***🚦 OVERALL FLOW OF THE ROBOT***

1. Ultrasonic sensor mounted on a servo scans the surroundings (front, left, right).
2. The Arduino reads distance data from the sensor.
3. If an obstacle is detected in front, the robot:

* Stops.
* Rotates the servo to check left and right.
* Chooses the direction with more space and turns that way.

1. If no obstacle, the robot moves forward.

***🪛 CONNECTIONS***

🔌 Ultrasonic Sensor (HC-SR04):

|  |  |
| --- | --- |
| **Sensor Pin** | **Arduino UNO** |
| VCC | 5V |
| GND | GND |
| Trig | D9 |
| Echo | D8 |

**⚙️ *Servo Motor:***

|  |  |
| --- | --- |
| **Servo Pin** | **Arduino UNO** |
| Signal | D10 |
| VCC | 5V(external if needed) |
| GND | GND |

***⚡ L298N Motor Driver:***

|  |  |  |
| --- | --- | --- |
| L298N Pins | Arduino UNO | Description |
| IN1 | D2 | Motor A control 1 |
| IN2 | D3 | Motor A control 2 |
| IN3 | D4 | Motor B control 1 |
| IN4 | D5 | Motor B control 2 |
| ENA | 5V (jumper or PWM pin like D6) |  |
| ENB | 5V (jumper or PWM pin like D7) |  |
| Motor A Output | Left DC Motor |  |
| Motor B Output | Right DC Motor |  |
| VCC | External Battery (eg:, 9V or 12V) |  |
| GND | GND from Arduino and Battery |  |
| 5V (Regulated) | Connect jumper if powering Arduino from L298N |  |

**🔧 *ARDUINO CODE***

Make sure you’ve installed the Servo library in Arduino IDE. Here's the tested and corrected version of the code:

#include <Servo.h>

// Ultrasonic sensor pins

#define trigPin 9

#define echoPin 8

// Servo pin

#define servoPin 10

// Motor driver pins

#define IN1 2

#define IN2 3

#define IN3 4

#define IN4 5

Servo myServo;

long readDistance() {

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

long duration = pulseIn(echoPin, HIGH, 30000); // Timeout after 30ms

if (duration == 0) return 999; // No echo received

return duration \* 0.034 / 2;

}

void moveForward() {

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

}

void turnLeft() {

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

}

void turnRight() {

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

}

void stopMotors() {

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

}

void setup() {

Serial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

myServo.attach(servoPin);

myServo.write(90); // Face forward

delay(1000);

}

void loop() {

myServo.write(90);

delay(300);

long frontDist = readDistance();

Serial.print("Front Distance: ");

Serial.println(frontDist);

if (frontDist < 20) {

stopMotors();

delay(500);

myServo.write(0); // look left

delay(500);

long leftDist = readDistance();

Serial.print("Left Distance: ");

Serial.println(leftDist);

myServo.write(180); // look right

delay(500);

long rightDist = readDistance();

Serial.print("Right Distance: ");

Serial.println(rightDist);

myServo.write(90);

delay(300);

if (leftDist > rightDist) {

turnLeft();

delay(600);

} else {

turnRight();

delay(600);

}

} else {

moveForward();

}

}

**⚠️ *Power Supply Tips:***

Use external battery (9V/12V) to power the L298N and motors.

Don't run servo and motors off the Arduino's 5V directly — it can reset or brown-out the board.

Consider using a power bank or regulated 5V module for the servo.

**🔍** ***TROUBLESHOOTING STEPS***

1. ✅ Make sure you selected the correct board:

Tools > Board > "Arduino Uno"

2. ✅ Check the correct COM port is selected:

Tools > Port > COMX (your Arduino port)

3. ✅ Verify library:

The Servo library should come built-in with Arduino IDE.

If not: Sketch > Include Library > Manage Libraries > Search "Servo" > Install

4. ✅ Avoid using pins 0 and 1 if you're using Serial Monitor — they are shared with USB communication.

5. ✅ Don’t forget to click "Verify" (✔️) first to check for errors before uploading.

***GitHub****: Darshan-213*