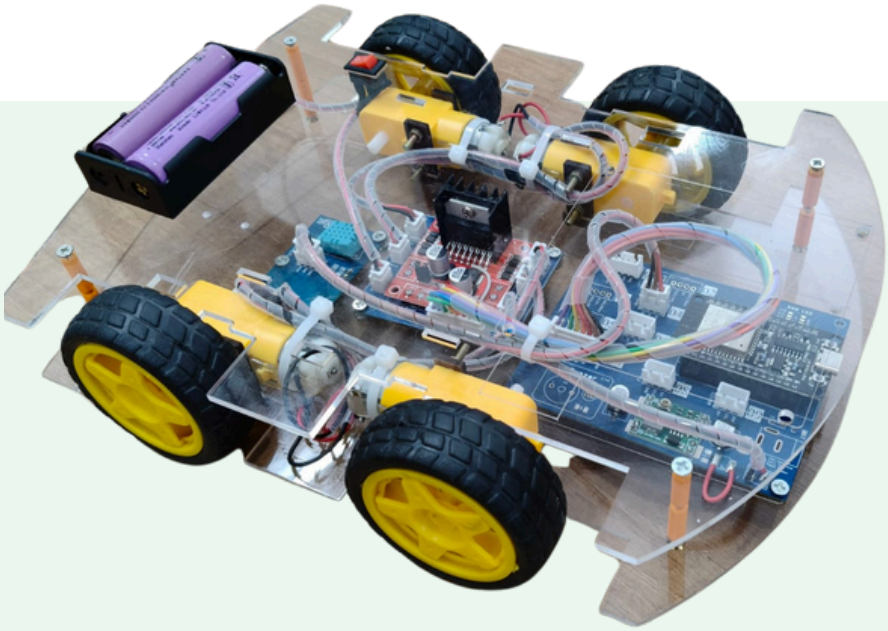
SAGEDEL
SKILL ADAPTIVE EDUCATION

MOISTURE FINDING ROVER

Autonomous Soil Moisture Mapping and Environmental Telemetry System

2026



SAGEDEL TECH LLP

Phone
9958007460

Website
www.sagedel.com

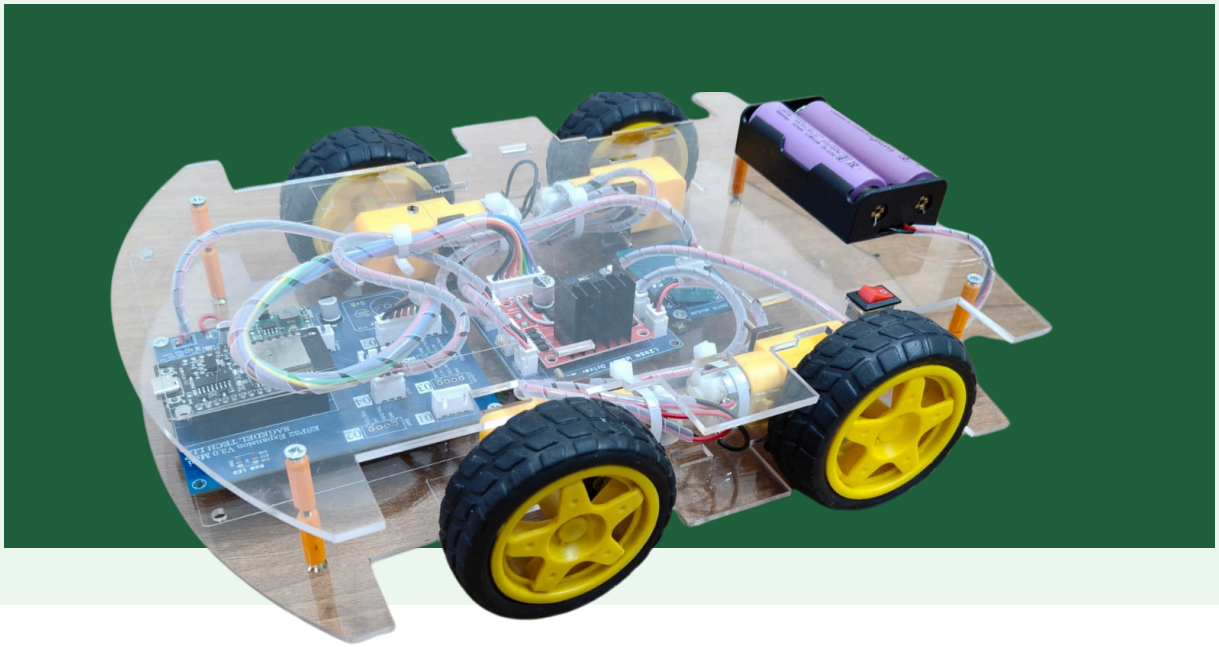
MOISTURE FINDING ROVER

Objective:

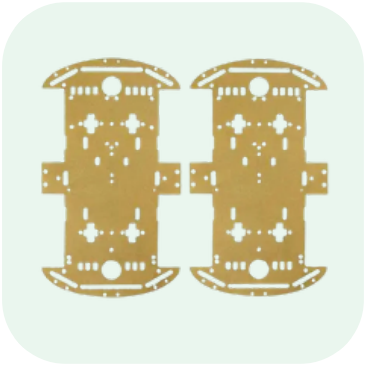
Learn how to interface soil moisture and movement sensors with a microcontroller to autonomously detect and map soil hydration levels across different field zones.

What it Does:

The microcontroller reads live moisture sensor data from the soil while the rover navigates. It processes these moisture levels and maps the hydration status in real time, pinpointing dry areas that require targeted watering.

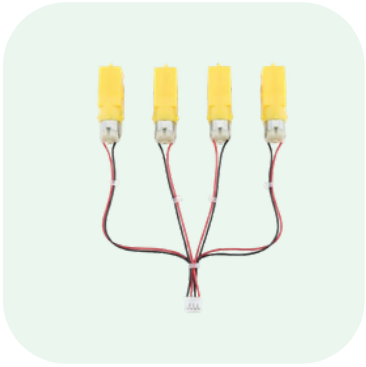


Modules & Components used:

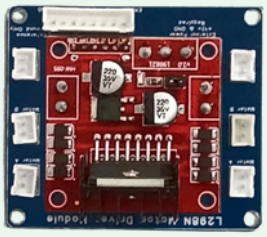


Two-Layer Acrylic Chassis

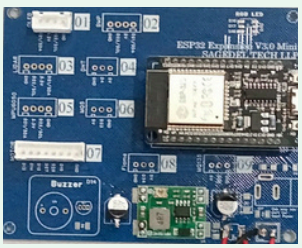
Wheels



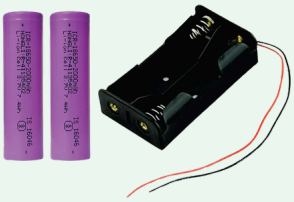
BO Motors



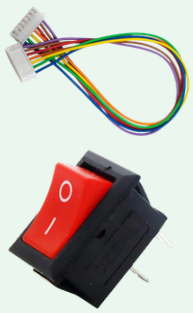
L298N Motor Driver



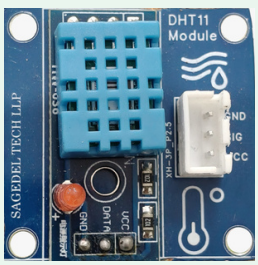
ESP32 Expansion Board



Li-ion Cells & Battery Holder



JST Cables & Switch

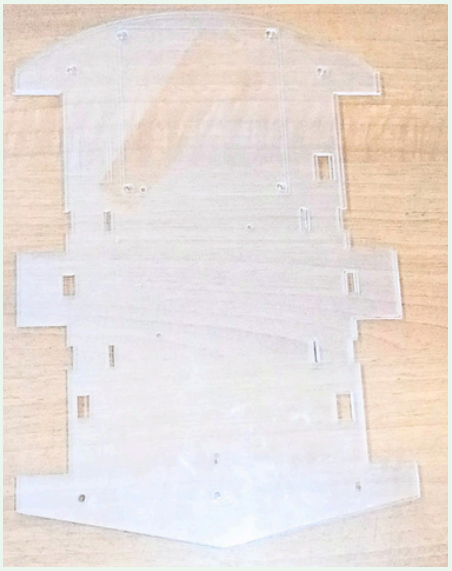


DHT11 Module

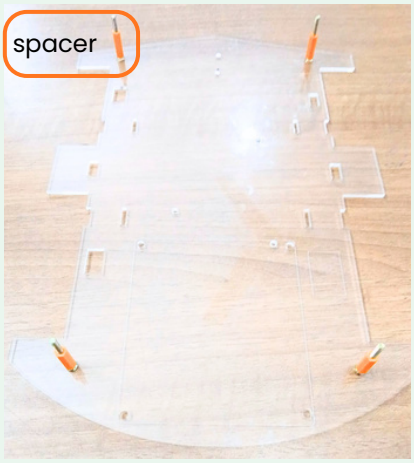
Spacers, Nuts- Bolts & Cable Ties

Step-by-Step Assembly & Connections

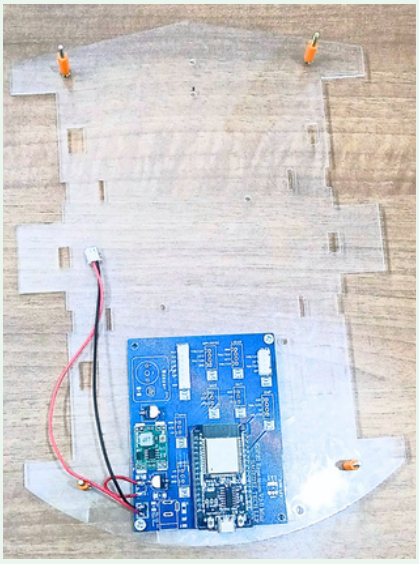
Step 1: Place the lower acrylic chassis on a flat surface and remove the protective film.
Step 2: Fix the four spacers at the corners of the lower chassis using nuts and bolts.
Step 3: Mount the ESP32 Expansion Board on the chassis using screws.



Step 1

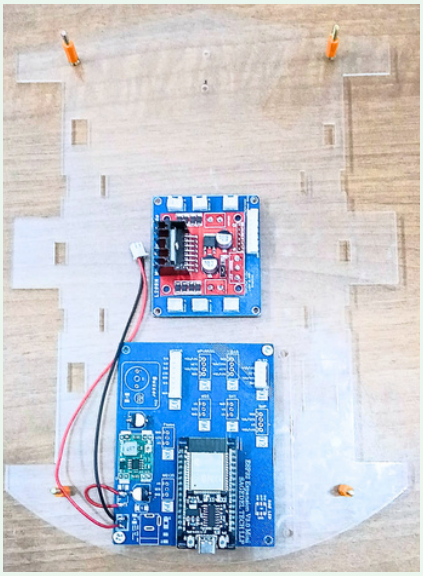


Step 2

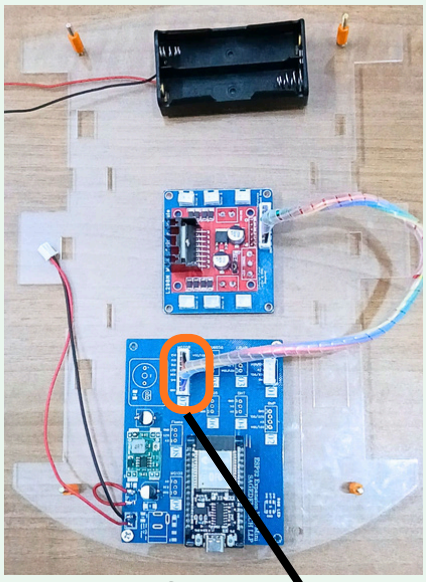


Step 3

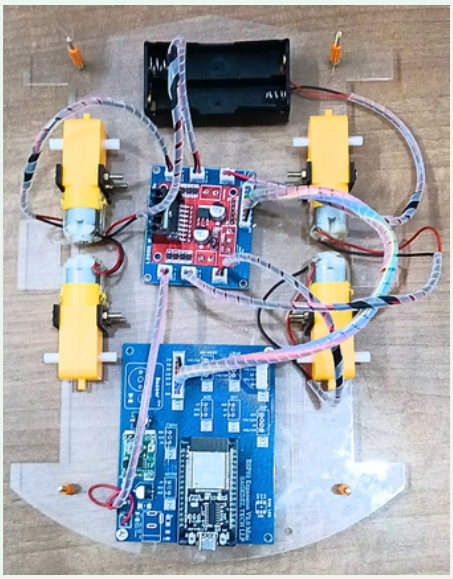
Step 4: Mount the L298N Motor Driver on the chassis using screws. Connect it to the ESP32 Expansion Board **(Connector No. 7)**.
Step 5: Mount the Li-ion Battery Holder on the chassis using screws.
Step 6: Attach the 4 BO motors to the chassis. Connect the motor wires to Motor A and Motor B on the L298N Motor Driver. **Follow the Motor Assembly Steps for all four motors.**



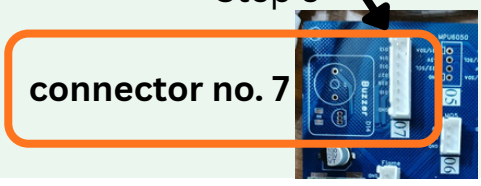
Step 4



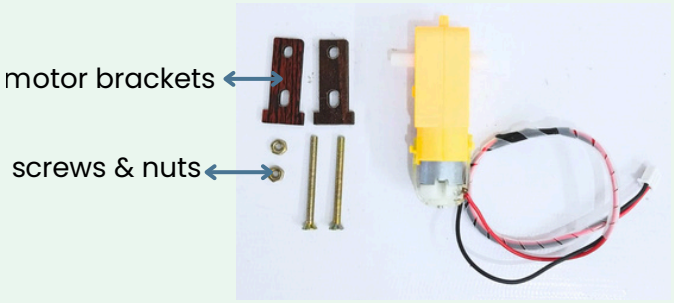
Step 5

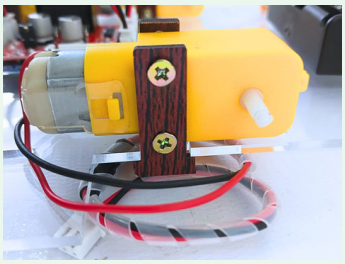


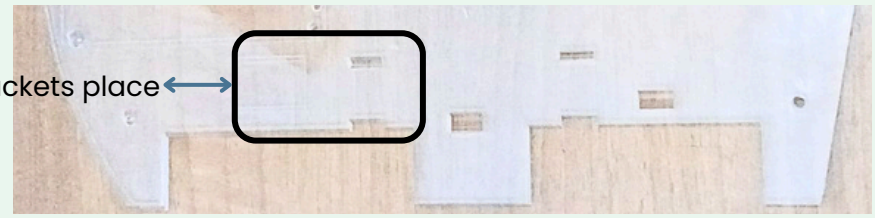
Step 6



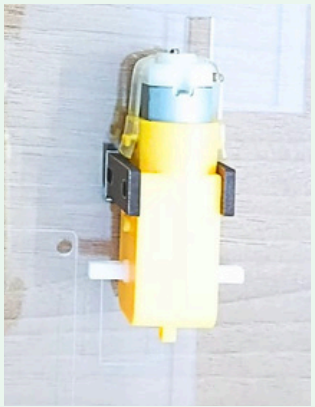
Motor Assembly



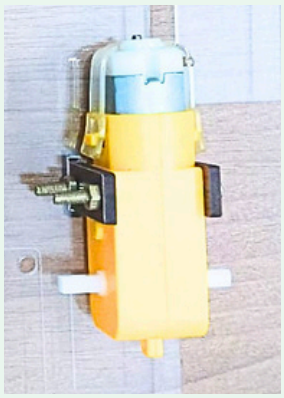




Step 1: motor brackets on the chassis.



Step 2: BO motor between the brackets.

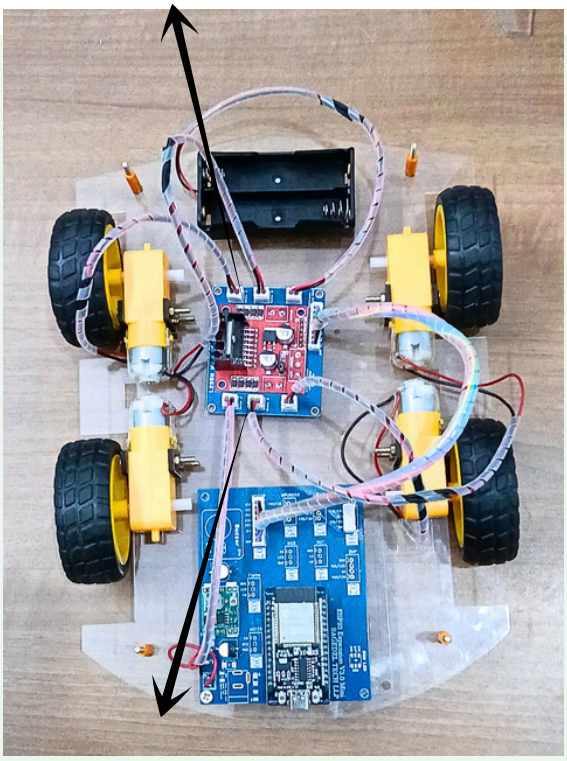


Step 3: Secure it using the screws and nuts.

Motor-to-Driver Connections :

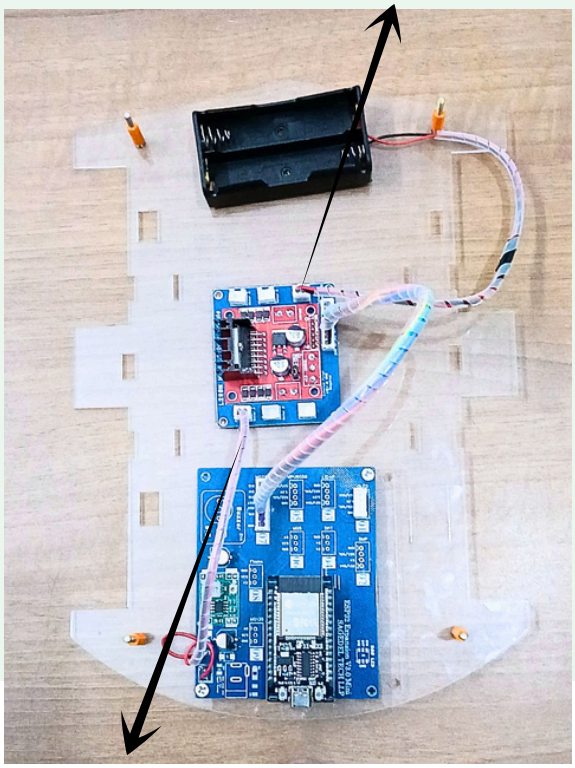
Step 4 : Connect the left-side pair of BO motors in parallel to **OUT1** and **OUT2** on the L298N motor driver.
Step 5 : Connect the right-side pair of BO motors in parallel to **OUT3** and **OUT4** on the L298N motor driver.

Connect the right motors to OUT1 & OUT2



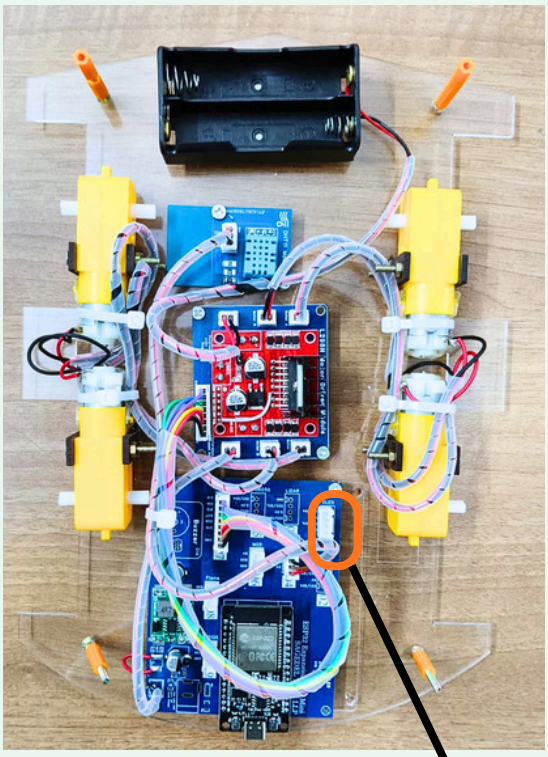
Connect the left motors to OUT3 & OUT4

Connect the battery to 12V & GND

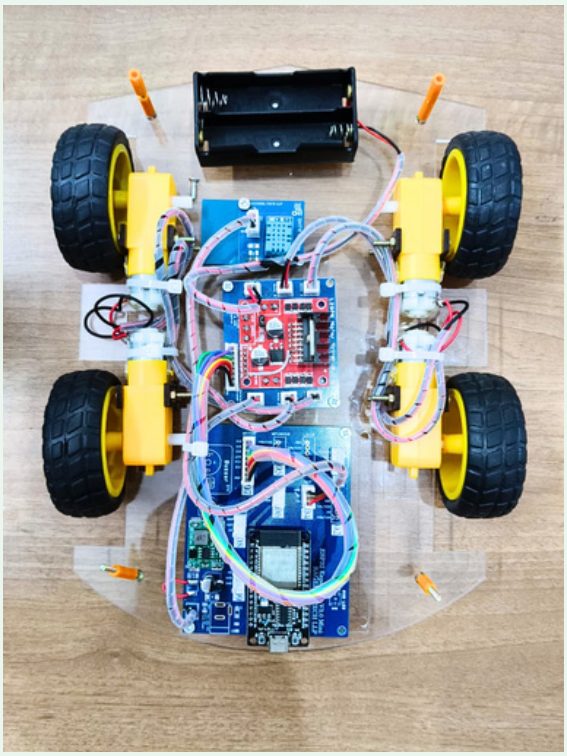


Connect the ESP power input to the reference ground.

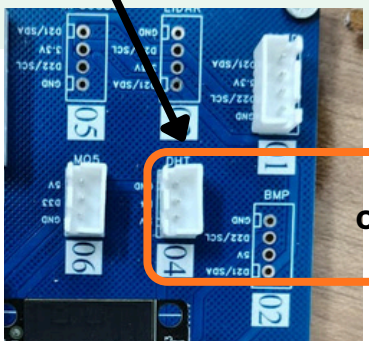
Step 7: Mount the DHT11 Sensor on the chassis using screws. Connect it to the ESP32 Expansion Board **(Connector No. 4)**.
Step 8: Attach the 4 wheels to the BO motor shafts.



Step 7

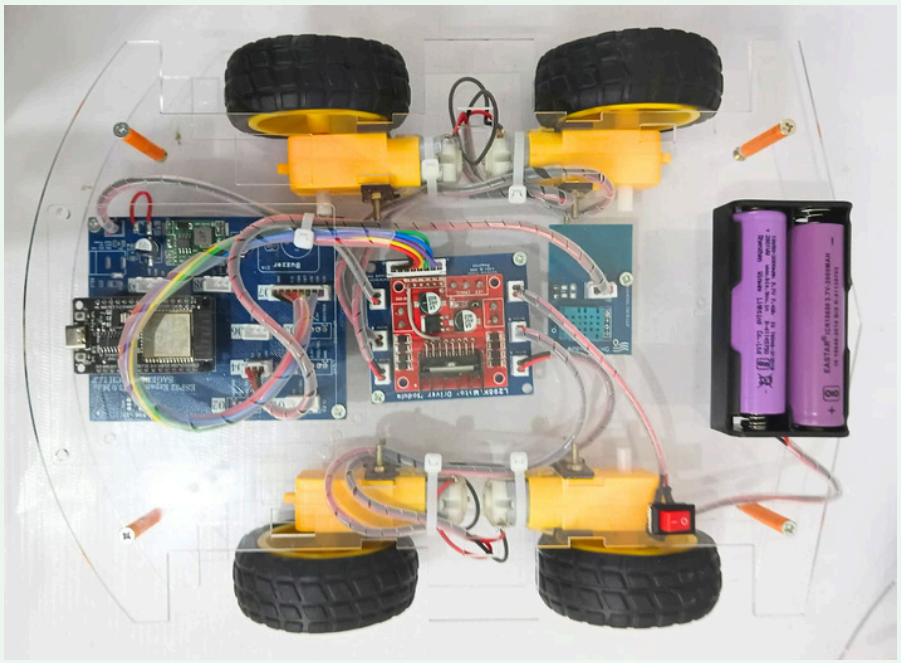


Step 8



Final Management

- Bundle all loose wires neatly using your spiral wire wraps and cable ties so they do not interfere with the wheels.
- Place the upper acrylic chassis over the spacers and secure it using screws.



Calibration:

It involves mapping the raw analog sensor readings between completely dry and fully saturated conditions to establish an accurate 0% to 100% moisture scale.

💻 Code

```
##include "BluetoothSerial.h"
#include "DHT.h"

BluetoothSerial SerialBT;

#define DHTPIN 18
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Motor Pins
#define IN1 27
#define IN2 26
#define IN3 25
#define IN4 33
#define ENA 14
#define ENB 32

void setup() {
  Serial.begin(115200);
  SerialBT.begin("ESP32_CAR");

  dht.begin();

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  ledcAttach(ENA, 1000, 8);
  ledcAttach(ENB, 1000, 8);

  stopCar();
}

void loop() {
  if (SerialBT.available()) {
    char cmd = SerialBT.read();
```

💻 Code

```
switch (cmd) {
      case 'F': forward();   break;
      case 'B': backward();  break;
      case 'L': turnLeft();  break;
      case 'R': turnRight(); break;
      case 'S': stopCar();   break;
      case 'T': sendTemp();  break;
    }
  }
}

void forward() {
  ledcWrite(ENA, 200);
  ledcWrite(ENB, 200);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void backward() {
  ledcWrite(ENA, 200);
  ledcWrite(ENB, 200);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

// Pivot left: right motor forward, left motor reverse
void turnLeft() {
  ledcWrite(ENA, 180);
  ledcWrite(ENB, 180);
  digitalWrite(IN1, HIGH);   // right side forward
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);    // left side reverse
  digitalWrite(IN4, HIGH);
}

// Pivot right: left motor forward, right motor reverse
void turnRight() {
  ledcWrite(ENA, 180);
  ledcWrite(ENB, 180);
  digitalWrite(IN1, LOW);    // right side reverse
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);   // left side forward
  digitalWrite(IN4, LOW);
}
```

💻 Code

```
void stopCar() {
  ledcWrite(ENA, 0);
  ledcWrite(ENB, 0);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}

void sendTemp() {
  float t = dht.readTemperature();
  float h = dht.readHumidity();

  if (isnan(t) || isnan(h)) {
    SerialBT.println("DHT Error");
    return;
  }

  SerialBT.print("Temp: ");
  SerialBT.print(t);
  SerialBT.print(" C  Hum: ");
  SerialBT.print(h);
  SerialBT.println(" %");
}
```