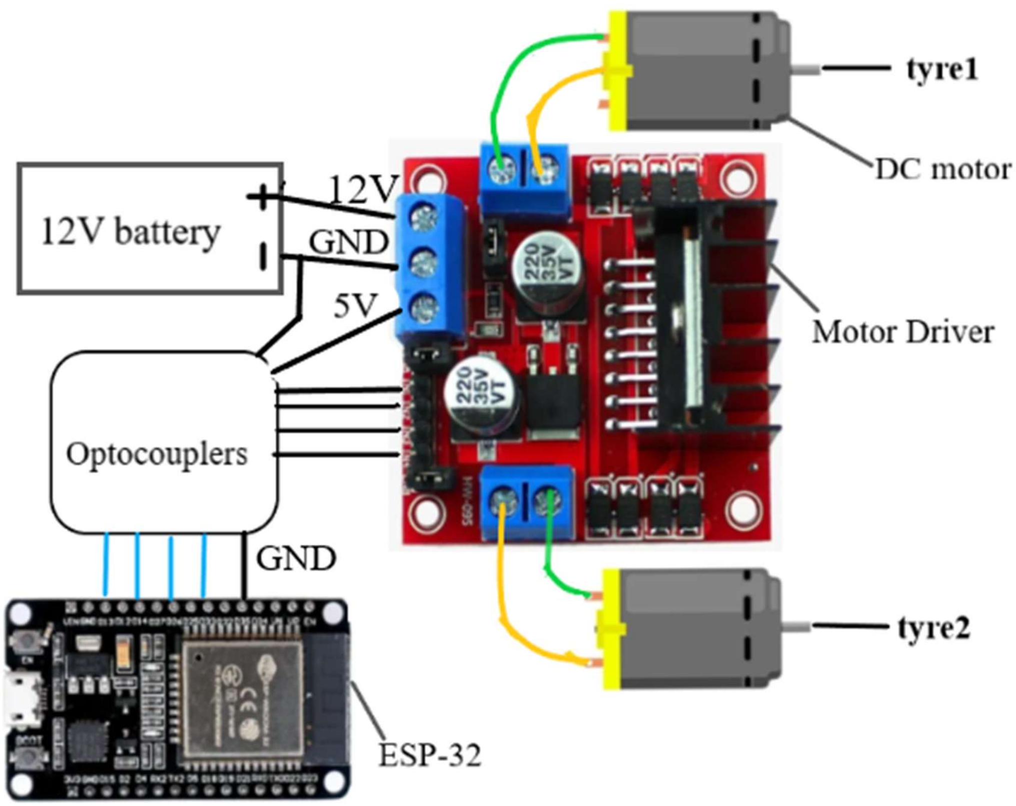
MPS CEP REPORT

Gesture Controlled Car

1. Block diagram of hardware:



2. Discussion:

Motor Driver:

• The motor driver controls the motion of the car. It is connected to two DC motors, facilitating control over both wheels.
• The driver receives signals from the ESP32 via Optocouplers, determining the direction and speed of the motors.
• Two motors are connected in parallel, ensuring synchronized and coordinated movement.

DC Motors:

• The movement of the car is driven by the DC motors, with each motor directly connected to its respective tire.
• Positive and negative terminals of each motor connect to the motor driver, allowing for precise control over the rotational direction and speed of the motors.

12V Battery:

• The primary power source for the entire system, the 12V battery ensures a stable and continuous supply of electrical energy.
• Connected to the motor driver, the battery powers the DC motors and, consequently, the movement of the car.

Optocouplers:

• Positioned between the ESP32 and the motor driver, Optocouplers act as signal intermediaries.
• Four pins (in1, in2, in3, in4) on the motor driver receive signals from the Optocouplers, representing distinct motor control commands transmitted by the ESP32.

ESP32:

• The ESP32 plays a central role in the gesture-controlled car's operation, serving as the brain of the system.
• Through wireless communication, the ESP32 sends control signals to the motor driver via Optocouplers, dictating the car's movement based on interpreted hand gestures.

Microprocessor Unit (MPU) Integration:

To enhance the gesture-controlled car's precision, an Inertial Measurement Unit (MPU) is incorporated into the system. The MPU is connected to the Tiva-C Launchpad through the I2C communication protocol.
This integration allows the Tiva-C Launchpad to receive real-time data from the MPU, capturing the hand gestures made by the user.

Stability Check and Calibration:

To ensure the reliability of the captured MPU values for a specific gesture, a stability check mechanism is implemented.
If the MPU values are found to be unstable, a calibration process is initiated to make them stable. This step is crucial for the accurate interpretation of the user's hand gestures, contributing to the precise control of the gesture-controlled car.

Motion Ranges Determination:

The system defines five distinct motion ranges, each corresponding to a specific gesture. These ranges are carefully calibrated based on the stabilized MPU values.
The precise definition of these ranges is essential for accurately translating the user's gestures into meaningful commands for the gesture-controlled car.

ESP01:

The communication between the Tiva-C Launchpad and the ESP01 module is established using UART2 interfacing. The ESP01 module acts as a server in this setup. It receives the defined motion ranges for specific gestures from the Tiva-C Launchpad through UART2.

ESP32 Client Communication:

The ESP01 module, functioning as a server, communicates with the ESP32 client over WiFi. It transmits the predefined motion ranges for each gesture to the ESP32 client. The ESP32 client, acting as the central control unit, interprets these ranges and arranges the corresponding movements of the gesture-controlled car.

Motor Control Commands:

Upon receiving the motion ranges from the ESP01 module, the ESP32 client generates specific motor control commands. These commands are then sent to the motor driver, directing the movement of the car. The integration of ESP32 into the communication loop ensures seamless coordination between the user's gestures, MPU data, and the physical motion of the gesture-controlled car.