# Circuit Documentation

## Summary

This circuit is designed to control a mobile robot with obstacle avoidance capabilities. It uses an Arduino UNO as the central microcontroller to manage inputs from an HC-SR04 ultrasonic sensor and outputs to control a servo motor and two DC motors through an L298N motor driver. The Arduino is powered by a 9V battery, while the motor driver is powered by a separate 12V battery to handle the higher current requirements of the motors.

## Component List

### Arduino UNO

- Microcontroller board based on the ATmega328P
- It has 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button.

### 9V Battery

- Provides power to the Arduino UNO.

### L298N DC Motor Driver

- An integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages.
- It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors.

### Servo

- A rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration.

### 12V Battery

- Provides power to the L298N motor driver and the DC motors.

### DC Motors

- Two plastic geared DC motors used for the robot's movement.

### HC-SR04 Ultrasonic Sensor

- An ultrasonic distance sensor that provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm.

# Wiring Details

## Arduino UNO

- **GND** connected to the 9V Battery negative terminal and the GND pins of the Servo and HC-SR04 Ultrasonic Sensor.
- **Vin** connected to the 9V Battery positive terminal.
- **3.3V** connected to the Servo VCC.
- **5V** connected to the HC-SR04 Ultrasonic Sensor VCC.
- **D2** connected to the Servo PWM.
- **D3, D5, D6, D11** connected to the L298N DC motor driver IN1, IN2, IN3, IN4 respectively.
- **D9** connected to the HC-SR04 Ultrasonic Sensor TRIG.
- **D10** connected to the HC-SR04 Ultrasonic Sensor ECHO.

## L298N DC Motor Driver

- **GND** connected to the 12V Battery negative terminal.
- **12V** connected to the 12V Battery positive terminal.
- **OUT1, OUT2, OUT3, OUT4** connected to the DC Motors.

## Servo

- **GND** connected to the Arduino UNO GND.
- **VCC** connected to the Arduino UNO 3.3V.
- **PWM** connected to the Arduino UNO D2.

## DC Motors

- Each motor has one terminal connected to the L298N DC motor driver OUT pins and the other terminal to the L298N GND.

## HC-SR04 Ultrasonic Sensor

- **VCC** connected to the Arduino UNO 5V.
- **TRIG** connected to the Arduino UNO D9.
- **ECHO** connected to the Arduino UNO D10.
- **GND** connected to the Arduino UNO GND.

# Documented Code

```
#include <Servo.h>

const int trigPin = 9;
const int echoPin = 10;
const int servoPin = 2;
const int motor1PinA = 3;
const int motor1PinB = 5;
const int motor2PinA = 6;
const int motor2PinB = 11;

Servo myservo;
int duration, distance, servoAngle;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(motor1PinA, OUTPUT);
  pinMode(motor1PinB, OUTPUT);
  pinMode(motor2PinA, OUTPUT);
  pinMode(motor2PinB, OUTPUT);
  myservo.attach(servoPin);
  myservo.write(0); // Initialize servo to 0 degrees
}

void loop() {
  distance = getDistance();

  if (distance <= 20) {
    avoidObstacle();
  } else {
    moveForward();
  }
}

int getDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  return duration * 0.034 / 2;
}

void moveForward() {
  digitalWrite(motor1PinA, HIGH);
  digitalWrite(motor1PinB, LOW);
  digitalWrite(motor2PinA, HIGH);
  digitalWrite(motor2PinB, LOW);
}

void stopMotors() {
  digitalWrite(motor1PinA, LOW);
  digitalWrite(motor1PinB, LOW);
  digitalWrite(motor2PinA, LOW);
  digitalWrite(motor2PinB, LOW);
}

void avoidObstacle() {
  stopMotors();
  moveBackward();
  delay(500);
  stopMotors();

  if (scanForClearPath()) {
    turnRover();
  }
}

void moveBackward() {
  digitalWrite(motor1PinA, HIGH);
  digitalWrite(motor1PinB, LOW);
  digitalWrite(motor2PinA, HIGH);
  digitalWrite(motor2PinB, LOW);
}

bool scanForClearPath() {
  for (servoAngle = 0; servoAngle <= 90; servoAngle += 10) {
    myservo.write(servoAngle);
    delay(100);
    if (getDistance() <= 20) {
      return false;
    }
  }
  for (servoAngle = 90; servoAngle >= 0; servoAngle -= 10) {
    myservo.write(servoAngle);
    delay(100);
    if (getDistance() <= 20) {
      return false;
    }
  }
  return true;
}

void turnRover() {
  if (servoAngle > 45) {
    // Turn right
    digitalWrite(motor1PinA, HIGH);
    digitalWrite(motor1PinB, LOW);
    digitalWrite(motor2PinA, LOW);
    digitalWrite(motor2PinB, HIGH);
  } else {
    // Turn left
    digitalWrite(motor1PinA, LOW);
    digitalWrite(motor1PinB, HIGH);
    digitalWrite(motor2PinA, HIGH);
    digitalWrite(motor2PinB, LOW);
  }
  delay(500);
  stopMotors();
}
```

This code is responsible for controlling the robot's movement and obstacle avoidance. It initializes the servo and motors, measures distance using the ultrasonic sensor, and determines the robot's actions based on the distance to obstacles. The robot can move forward, backward, stop, and turn based on the sensor readings and servo position.