GESTURE CONTROLLED ROBOT USING ESPNOW

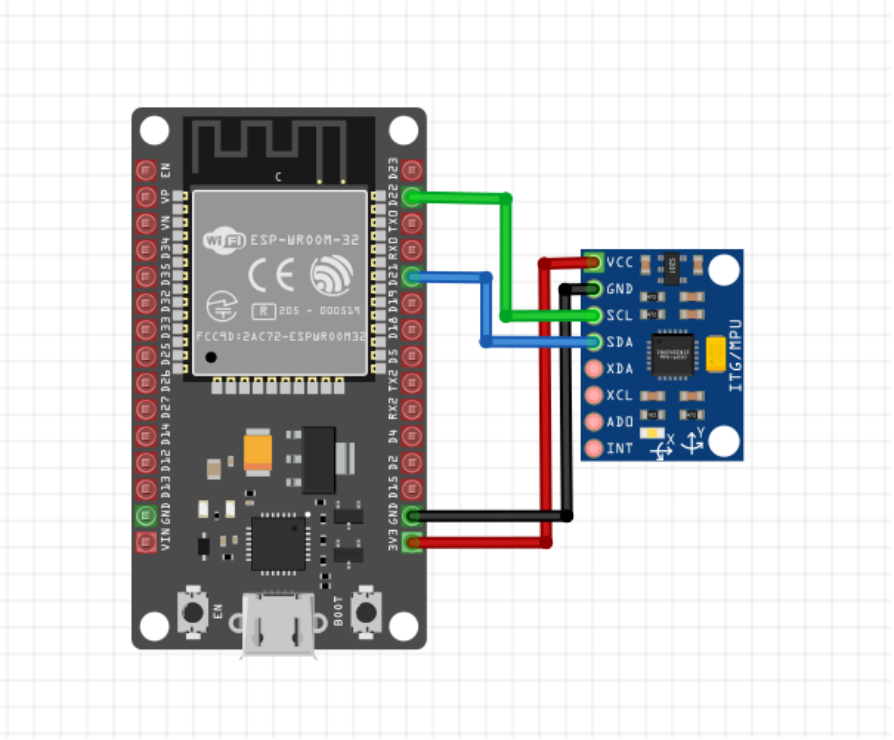
Objective: The objective is to build a gesture-controlled robot using ESP-NOW communication, where the car moves in different directions based on control signals received from the transmitter, allowing wireless and responsive control.

Components: ESP32-2, L298N Motor Driver, 4-wheel robotic chassis, 18650 Li-ion battery - 2, MPU6050 , Half Breadboard, 5V battery supply, F-F, M-F,M-M Jumper wires.

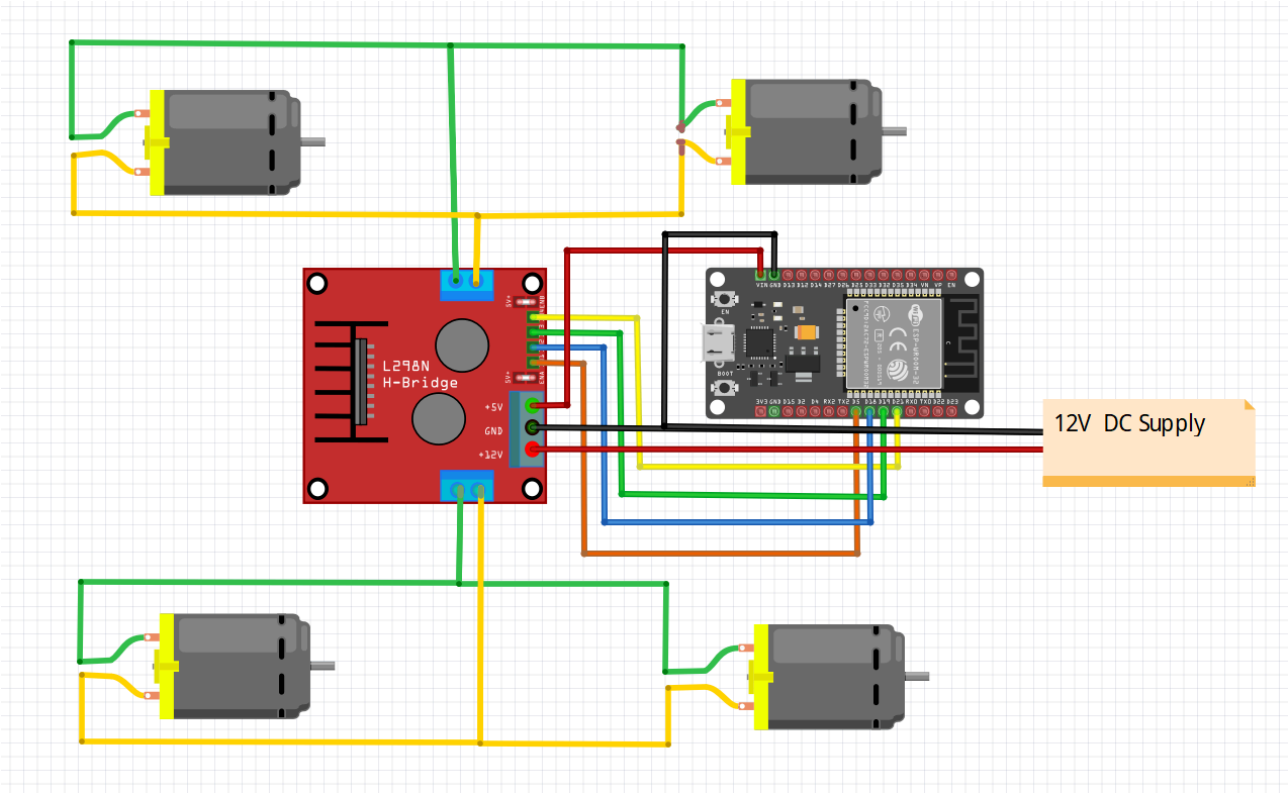
Software Requirements: Arduino IDE.

Wiring Diagram:

Transmitter:



Receiver:



Procedure:

- Firstly upload the mac address code in ESP32 board of Receiver and retrive the mac address.
- Then place the receiver mac address in transmitter code and upload into the transmitter ESP32 and also upload receiver code in receiver ESP32.
- Make sure to connect Transmitter and Receiver ESP32 to the WIFI STA.
- After connecting the wires according to the diagram. Powerup the setup
- Then according to the hand movements we give through the transmitter the car moves its directions.

Troubleshooting points:

• Make sure the ESP-NOW communication between the transmitter and receiver is properly set up. Both devices should be initialized with correct MAC addresses.
• Verify that the motor control pins (IN1, IN2, and En) are correctly connected to the corresponding GPIO pins on the ESP32.
• Ensure the PWM channels are properly configured and attached to the correct motor enable pins for speed control.
• Check the motor driver connections to ensure no loose wires, as improper connections can lead to inconsistent motor performance.
• Ensure the motor speeds and directions are correctly assigned in the code, particularly during diagonal movements and turns.
• Monitor signal timeout: If no signal is received for a long period, the car should stop. Ensure this timeout is functioning as expected.
• Make sure WiFi mode is set to STA (Station Mode) for ESP-NOW to work correctly, as using another mode can disrupt communication.
• Check for power supply issues like voltage drops, as they can affect motor performance and ESP32 behavior. Fully charge batteries before use.
• Verify that the received data values (x, y, z) from the transmitter are within expected ranges for the car to move in the right direction.

Output :

- The car is controlled by hand gestures and moves in response to signals received from the transmitter.
- Based on the tilt direction, the car moves forward, backward, left, or right.
- The transmitter and receiver need to stay stable for 1 minute to establish a proper connection.
- Tilting the transmitter forward makes the car move forward, while tilting it backward makes the car move in reverse.
- Tilting the transmitter sideways causes the car to turn left or right accordingly.

Applications :

- **Assistive Technology**: Enable people with physical disabilities to control vehicles or devices using hand gestures.

- **Remote Control**: Use hand gestures to wirelessly control toy cars, drones, or robots for entertainment or education.

- **Gaming**: Create immersive driving experiences in virtual reality (VR) or simulation games using gesture inputs.
- **Smart Homes**: Use gesture-controlled cars as part of smart home systems, where small vehicles deliver items or perform tasks within a home.

Future Enhancements:

Ultrasonic Sensor and Buzzer for Obstacle Detection:

- Integrate an ultrasonic sensor to detect obstacles in the car's path.
- Trigger a buzzer alarm when the car encounters an obstacle while being controlled through gestures via the transmitter.

LCD Display for Speed and Direction Monitoring:

- Add an LCD display to show real-time information on the car's speed and direction.
- Implement a buzzer alarm to trigger if the car's speed exceeds a predefined limit, ensuring safety.

Project By :

Nagiri Sai Akshaya,
BTech(IoT),
ACE Engineering College.