# Project Title: Obstacle-Avoiding Robot with Arduino and Ultrasonic Sensor

**1. Introduction**

This project constructs an autonomous mobile robot that effectively navigates its environment by detecting and circumventing obstacles using an ultrasonic sensor and an Arduino Uno microcontroller.

**2. Project Objectives**

* Design and build a functional robot chassis that can accommodate the essential electronic components.

* Implement obstacle detection using an ultrasonic sensor interfaced with the Arduino.

* Develop control algorithms for the robot to move forward and avoid obstacles.

* Integrate all components and program the robot's autonomous behavior.

**3. Materials used**

*** Hardware:**

* Arduino Uno microcontroller

* HC-SR04 ultrasonic sensor (operating voltage: 5V, maximum sensing range: 400 cm)

* L298N motor driver module (refer to datasheet for specific voltage and current ratings)

* Two DC gear motors (compatible voltage and torque with the motor driver)

* Robot chassis kit (with wheels)

* Jumper wires

* Breadboard (optional)

* Power supply (AA batteries or DC wall adapter with voltage matching the Arduino's requirements)

*** Software:**

* Arduino IDE (Integrated Development Environment)

* NewPing library (for simplified ultrasonic sensor communication)

**4. System Design**

*** Hardware Design:**

* The robot chassis should be sturdy and well-balanced to accommodate the weight of the electronic components (Arduino, motor driver, ultrasonic sensor, and power supply). Consider using acrylic sheets, wood, or 3D-printed parts for the chassis construction.

* The L298N motor driver module acts as an intermediary between the Arduino and the DC gear motors, enabling control over their direction and speed. Ensure the motor driver can handle the current requirements of the motors.

* The ultrasonic sensor (HC-SR04) functions as the robot's primary obstacle detection mechanism. It emits sound waves and interprets the reflected echoes to determine object proximity.

*** Software Design:**

* The Arduino IDE is used to program the robot's behavior.

* The NewPing library simplifies communication with the ultrasonic sensor, providing a user-friendly interface for reading distance measurements.

* The core logic involves continuously reading sensor data, implementing obstacle detection using a threshold value (e.g., 30 cm), and incorporating obstacle avoidance maneuvers (e.g., turning, reversing) when an obstacle is detected within the threshold distance.

**5. Assembly Instructions**

*** Chassis Assembly:**

* Carefully construct the robot chassis following the manufacturer's instructions. Ensure the chassis is robust and can house the motors, sensors, and microcontroller securely.

*** Motor Driver and Motor Connection:**

* Refer to the L298N motor driver's datasheet for specific pin connections.

* Connect the L298N motor driver to the Arduino Uno as follows:

| Arduino Pin | L298N Pin | Function |

|---|---|---|

| 9 | IN1 | Motor 1 (Right) – Forward |

| 10 | IN2 | Motor 1 (Right) – Backward |

| 5 | IN3 | Motor 2 (Left) – Forward |

| 6 | IN4 | Motor 2 (Left) – Backward |

* Connect the DC gear motors to the corresponding outputs of the L298N motor driver, ensuring proper voltage and current compatibility.

*** Ultrasonic Sensor Connection:**

* Connect the HC-SR04 ultrasonic sensor to the Arduino Uno as follows:

| Arduino Pin | HC-SR04 Pin | Function |

|---|---|---|

| 12 | Trig | Trigger pulse |

| 11 | Echo | Echo pulse |

* You can use a breadboard for easier prototyping or connect directly using jumper wires.

*** Power Supply Connection:**

* Connect the power supply to the Arduino Uno (either via the DC barrel jack or the Vin and GND pins). Ensure the voltage rating of the power supply matches the Arduino's requirements (typically 5V).

**6. Code**

The provided Arduino code incorporates the NewPing library for simplified ultrasonic sensor communication.

```
#include <NewPing.h>
const int trigPin = 12;  // Ultrasonic sensor trigger pin
const int echoPin = 11;  // Ultrasonic sensor echo pin
const int motor1ForwardPin = 9;  // Motor 1 forward pin (right motor)
const int motor1BackwardPin = 10; // Motor 1 backward pin
const int motor2ForwardPin = 5;  // Motor 2 forward pin (left motor)
const int motor2BackwardPin = 6;  // Motor 2 backward pin

NewPing sonar(trigPin, echoPin, 400); // Maximum distance is 400 cm

int threshold = 30; // Obstacle detection threshold distance (cm)

void setup() {
pinMode(motor1ForwardPin, OUTPUT);
pinMode(motor1BackwardPin, OUTPUT);
pinMode(motor2ForwardPin, OUTPUT);
pinMode(motor2BackwardPin, OUTPUT);
}

void loop() {
long duration = sonar.ping();
int distance = duration / 2 / 29.1; // Convert duration to distance (cm)

if (distance <= threshold) {
stopMotors();  // Stop the robot if an obstacle is detected
delay(100);    // Wait for a brief period
```

```
// Implement obstacle avoidance logic here
// (e.g., turn left, turn right, reverse)
turnLeft();
} else {
moveForward();  // Move forward if no obstacle is detected
}

delay(100);
}

void moveForward() {
digitalWrite(motor1ForwardPin, HIGH);
digitalWrite(motor1BackwardPin, LOW);
digitalWrite(motor2ForwardPin, HIGH);
digitalWrite(motor2BackwardPin, LOW);
}

void stopMotors() {
digitalWrite(motor1ForwardPin, LOW);
digitalWrite(motor1BackwardPin, LOW);
digitalWrite(motor2ForwardPin, LOW);
digitalWrite(motor2BackwardPin, LOW);
}

void turnLeft() {
// Implement logic to turn the robot left (e.g., reverse right motor, move forward left motor)
digitalWrite(motor1ForwardPin, LOW);
digitalWrite(motor1BackwardPin, LOW);
digitalWrite(motor2ForwardPin, HIGH);
digitalWrite(motor2BackwardPin, LOW);
delay(500);
```

**Conclusion:** I have tested this obstacle avoiding robot in different places it gives good results ,it finds obstacles accurately

THANK YOU FOR READING THIS DOCUMENTATION