REPORT

Project Title:Speed-Controlled Robotic Vehicle using ESP8266 **

1. Introduction:**

The Speed-Controlled Robotic Vehicle is a project aimed at designing a remote-controlled robot that can be controlled wirelessly using the ESP8266 microcontroller. The system allows users to control the speed and direction of the robot through a mobile application or a web-based interface. The ESP8266 provides Wi-Fi connectivity, enabling communication between the robot and the control interface.

2. Components Used:

- ESP8266 board: Serves as the main microcontroller and provides Wi-Fi connectivity.
- Motor driver module: Controls the speed and direction of the robot's motors.
- DC motors: Drives the wheels of the robot.
- Jumper wires: Establishes connections between the components.
- Breadboard or PCB: Provides a platform for mounting and connecting the components

3. Schematics:

The following schematic diagram illustrates the connection of the motor driver module and the DC motors to the ESP8266 board:

```
+-------------------+
| ESP8266 |
| |
| D1 (GPIO5)---|----IN1 (Motor 1)
| D2 (GPIO4)---|----IN2 (Motor 1)
| D3 (GPIO0)---|----IN1 (Motor 2)
| D4 (GPIO2)---|----IN2 (Motor 2)
+-------------------+
```

4. Procedure of Connection:

Follow the steps below to establish the connections between the components:

- Connect the motor driver module to the ESP8266 board using jumper wires.
- Connect the IN1 and IN2 pins of Motor 1 to the D1 (GPIO5) and D2 (GPIO4) pins on the ESP8266 board, respectively.
- Connect the IN1 and IN2 pins of Motor 2 to the D3 (GPIO0) and D4 (GPIO2) pins on the ESP8266 board, respectively.
- Connect the positive terminal (+) of the power supply to the VCC pin of the motor driver module.
- Connect the negative terminal (-) of the power supply to the GND pin of the motor driver module.
- Connect the motors to the motor driver module.
- Ensure that the connections are secure and there are no loose connections or short circuits.

5. Software Implementation:

implement the software part of the project, follow these steps:

- Set up the Arduino IDE and install the required libraries: ESP8266WiFi.
- Open the Arduino IDE and create a new sketch.
- Copy the provided code into the sketch.
- Modify the code to replace the placeholders with your Wi-Fi credentials (WLAN_SSID and WLAN_PASS).
- Upload the code to the ESP8266 board.
- Open the Serial Monitor to view the status and debug information.

6. System Operation:

Once the system is powered on and the code is uploaded successfully, the ESP8266 board starts executing the program. The motor driver module receives commands from the control interface (mobile application or web-based interface) via Wi-Fi. The commands control the speed and direction of the robot's motors.

The user can control the robot by sending commands to the ESP8266 board through the control interface. The commands are interpreted by the ESP8266 board, which adjusts the motor speed and direction accordingly.

7. Conclusion:

The Speed-Controlled Robotic Vehicle project demonstrates the use of ESP8266 as a Wi-Fi-enabled microcontroller for remote controlling of a robot. With this system, users can wirelessly control the speed and direction of the robot, enablingremote navigation and operation.

8. Code:

```
#include <ESP8266WiFi.h>

const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";

WiFiServer server(80);

const int motor1PWM = 5;  // GPIO5/D1 - Motor 1 PWM
const int motor1A = 4;    // GPIO4/D2 - Motor 1 direction A
const int motor1B = 0;    // GPIO0/D3 - Motor 1 direction B

const int motor2PWM = 2;  // GPIO2/D4 - Motor 2 PWM
const int motor2A = 14;   // GPIO14/D5 - Motor 2 direction A
const int motor2B = 12;   // GPIO12/D6 - Motor 2 direction B

void setup() {
  Serial.begin(115200);
  delay(10);

  // Connect to Wi-Fi
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.print("Use this URL to control the robot: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

  // Set motor pins as output
  pinMode(motor1PWM, OUTPUT);
  pinMode(motor1A, OUTPUT);
  pinMode(motor1B, OUTPUT);
  pinMode(motor2PWM, OUTPUT);
  pinMode(motor2A, OUTPUT);
  pinMode(motor2B, OUTPUT);

  // Stop the motors initially
  stopMotors();
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait for data from the client
  while (!client.available()) {
    delay(1);
  }

  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  // Process the request
  if (request.indexOf("/forward") != -1) {
    moveForward();
  } else if (request.indexOf("/backward") != -1) {
    moveBackward();
  } else if (request.indexOf("/stop") != -1) {
    stopMotors();
  } else if (request.indexOf("/speed") != -1) {
    int speedValue = getValue(request, "speed=");
    setSpeed(speedValue);
  }

  // Send HTTP response
  sendHttpResponse(client);

  // Delay to allow the client to receive the response
  delay(10);
  client.stop();
}

void moveForward() {
  digitalWrite(motor1A, HIGH);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor2A, HIGH);
  digitalWrite(motor2B, LOW);
}

void moveBackward() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, HIGH);
  digitalWrite(motor2A, LOW);
  digitalWrite(motor2B, HIGH);
}

void stopMotors() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor2A, LOW);
  digitalWrite(motor2B, LOW);
```

```
}

void setSpeed(int speed) {
  analogWrite(motor1PWM, speed);
  analogWrite(motor2PWM, speed);
}

void sendHttpResponse(WiFiClient client) {
  client.println("HTTP/1.1 200 OK");
  client.println("Content-type:text/html");
  client.println();
  client.println("<html>");
  client.println("<head><title>Speed-Controlled Robotic Vehicle</title></head>");
  client.println("<body>");
  client.println("<h1>Control the Robot</h1>");
  client.println("<p><a href=\"/forward\">Move Forward</a></p>");
  client.println("<p><a href=\"/backward\">Move Backward</a></p>");
  client.println("<p><a href=\"/stop\">Stop</a></p>");
  client.println("<form action=\"/speed\">");
  client.println("Speed (0-255): <input type=\"number\" name=\"speed\" min=\"0\" max=\"255\">");
  client.println("<input type=\"submit\" value=\"Set Speed\">");
  client.println("</form>");
  client.println("</body>");
  client.println("</html>");
}

int getValue(String data, String key) {
  int value = 0;
  int start = data.indexOf(key) + key.length();
  int end = data.indexOf("&", start);
  if (end == -1) {
    end = data.length();
  }
  String valueString = data.substring(start, end);
  value = valueString.toInt();
  return value;
}
```