Lab name : Introduction to Integrated Robotic Car System

Description : The Integrated Robotic Car System is a self-driving vehicle that combines robotics, AI, and advanced sensors for autonomous navigation. It uses LiDAR, cameras, and GPS to detect surroundings and make real-time decisions, such as avoiding obstacles and adjusting speed. The system also integrates with IoT for remote monitoring and updates, ensuring safety and reliability. This technology aims to revolutionize transportation by offering safer, more efficient, and fully automated driving experiences.

Level : medium

LabGuide : Basic Assembly and Motor Control

Objective:
Familiarize with assembling the robotic car chassis and programming basic motor control for forward and backward movement.

Materials Needed:

- Robotic car chassis
- Motors with driver module (e.g., L298N)
- Microcontroller (e.g., Arduino Uno or similar)
- Battery pack
- Jumper wires

Steps:

1. **Assemble the Chassis**:
   - Attach wheels and motor mounts to the chassis as instructed in your kit.
   - Ensure the motors are securely fastened to avoid movement issues.
2. **Connect Motors to Motor Driver**:
   - Attach each motor to the motor driver module, connecting the motor's output pins to the motor driver's OUT pins.
   - Connect the motor driver's power and ground pins to the microcontroller's corresponding pins.
3. **Program Basic Motor Control**:

   - Write a simple Arduino code to control the motor movement. Use the code below to make the car

```cpp
CODE (C++)
const int motorPin1 = 3;
const int motorPin2 = 4;

void setup() {
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
}
```

```
void moveForward() {
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
}

void moveBackward() {
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
}

void stopCar() {
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
}

void loop() {
    moveForward();
    delay(2000);
    stopCar();
    delay(1000);
    moveBackward();
    delay(2000);
    stopCar();
    delay(1000);
}
```

- **Testing**:
  - Power on the robotic car and upload the code.
  - Verify the forward, stop, and backward movements are functioning as expected.
-