Hand Gesture Controlled Robotic Car Using ESP32

Sadman Sakib
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
sakibsadman035@gmail.com

Jannatul Adan Adreeta
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
jannatuladreeta@gmail.com

Md. Toha
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
asfakulislam16@gmail.com

Md. Fahim Muhtashim
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
23-52500-2@student.aiub.edu

***Abstract*—The primary aim of this project is to design and implement a real-time, intuitive control system for a robotic car using hand gestures. Traditional control methods, such as joysticks or remotes, are replaced with a wearable gesture-recognition transmitter utilizing the wireless capabilities of the ESP32 microcontroller. The system integrates an MPU6050 accelerometer and gyroscope to accurately capture and process hand-tilt gestures. A low-latency wireless communication link is established using the ESP-NOW protocol, chosen for its speed and efficiency over standard Wi-Fi. The robotic chassis features a receiver ESP32 and a motor driver to translate these gestures into precise vehicle movements. This project not only validates the responsiveness of gesture based control but also introduces a hybrid control mechanism featuring a web based backup interface, ensuring system reliability and redundancy for assistive technology applications.**

***Index Terms*—ESP32, MPU6050, ESP-NOW, Hand Gesture Control, Robotics, Embedded Systems.**

I. INTRODUCTION

A. Background of Study and Motivation

The development of intuitive Human Machine Interaction (HMI) has increasingly shifted focus from traditional remote controls toward more natural interfaces, such as hand gestures. While vision based gesture recognition is an option, it often suffers from computational overhead [1]. An alternative for wearable applications is the use of Inertial Measurement Units (IMUs), such as the MPU6050, which effectively translates physical hand tilts into directional commands [2]. This project is motivated by the need for low-latency, connectionless control systems that can be applied to assistive technology, entertainment and industrial usage.

B. Project Objectives

The specific objectives of this project are as follows:

- To design and build a wearable transmitter module integrating an MPU6050 accelerometer and gyroscope with an ESP32 to accurately capture hand-tilt gestures.
- To develop a robotic car chassis incorporating a second ESP32 as the receiver and a motor driver for precise vehicle movement control.
- To establish a low-latency wireless communication link between the transmitter and receiver modules, preferably using the ESP-NOW protocol for its speed and efficiency [3].
- To design and implement a secondary web based control interface using the Wi-Fi Access Point Mode of the ESP32, providing a fail-safe mechanism for redundancy and reliability.
- To write and implement embedded software that translates specific gestures into corresponding robotic car movement commands.
- To test and validate the system's responsiveness, accuracy, and reliability in controlling the robotic car, ensuring a user-friendly and effective interaction [2].

II. LITERATURE REVIEW

The development of intuitive Human Machine Interaction (HMI) has driven research away from traditional remote controls toward more natural interfaces. Shinde et al. explored gesture controlled mecanum wheel cars using ESP32 and MPU6050, highlighting the effectiveness of these components for omnidirectional movement [3]. Similarly, Udugampala et al. focused on the development of accelerometer based data acquisition systems, validating the precision of IMUs in capturing hand gestures [2].

In the context of IoT robotics, Bhimashankar et al. demonstrated IoT based hand gesture control robots, though such systems often rely on Wi-Fi infrastructure [1]. In contrast, the ESP-NOW protocol used in this project offers a distinct advantage by bypassing central routers or TCP/IP stacks for lower latency [3].

Applications extend beyond just movement; Al-Ali et al. developed a smart wheelchair control system using hand gestures, emphasizing the potential for assistive technology for individuals with motor disabilities [4]. Furthermore, Mydin et al. reviewed the effectiveness of such robotics projects in STEM education, improving technological literacy and engagement [5].

III. Methodology and Modeling

A. Introduction

The system architecture consists of two main units: a wearable Transmitter Unit and a Receiver Unit mounted on the robotic car. The system relies on the interpretation of hand gestures (tilt) to generate PWM signals for DC motors, with a secondary web interface for redundancy.

B. Working Principle of the Proposed Project

The working principle involves three stages: data acquisition, transmission, and execution. The MPU6050 sensor reads the acceleration and gyroscope data from the user's hand gestures. This data is processed by the transmitter ESP32 to determine the orientation (forward, backward, left, right). The command is sent wirelessly via ESP-NOW to the receiver ESP32. The receiver decodes the signal and drives the L298N motor driver to actuate the DC motors. Simultaneously, the receiver acts as a Wi-Fi Access Point, allowing a smartphone to connect and send control commands via a web browser if the controller is unavailable. Additionally, for safety the transmitter unit has a latching push switch for locking or unlocking the controller, such that the conmmands could be paused momentarily. The logic for gesture recognition was analyzed by defining threshold angles for the MPU6050. For example, a forward tilt exceeding 15 degrees triggers the "Forward" command. This threshold based logic ensures that minor, unintentional hand movements do not trigger the motors, preventing erratic behavior.

C. Process of Work

The workflow began with project planning and component procurement, followed by circuit design and simulation. The software was developed to implement the hybrid `Wi-Fi_AP_STA` mode, enabling both ESP-NOW and standard Wi-Fi simultaneously. Finally, the hardware was assembled, and the system underwent testing and debugging to ensure accurate mapping of gestures and web commands to car movements.

D. Description of the Components

1) **ESP-32 Microcontroller**
The ESP32 is a series of low-cost, low-power system on a chip (SoC) microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. Developed by Espressif Systems, it utilizes a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations. It is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features robust GPIOs, hardware accelerators for encryption, capacitive touch sensors, and ADC/DAC capabilities.
In this project, the ESP32 is used for both the wearable Transmitter and the robotic Receiver.
   - **Transmitter:** Interacts with the MPU6050 sensor via I2C, processes gesture data, and sends commands wirelessly.

Fig. 1. ESP32 Development Board Module

   - **Receiver:** Operates in a hybrid Wi-Fi mode (`WIFI_AP_STA`), simultaneously listening for ESP-NOW commands from the glove and hosting a Web Server for smartphone backup control. It also generates the PWM signals necessary for motor speed control.

2) **MPU6050 MotionTracking Sensor**
The MPU6050 is a 6-axis MotionTracking device comprised of a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die, along with an onboard Digital Motion Processor (DMP). It utilizes Micro-Electro-Mechanical Systems (MEMS) technology. The internal accelerometer measures the proper acceleration (rate of change of velocity) along the X, Y, and Z axes, allowing for the detection of gravity vector and hand tilt. The internal gyroscope measures the angular velocity (rate of rotation) about these same axes. The data is typically communicated to a microcontroller via the I^2C protocol.

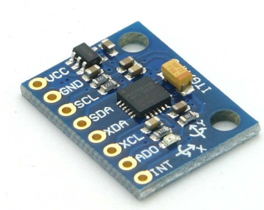

Fig. 2. MPU6050 Gyroscope and Accelerometer Module

The MPU6050 is mounted on the glove. When the user tilts their hand, the accelerometer detects the change in gravity relative to its axes. The ESP32 reads this raw data and translates it into directional commands based on specific angle thresholds defined in the code.

3) **L298N Dual H-Bridge Motor Driver**
The L298N is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Specifically, it utilizes two H-Bridge circuits, which allow voltage to be applied across a load in either direction, enabling the reversal of DC motors. It is often mounted on a breakout board containing logic supply

regulation, protective diodes, and terminal blocks for easy wiring.

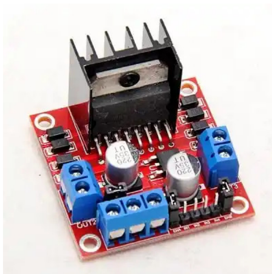

Fig. 3. L298N Motor Driver Breakout Board

This driver is essential because the ESP32 cannot supply the high current (up to 2A per motor) required by the DC motors. The Receiver ESP32 sends logic signals to the L298N to control direction and provides a PWM (Pulse Width Modulation) signal to enable pins to regulate the motor speed (via voltage averaging).

4) **DC Geared Motors**
A DC motor is an electric motor that runs on direct current (DC) electricity. To be useful for robotic movement, a standard DC motor is often coupled with a gearbox. The motor provides high-speed, low-torque rotation; the gearbox reduces the rotational speed while significantly increasing the torque (turning force). This allows the robot to carry the weight of the chassis, batteries, and electronics, and provide sufficient traction.

Fig. 4. Standard Yellow DC Geared Motor used in DIY Robotics

The car utilizes two of these motors (typically 3-12V rating) in a two-wheel drive configuration. They receive modulated power from the L298N driver to move the robotic chassis based on the commands received from the gesture glove or web interface.

5) **Power Supply and Batteries**
Robotic systems require reliable power sources. While logic circuits like the ESP32 require regulated low voltage (3.3V or 5V) and draw low current, the motors require higher voltage (7V-12V) and draw significantly higher current, which is prone to noise. DIY robots frequently utilize Lithium-Ion (Li-ion) rechargeable batteries (such as 14500 cells) due to their high energy density and ability to provide sufficient current for DC motors.

Fig. 5. 14500 Li-ion battery cell

The robotic car is powered by three 3.7V 18650 Li-ion batteries in series, providing approximately 11V to the motors via the L298N driver. An efficient voltage regulator (or the L298N's onboard 5V regulator) steps this voltage down to 5V to power the Receiver ESP32. A separate Li-ion cell was used to power the wearable Transmitter glove to keep it portable.

6) **Robotic Car Chassis (Wooden)**
The chassis serves as the structural backbone of the robot, providing a rigid platform to mount all electronic and mechanical components. In this project, a laser-cut wooden chassis is utilized due to its ease of customization, lightweight properties, and cost-effectiveness. It features pre-drilled mounting holes for the DC motors ensuring a compact and organized layout for the differential drive system.

Fig. 6. Laser-cut Wooden Robot Chassis

IV. Simulation and Experimental Setup

A. Simulation

The Receiver Unit was simulated on Proteus to test the viability of the of the project, also the Transmitter Unit was simulated on Wokwi to test its outputs. Here, the figure 7 and 8 shows the simulation and the circuit schematics of the project. Note that the pins connected in the simulation may not exactly match with the pins actually used in the hardware implementation due to slight variation of the ESP-32 modules. The program code contains the exact pins used on the hardware implementation.

B. Experimental Setup

The experimental setup includes The Receiver Unit and the Transmitter Unit. The Receiver Unit, which is simply an ESP-

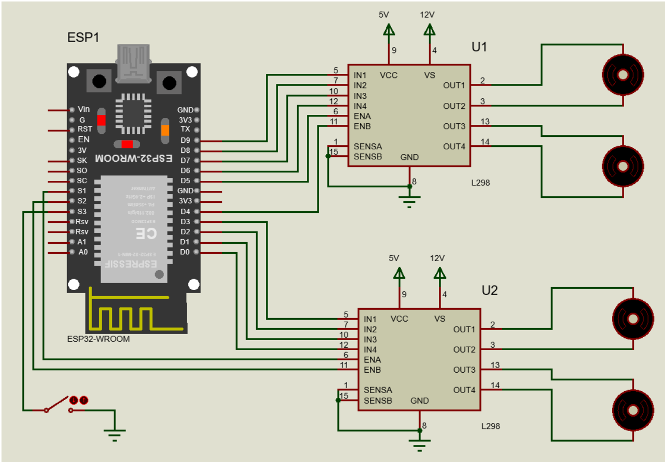

Fig. 7. Circuit schematics for the Receiver Unit

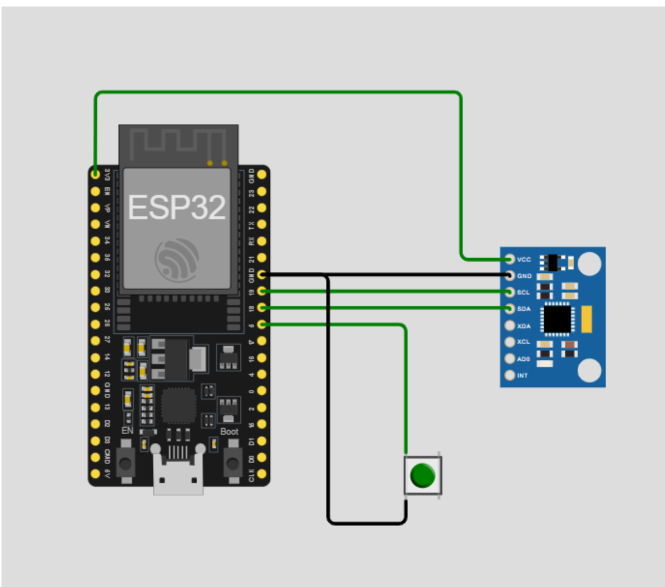

Fig. 8. Circuit schematics for the Transmitter Unit

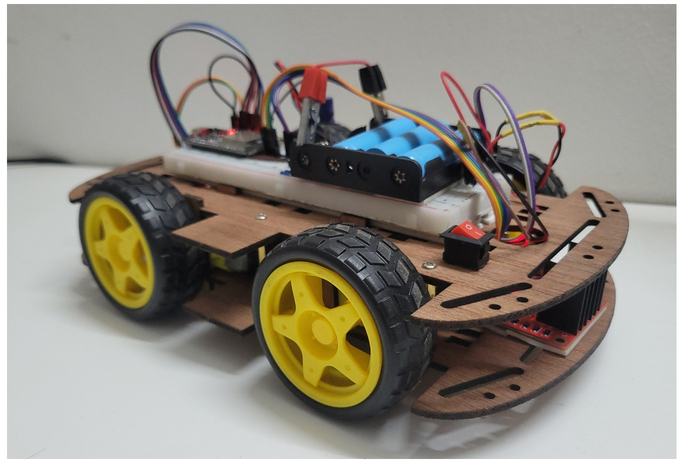

Fig. 9. Side view of the hardware implementation of the car

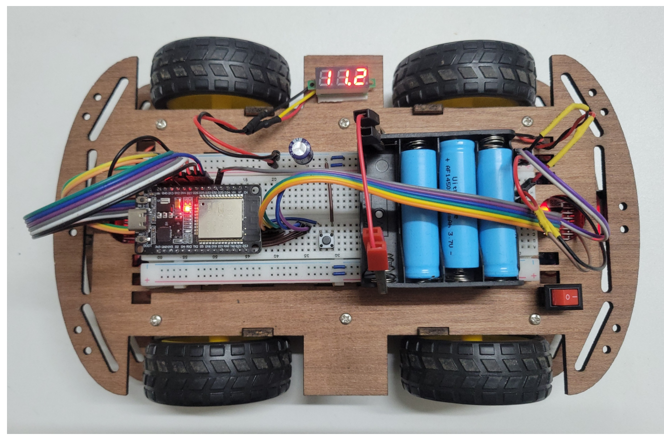

Fig. 10. Top view of the hardware implementation of the car

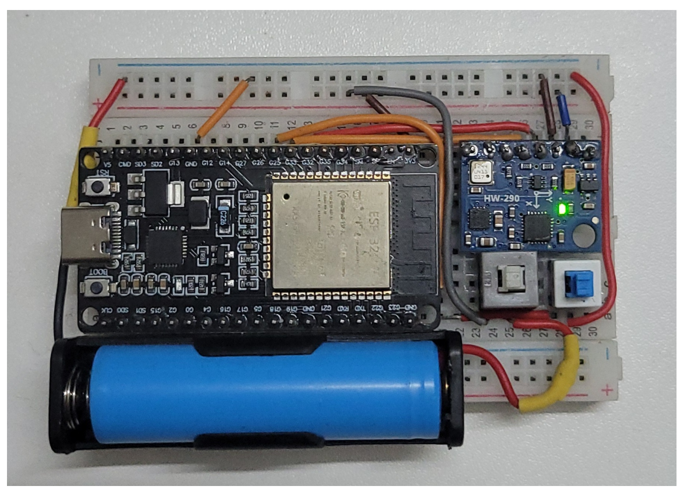

Fig. 11. The Transmitter Unit (Controller) hardware implementation

32, is mounted on a 4-wheel drive chassis along with the two L298N motor driver and a lithium ion battery pack, which contains three lithium ion battery cells. A simple ICL7107-based voltmeter was also mounted to monitor the voltage of the battery pack, which acts like a pseudo charge level indicator. The figure 9 and figure 10 shows the hardware implementation of The Receiver Unit.

As for The Transmitter Unit, which consists of another ESP-32 along with the MPU6050 and another lithium battery pack of one cell. It also was fitted with a lock/unlock switch for pausing the transmission of the tilt commands. All the components was fitted on a small breadboard for compactness.

The ESP-NOW protocol was configured to pair the MAC addresses of the two ESP-32 units, establishing a dedicated communication channel on Wi-Fi Channel 1. Additionally, the receiver ESP-32 was configured to start a web server on its own Wi-Fi access point for controlling the car from the web; the figure 12 shows the screenshot of the web-based controller UI.

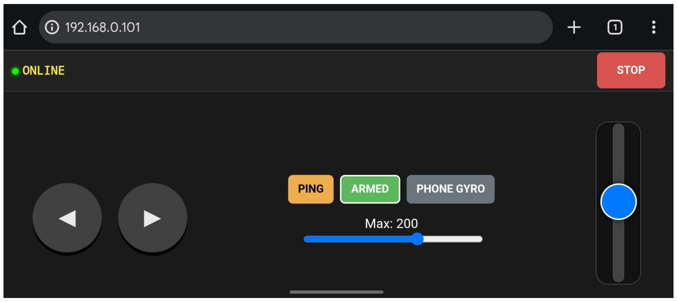

Fig. 12. Screenshot of the web-based controller UI

V. Results and Discussions

A. Experimental Results

During testing, the robotic car successfully responded to four distinct hand gestures:

1) **Forward Tilt:** Car moves forward.
2) **Backward Tilt:** Car moves backward.
3) **Left/Right Tilt:** Car turns in the respective direction.
4) **Flat/Neutral:** Car stops.

The latency was observed to be minimal due to the connectionless nature of ESP-NOW.

B. Web Interface & Hybrid Mode Tests

To enhance system reliability, a secondary web interface was developed. The ESP32 hosts a lightweight web server on `192.168.4.1`, allowing control via any smartphone browser. Testing confirmed that the system successfully prioritizes controller input (ESP-NOW) while maintaining the Web Access Point in the background. A push button was added on top of the car to switch between the two modes, If pressed for 1 second the reciever ESP32 takes conmand from the web interface and vice versa.

C. Cost Analysis

The project was designed to be cost-effective. A breakdown of the approximate component costs is provided in the table I.

TABLE I
Cost of Components

Component	Price (BDT)	Quantity	Subtotal (BDT)
ESP32 ESP-32S 30P NodeMCU	630	2	1,260
4WD Robotics Wooden Chassis With Motors & Wheels	990	1	990
L298N H-Bridge Dual Motor Driver	168	2	336
MPU6050	690	1	690
14500 Rechargeable Lithium Battery 3.7V	80	4	320
ICL7107-based voltmeter	120	1	120
7x7mm latching Push	10	2	20
Jumper Wires	130	1	130
Battery Holder	18	1	18
Breadboard Full Size	150	1	150
Breadboard Half Size	90	1	90
Total			**4,124**

D. Limitations in the Project

While functional, the system has limitations. The range of ESP-NOW is limited compared to long-range RF modules. Additionally, the system currently lacks obstacle avoidance sensors, meaning the user must maintain visual line-of-sight to prevent collisions.

VI. Conclusion and Future Endeavors

This project successfully demonstrated the design and implementation of a hand gesture controlled robotic car using the ESP32 and MPU6050. By leveraging a hybrid communication architecture, the system achieved low-latency gesture control via ESP-NOW while providing a fail safe web interface for manual override. The intuitive, low effort control system could empower users to operate electric wheelchairs or robotic assistance arms, promoting greater independence [4]. Additionally, it creates a maker culture, which is a proven method for improving engagement and technological literacy in STEM education [5].

Future endeavors will focus on integrating obstacle avoidance sensors to enhance safety and autonomy. Furthermore, the principles developed here can be scaled for industrial applications, such as remotely controlling machinery in hazardous environments.

References

[1] M. Bhimashankar, N. Nandan, M. Azhar, and M. Muzammil, “Iot based hand gesture control robot,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 12, no. 09, pp. 1–5, 2023.

[2] P. P. K. Udugampala, S. G. P. L. Amarathunga, A. S. D. G. Senarathne, M. D. D. J. M. B. Dissanayaka, and A. P. P. D. B. P. Adhikari, “Development of an accelerometer-based data acquisition system for hand gesture recognition,” in *2024 8th International Conference on Devices, Circuits and Systems (ICDCS)*, pp. 1–5, 2024.

[3] P. G. N. Shinde, M. P. G. Shinde, M. A. S. Gaikwad, and M. A. S. Pawar, “Gesture controlled mecanum wheel car using esp32 and mpu6050,” *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 11, no. 6, pp. h742–h748, 2024.

[4] A. R. Al-Ali, M. Al-Naimi, A. M. Al-Sadik, and M. Al-Ali, “A smart wheelchair control system using hand gestures and iot,” *IEEE Access*, vol. 9, pp. 76366–76378, 2021.

[5] M. S. M. Mydin, U. F. B. A. Manaf, Y. B. M. Yusof, and N. B. Ahmad, “The effectiveness of robotics in stem education: A review,” *International Journal of Academic Research in Business and Social Sciences*, vol. 10, no. 11, pp. 436–447, 2020.

Appendix

The program and other related materials for the project can be found in the following Github repository. https://github.com/r2sakib/esp32-gesture-rc-car