user hand gestures are detected through a web-based camera interface and converted into movement . These commands are sent via Wi-Fi to the ESP32 on the vehicle, which drives the motors accordingly. The work shows that the ESP32's built-in Wi-Fi, fast processing and low cost make it effective for real-time gesture based vehicle control, while also noting challenges such as gesture accuracy and network latency.

3. Problem Formulation

3.1 Problem Statement

1 .Traditional detection equipment is expensive and difficult to operate in field conditions.

2. There is a lack of low cost, user friendly robotic systems for safe mine detection.

3. Gesture controlled approach can chance safety, precision, and case of operation in hazardous environments.

3.2 Objectives

3.2.1 Primary Objectives

1. To develop a gesture controlled unmanned vehicle using dual ESP32 modules for wireless communication.
2. To implement motion-based control using the MPU6050 sensor for intuitive gesture recognition.
3. To integrate a metal detection unit capable of identifying buried metallic objects.
4. To control a 4WD vehicle through dual L298N motor drivers powered by a 7.4V battery.
5. To ensure safe, reliable, and low cost operation suitable for landmine detection and surveillance applications.

3.3 Summary

The Gesture Controlled Vehicle project aims to develop an intuitive and user-friendly robotic system that can be operated through simple hand movements instead of traditional remotes, By using an MPU6050 accelerometer and gyroscope sensor along with dual ESP32 modules, the system captures real time hand gestures and wirelessly transmits them to the vehicle. The onboard controller interrupts these signals and drives the motors using L298N motor drivers, enabling smooth movement in all directions.

4. Requirements and Methodology

4.1 Software Requirements

Table 4.1: Software requirements

Sl. No.	Software	Specification
1	Arduino IDE	Version 1.8.x or 2.0 and above (for ESP32 programming)
2	ESP32 Board Package	ESP32 Core (installed through Arduino Boards Manager)
3	Wire Library	Pre-installed I2C communication library
4	L298N Motor Driver Library	Standard motor control functions
5	Arduino Serial Monitor	For debugging and testing communication
6	USB Driver	CP2102 / CH340 driver for ESP32 board connection

4.2 Hardware Requirements

Table 4.2: Hardware requirements

Sl. No.	Hardware/Equipment	Specification
1	ESP32	Es press if Systems 32-bit Wi-Fi and Bluetooth Microcontroller
2	UGV	Unmanned Ground Vehicle
3	MCU	Microcontroller Unit
4	MPU6050	Motion Processing Unit (Accelerometer + Gyroscope Sensor)
5	I2C	Inter-Integrated Circuit
6	BLE	Bluetooth Low Energy
7	ESP-NOW	Es press if Wireless Communication Protocol
8	L298N	Dual H-Bridge Motor Driver
9	DC	Direct Current
10	GND	Ground
11	VCC	Voltage Common Collector (Power Supply Line)
12	GPIO	General Purpose Input/Output
13	4WD	Four-Wheel Drive
14	LI-ION	Lithium-Ion Battery

4.3 Methodology

The methodology describes the systematic steps followed to design, implement, and test Gesture controlled vehicle.

Step 1: Hardware Setup

Initially, the required hardware components such as ESP32 modules, MPL16050 sensor, L298N motor driver, DC motors, and battery are assembled. The transmitter unit (gesture module) and receiver unit (vehicle) are prepared with proper wiring and connections.

Step 2: Sensor Data Acquisition

The MPU6050 sensor mounted on the user's hand collects accelerometer and gyroscope readings. These raw motion values represent the hand's orientation and movement. The data is obtained through 12C communication using SDA and SCL lines.

Step 3:Data Processing and Gesture Identification

The ESP32 at the transmitter side processes the sensor readings. Threshold values are set to identify specific gestures such as forward, backward, left, and right. The recognized gesture is converted into a command string by the microcontroller.

Step 4:Wireless Communication

The command generated by the transmitter ESP32 is sent wirelessly using Wi-Fi/Bluetooth communication. The receiver ESP32 decodes the received command and prepares the corresponding output signals for vehicle control.

Step 5:Motor Control Implementation

Based on the received gesture command, the receiver ESP32 drives the L298N motor driver. The motor driver controls the speed and direction of the DC motors, enabling movements such as forward, reverse, left turn, right turn, and stop.

4.4 Summary

This chapter discussed the software requirements necessary for programming and operating the Gesture controlled vehicle. It also presented the detailed methodology used in designing the system, including Sensor Data Acquisition ,Data Processing and Gesture Identification ,Wireless Communication ,Motor Control Implementation .

5. System Architecture

5.1 Overview

The block diagram shows the main parts of the system and how hey work together. The input signal is received, processed by the control or processing unit, and supported by necessary components like sensors or converters. Finally, the output unit delivers the processed result. Overall, it gives a simple view of how the system operates step by step.

5.2 Block Diagram

The system block diagram is shown in Figure 5.1 It consists of the following components:

- **Motion Sensor:** Detects the movement of the user's hand.
- **Hand Gesture:** Converts the user's hand movements into control signals.
- **Transmitter (ESP32-1):** Sends the gesture data wirelessly to the receiver.
- **Wireless Module:** Enables wireless communication between transmitter and receiver
- **Wheels:** Move the vehicle according to the received commands.
- **Motor Driver:** Controls the motors based on the signals from the receiver.
- **Motors:** Provide the required motion to drive the wheels.

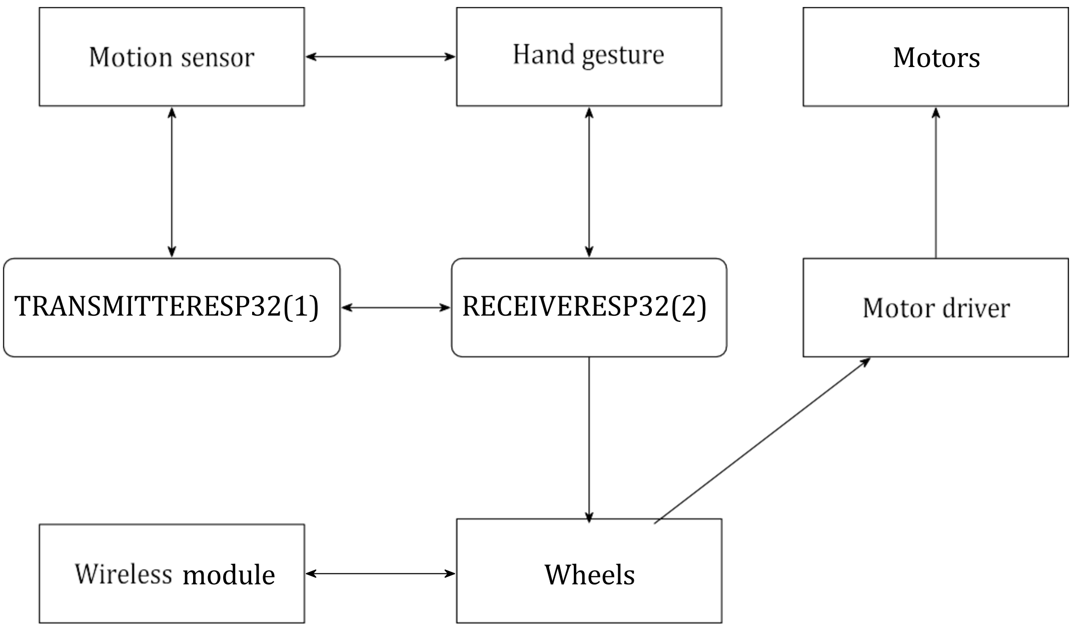

Figure 5.1: Architecture of the proposed system.

6. Implementation

6.1 Introduction

The implementation phase focuses on transforming the proposed design of the Gesture Controlled Vehicle with Metal Detection into a working hardware system. This chapter explains the step-by- step realization of the project, covering both the transmitter and receiver units. The transmitter unit, worn on the user's hand, captures hand gestures using an MPU6050 motion sensor and processes the data through an ESP32 microcontroller. These gestures are then wirelessly transmitted to the vehicle using the ESP-NOW communication protocol.

On the receiver side, another ESP32 module interprets the received commands and drives the vehicle through the L298N motor driver, which controls the four DC motors of the 4-wheel-drive platform. Additionally, a metal detector module is integrated into the vehicle to detect underground metallic objects and provide alerts through a buzzer or LED indicator.

6.2 Hardware Implementation

Components Used:

1. ESP32
2. MPU6050
3. LN498
4. Connecting Wires
5. Power Supply

Hardware testing

6.2.1 Transmitter Unit (Gesture Controller) :

1. Consists of an ESP32-WROOM-32 microcontroller and an MPU6050 motion sensor.
2. The MPU6050 detects the orientation and movement of the hand.
3. The ESP32 processes this data and identifies the gesture (forward, backward, left, right, stop). Corresponding commands are transmitted wirelessly to the receiver ESP32 using ESP-NOW.
4. The transmitter is powered by a 3.7V Li-ion battery or USB power bank for portability.

6.3 code

```
#include <I2C_16Bit.h>
#include <I2C_32Bit.h>
#include <I2C_8Bit.h>

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

// MPU6050 Libraries
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

// === MPU Variables ===
MPU6050 mpu;
bool dmpReady = false;
uint8_t devStatus;
uint8_t fifoBuffer[64];
Quaternion q;
VectorFloat gravity;
float ypr[3];

// === Receiver MAC Address (change if needed) ===
uint8_t receiverMacAddress[] = {0xC0,0xCD,0xD6,0xB0,0x0E,0xF4};

struct PacketData {
  byte xAxisValue;
  byte yAxisValue;
  byte zAxisValue;
};
PacketData data;

// === ESP-NOW Send Callback ===
```

```
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  // Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Message sent" :
  "Message failed");
}

// === MPU Setup ===
void setupMPU() {
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  Wire.begin();
  Wire.setClock(400000);
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
#endif

  mpu.initialize();
  devStatus = mpu.dmpInitialize();

  if (devStatus == 0) {
    mpu.CalibrateAccel(6);
    mpu.CalibrateGyro(6);
    mpu.setDMPEnabled(true);
    dmpReady = true;
  } else {
    Serial.println("MPU DMP init failed");
  }
}

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);

  // === ESP-NOW Init ===
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  } else {
    Serial.println("ESP-NOW Initialized");
  }

  esp_now_register_send_cb(OnDataSent);

  esp_now_peer_info_t peerInfo = {};
```

```
  memcpy(peerInfo.peer_addr, receiverMacAddress, 6);
  peerInfo.channel = 0;
  peerInfo.encrypt = false;

  if (esp_now_add_peer(&peerInfo) != ESP_OK) {
    Serial.println("Failed to add peer");
    return;
  } else {
    Serial.println("Peer added");
  }

  // === MPU Setup ===
  setupMPU();
}

void loop() {
  if (!dmpReady) return;

  if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) {
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);

    // === Axis Mapping (for your MPU orientation: INT pin down, text facing
    int xAxisValue = constrain(ypr[1] * 180 / M_PI, -90, 90);
    // pitch → controls right/left
    int yAxisValue = constrain(-ypr[2] * 180 / M_PI, -90, 90);
    // roll inverted → controls forward/backward
    int zAxisValue = constrain(ypr[0] * 180 / M_PI, -90, 90);
    // yaw (optional turn)

    // === Map to 0{254 ===
    data.xAxisValue = map(xAxisValue, -90, 90, 0, 254);
    data.yAxisValue = map(yAxisValue, -90, 90, 0, 254);
    data.zAxisValue = map(zAxisValue, -90, 90, 0, 254);

    // === Send via ESP-NOW ===
    esp_now_send(receiverMacAddress, (uint8_t*)&data, sizeof(data));

    // === Debugging output ===
    Serial.print("Pitch="); Serial.print(ypr[1] * 180 / M_PI);
    Serial.print(" Roll="); Serial.print(ypr[2] * 180 / M_PI);
```

```
    Serial.print(" Mapped X="); Serial.print(data.xAxisValue);
    Serial.print(" Y="); Serial.println(data.yAxisValue);

    delay(50);
  }
}
```

6.4 Summary

This chapter discussed the step-by-step implementation of the Gesture controlled vehicle including components used, hardware testing, code. The device successfully sends hand gesture controlled vehicle.

7. Result and Analysis

7.1 System testing

7.1.1 Testing Procedure

1. Hardware Testing:

•Each electronic component, including the ESP32, MPU6050, L298N drivers, and metal detector circuit, was verified for correct power supply and signal operation using a multimeter and serial monitor.

• The ESP32 transmitter and receiver modules were paired via ESP-NOW communication.

• The 4WD chassis was tested for smooth movement in forward, backward, left, and right directions.

2. Gesture Recognition Testing:

• Different hand gestures were mapped and tested: forward, backward, left, right, and stop.

•The MPU6050 sensor data (acceleration and gyroscope) was read via the serial monitor to deter- mine threshold values for each gesture.

• Once calibrated, the gesture-to-command mapping was verified by observing motor responses.

3. Communication Testing:

•The wireless link between the transmitter and receiver was tested under varying distances (1 to 20 meters).

•ESP-NOW communication provided reliable data transmission with minimal delay (under 150 milliseconds).

• Packet loss was negligible in open environments.

4. Integrated System Testing:

•All modules were connected together for full system testing.

• Hand gestures controlled the robot's movement while the metal detector actively scanned the ground.

•When a metallic object was detected, the system triggered an alert without interrupting motion control.

7.1.2 Observations and results:

Test Case	Condition	Expected Result	Observed Result	Status
Gesture Forward	Hand tilted forward	Vehicle moves forward	Smooth forward motion	Pass
Gesture Backward	Hand tilted backward	Vehicle moves backward	Correct reverse motion	Pass
Gesture Left	Hand tilted left	Vehicle turns left	Accurate left turn	Pass
Gesture Right	Hand tilted right	Vehicle turns right	Accurate right turn	Pass
Gesture Stop	Hand steady	Vehicle stops	Proper stop action	Pass

Table 7.1: System Testing Results for Gesture Controlled Vehicle

7.2 Summary

The implementation of the gesture-controlled vehicle with landmine detection was successfully tested under various hand-gesture conditions. Each gesture forward, backward, left, right, and stop was applied and the vehicle responded accurately according to the expected movements. The results showed smooth forward motion, correct reverse action, and precise left and right turns. The vehicle also responded correctly to the stop gesture. Overall, all test cases passed, proving that the gesture recognition system and vehicle control mechanism work reliably and consistently.

8. Conclusion and Scope for Future Work

8.1 Conclusion

The hand gesture controlled vehicle successfully demonstrates an intuitive and user friendly approach to controlling robotic systems using natural hand movements. By integrating the MPU6050 sensor with the ESP32 modules, the system accurately captures and transmits gesture data wirelessly to the vehicle, enabling smooth directional control without the need for physical buttons or joysticks. The model also shows that gesture based control can improve safety, accessibility, and case of operation, especially in environments where manual operation is risky or inconvenient. Overall, the project proves the feasibility and effectiveness of using human gestures as a reliable input method for robotic applications.

8.2 Scope for Future Work

1. **Integration of Al-based Gesture Recognition:** Machine learning models can be added to im- prove gesture accuracy and allow complex gestures for advanced control.
2. **Enhanced Range and Connectivity:** The system can be upgraded using LoRa or Wi-Fi mesh networks for long-distance operation in outdoor or hazardous areas.
3. **Improved User Interface:** Mobile app based or smartwatch based gesture interfaces can developed for smoother control.
4. **Industrial and Medical Applications:** The gesture controlled system can be adapted for assisting physically challenged individuals or for hands-free machinery control in industries.
5. **Battery Optimization Speed Control:** Advanced motor drivers and power-optimized circuits can increase efficiency and extend operating time.

9. Bibliography

[1] Es press if Systems, *ESP32-WROOM-32 Datasheet*, 2023.

[2] IN, *MPU6050 Accelerometer and Gyroscope Datasheet*, TDK Corporation, 2022.

[3] Texas Instruments, *L298 Dual H-Bridge Motor Driver IC Datasheet*, 2021.

[4] R. Kumar and S. Rajesh, Design and Implementation of Gesture Controlled Robotic Vechicle Using Accelerometer, *International Journal of Engineering Research and Technology (IJERT)*, Vol. 8, Issue 5, 2021.

[5] P. Sharma, A. Kumar, and R. Singh, Wireless Gesture Controlled Robot Using MEMS Sensor *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 7, Issue 3, 2020.

[6] D. B. Pathakel, Development of Low Cost Metal Detector for Humanitarian Demining, *Journal of Sensors and Actuators*, Vol. 5, Issue 2, 2020.

[7] Arduino Documentation, *ESP32 and I2C Interface Programming Guide*, Arduino.cc, 2024.

[8] S. Mishra ,Design and Performance Evaluation of BFO Based Metal Detector, *International Journal of Scientific Engineering Research*, Vol. 11, Issue 8, 2020.

[9] N. Verma and R. Gupta, Wireless Control and Monitoring of Unmanned Vehicles Using ESP-NOW Protocol, *IEEE Access*, 2023.

[10] Open-Source Community, *ESP NOW Communication Examples and Libraries*, GitHub Repository, 2024.

10.Personal Profile

Name Dr. Mohan Naik R
Designation Associate Professor
Role Project Guide
Email mohannaik@sdmit.in
Contact 9901320434

He completed B.E. and M.Tech. in Electronics Communication Engineering and completed his Ph.D. from VTU Belagavi in 2022. He is Working as Associate Professor in Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire – 574 240 and affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India, with teaching experience of more than 16 years and his research area are Cryptography, Network Security, Cloud Security, IoT, Wireless Sensor Networks, Image processing. He published more than 14 research papers both in National International Journals.

Name Priyanka V S
USN 4SU23EC072
Address Mangalore–575001
Email 4su23ec072@sdmit.in
Contact 8197080633

Name Rakshitha K E
USN 4SU23EC078
Address Chithradurga– 577539
Email 4su23ec078@sdmit.in
Contact 8792682030

Name Spandana K S
USN 4SU23EC106
Address Davanagere – 577002
Email 4su23ec106@sdmit.in
Contact 9380122897

Name Tulasi C B
USN 4SU23EC114
Address Chitradurga– 577539
Email 4su23ec114@sdmit.in
Contact 8296704066