Obstacle-Avoiding Robot Using Arduino

By Prathyusha Baikani

(Robotics Engineer)

1. Introduction

An obstacle-avoiding robot is an autonomous robot that moves freely and avoids obstacles using ultrasonic sensors.

It continuously scans for obstacles in its path and changes direction when necessary. This project is ideal for beginners in robotics and helps in understanding the basics of sensors, motor control, and programming with Arduino.

2. Uses & Applications

- Autonomous navigation in robotics.
- Smart vacuum cleaners.
- Automated delivery bots.
- Educational robotics projects.

3. Components Required

1. **Arduino Uno** – Microcontroller to control the robot.
2. **Ultrasonic Sensor (HC-SR04)** – Detects obstacles.
3. **L298N Motor Driver Module** – Controls the motors.
4. **DC Motors with Wheels** – Enables movement.
5. **Chassis** – Base for mounting components.
6. **Battery (9V or 12V Li-ion)** – Power supply.
7. **Jumper Wires** – Connecting components.

4. Circuit Diagram

- **VCC** of Ultrasonic Sensor → **5V on Arduino**
- **GND** of Ultrasonic Sensor → **GND on Arduino**

- **Trig Pin → Digital Pin 9**
- **Echo Pin → Digital Pin 10**
- **Motor Driver Inputs → Arduino Digital Pins 4, 5, 6, 7**

5. Code Explanation (GitHub)

```
// Define pins for ultrasonic sensor
#define trigPin 9   // Trigger pin for ultrasonic sensor
#define echoPin 10  // Echo pin for ultrasonic sensor

// Define pins for motor driver
#define motor1A 4   // Motor 1 forward pin
#define motor1B 5   // Motor 1 backward pin
#define motor2A 6   // Motor 2 forward pin
#define motor2B 7   // Motor 2 backward pin

void setup() {
  // Set ultrasonic sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

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

  // Initialize serial communication for debugging
  Serial.begin(9600);
}

void loop() {
  long duration;
  int distance;

  // Send ultrasonic pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the time for echo to return
  duration = pulseIn(echoPin, HIGH);

  // Convert time to distance (in cm)
  distance = duration * 0.034 / 2;

  // Print the distance to the serial monitor
  Serial.println(distance);

  // Check if an obstacle is detected within 15 cm
  if (distance < 15) {
    stopRobot();   // Stop the robot
    delay(500);    // Wait for a short duration
    turnRight();   // Turn right to avoid the obstacle
    delay(600);    // Allow time for turning
  } else {
    moveForward(); // Move forward if no obstacle is detected
  }
}

// Function to move the robot forward
void moveForward() {
  digitalWrite(motor1A, HIGH);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor2A, HIGH);
  digitalWrite(motor2B, LOW);
}

// Function to stop the robot
void stopRobot() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, LOW);
  digitalWrite(motor2A, LOW);
  digitalWrite(motor2B, LOW);
}
```

```
// Function to turn the robot right
void turnRight() {
  digitalWrite(motor1A, LOW);
  digitalWrite(motor1B, HIGH);
  digitalWrite(motor2A, HIGH);
  digitalWrite(motor2B, LOW);  }
```

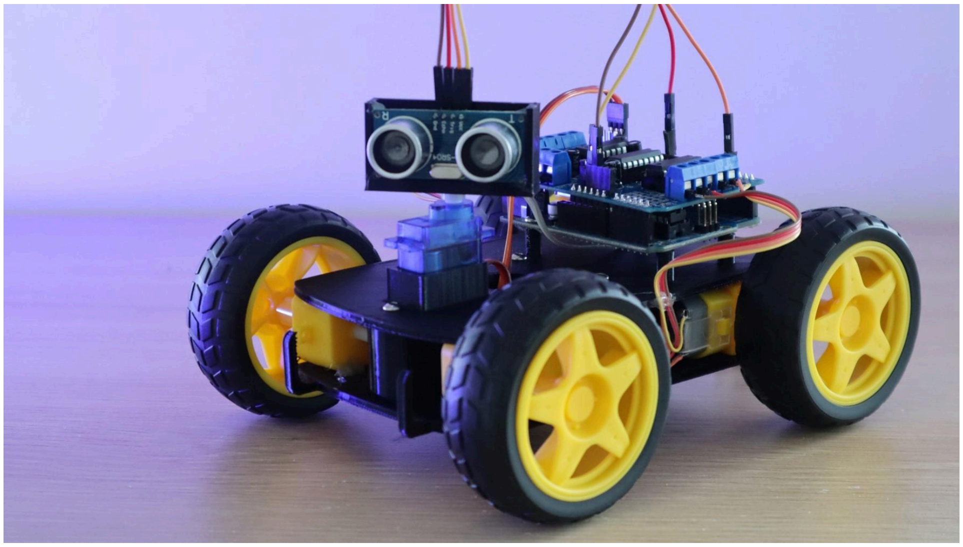
6. Working Principle

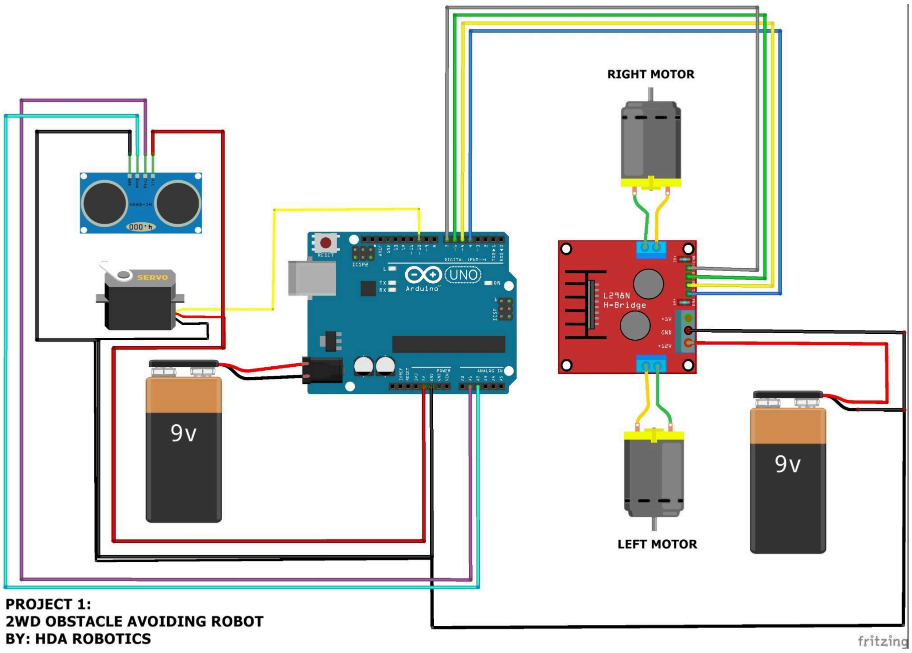
- The **ultrasonic sensor** sends out sound waves.
- If an obstacle is detected, it calculates the distance.
- If the distance is less than **15 cm**, the robot stops and turns.
- If no obstacle is detected, the robot moves forward.

7. Advantages

- Fully autonomous navigation.
- No need for human intervention.
- Teaches basics of Arduino and robotics.
- Can be enhanced with AI for smart navigation.

8. Images & Circuit Diagram





9. Future Enhancements

- Adding a **Bluetooth module** for remote control.
- Implementing **AI-based path planning**.
- Using **multiple sensors** for better navigation.

—---THANK YOU —---