Lab name : Building an Autonomous Obstacle-Avoiding Robotic Car

Description : In this lab, students will learn to design and program a robotic car capable of avoiding obstacles autonomously. This lab combines programming, electronics, and basic robotics to give participants hands-on experience in building intelligent systems. Main topics covered include: Obstacle Detection: Participants will configure and utilize ultrasonic sensors to detect obstacles and avoid collisions. Decision Making with Microcontroller: Students will program the microcontroller to make real-time decisions based on sensor data, allowing the robotic car to navigate paths independently. Power Management: Understanding the power requirements and ensuring stable power distribution will be essential for uninterrupted operation.

Level : medium

Lab Guide : Integrating Obstacle Detection with Autonomous Navigation

Objective:
Combine obstacle detection with autonomous navigation, allowing the car to move forward and stop when an obstacle is detected.
Materials Needed:

- Robotic car with chassis and motors (assembled in Lab Guide 1)
- Ultrasonic sensor
- Microcontroller (e.g., Arduino Uno)
- Jumper wires

Steps:

1. **Mount the Ultrasonic Sensor**:
   Attach the ultrasonic sensor to the front of the robotic car to detect obstacles ahead.
2. **Connect the Ultrasonic Sensor**:
   - Connect the sensor’s VCC and GND to the microcontroller.
   - Attach the Trigger and Echo pins of the sensor to two digital pins on the microcontroller (e.g., pin 9 for Trigger, pin 10 for Echo).
3. **Write Code for Obstacle Detection**:
   - Program the ultrasonic sensor to measure distance, and use the sensor data to control the car’s movement. Here’s the code for distance measurement and movement control.

     Code (C++)

```
const int motorPin1 = 3;
 const int motorPin2 = 4;
 const int trigPin = 9;
 const int echoPin = 10;
 void setup() {
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
 }
 int getDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    long duration = pulseIn(echoPin, HIGH);
    int distance = duration * 0.034 / 2;
    return distance;
 }
 void loop() {
    int distance = getDistance();
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
    if (distance < 15) { // Obstacle within 15 cm
        digitalWrite(motorPin1, LOW);
        digitalWrite(motorPin2, LOW);
        delay(500);  // Stop briefly
    } else {
        digitalWrite(motorPin1, HIGH);
        digitalWrite(motorPin2, LOW);
    }
    delay(100);
 }
```