

Wi-Fi Controlled Moon Rover Using Arduino & ESP32

Project Overview

This **Wi-Fi Moon Rover** is designed to explore rough terrains, similar to a real lunar rover. It uses **ESP32 for wireless communication**, **Arduino for motor control**, and multiple sensors for obstacle avoidance and environmental data collection. The rover can be controlled remotely via a **web dashboard** and can stream live video using an **ESP32-CAM module**.

Objectives

- ✓ **Remotely control the rover** over Wi-Fi.
- ✓ **Navigate rough terrain** using motorized wheels.
- ✓ **Stream live video** from the rover's camera.
- ✓ **Measure environmental conditions** (temperature, radiation, pressure).
- ✓ **Avoid obstacles** using ultrasonic sensors.

Components Required

1. **ESP32 (NodeMCU)** – Wi-Fi connectivity & video streaming.
2. **Arduino Uno** – Controls motors & sensors.
3. **L298N Motor Driver** – Drives the rover's wheels.
4. **DC Motors (4x)** – Enables movement.
5. **ESP32-CAM** – Provides live video streaming.
6. **Ultrasonic Sensors (HC-SR04)** – Detects obstacles.
7. **DHT11/DHT22 Sensor** – Measures temperature & humidity.
8. **BMP280 Sensor** – Measures atmospheric pressure.
9. **Solar Panel (Optional)** – Provides sustainable power.
10. **Li-Ion Battery (12V)** – Powers the rover.
11. **Jumper Wires & Chassis** – For building the rover body.

How the System Works

1. The **ESP32 hosts a web dashboard** for rover control.
2. The **user sends movement commands** (forward, backward, left, right).
3. The **Arduino controls the motors** via the L298N driver.
4. The **ESP32-CAM streams live video** to the web interface.
5. The **Ultrasonic sensors detect obstacles** and prevent collisions.

6. The **DHT11 & BMP280** sensors collect environmental data (temperature, pressure).
-

Circuit Diagram & Connections

L298N Motor Driver to Arduino

L298N Pin Arduino Uno

IN1	D6
IN2	D7
IN3	D8
IN4	D9
ENA	5V (PWM)
ENB	5V (PWM)
GND	GND
VCC	12V Battery

ESP32 to Ultrasonic Sensor

HC-SR04 ESP32

VCC	3.3V
GND	GND
TRIG	D5
ECHO	D18

DHT11 to ESP32

DHT11 ESP32

VCC	3.3V
GND	GND
DATA	D4

Arduino Code for Motor Control

This code moves the rover based on Wi-Fi commands from ESP32.

```
#define IN1 6
#define IN2 7
#define IN3 8
```

```

#define IN4 9

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()) {
    char command = Serial.read();
    if (command == 'F') { // Move Forward
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, HIGH);
      digitalWrite(IN4, LOW);
    }
    else if (command == 'B') { // Move Backward
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    }
    else if (command == 'L') { // Turn Left
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, HIGH);
      digitalWrite(IN4, LOW);
    }
    else if (command == 'R') { // Turn Right
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    }
    else if (command == 'S') { // Stop
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, LOW);
    }
  }
}

```

ESP32 Code for Web Control

This ESP32 code **hosts a web interface** for controlling the rover wirelessly.

```

#include <WiFi.h>

const char* ssid = "Your_WiFi_Name";
const char* password = "Your_WiFi_Password";

```

```
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected!");
  server.begin();
}

void loop() {
  WiFiClient client = server.available();
  if (client) {
    String command = client.readStringUntil('\n');
    Serial.println(command);
  }
}
```

How to Use the Wi-Fi Moon Rover

1. **Upload the motor control code to Arduino Uno.**
 2. **Upload the ESP32 web server code.**
 3. **Connect to Wi-Fi and get the IP address from the Serial Monitor.**
 4. **Open the web interface in a browser (e.g., <http://192.168.1.100>).**
 5. **Control the rover remotely using arrow buttons on the webpage.**
 6. **View live video streaming from ESP32-CAM.**
-

Features & Benefits

- ✓ **Remote Wi-Fi Control** – Operate the rover from anywhere.
 - ✓ **Live Video Streaming** – View terrain in real time.
 - ✓ **Obstacle Avoidance** – Uses ultrasonic sensors for safety.
 - ✓ **Environmental Data Collection** – Tracks temperature, pressure, humidity.
 - ✓ **Solar Charging Support** – For extended missions.
-

Future Enhancements

AI-based Object Detection using TensorFlow Lite.
GPS Navigation for autonomous movement.

Dust & Radiation Sensors for Mars-like conditions.
Energy-efficient Solar Power for long-term operation.



BINARY BRAINS