Arduino obstacle avoiding robot car

1. Project Overview :-

This autonomous obstacle-avoiding car has significant potential for future applications in various fields. It can serve as a foundation for developing advanced autonomous systems, including self-driving vehicles, automated delivery robots, and smart industrial equipment. The technology can be adapted for use in agricultural automation, where self-navigating machines can optimize farming processes. In search and rescue operations, similar autonomous vehicles could help navigate hazardous areas without human intervention. Furthermore, it opens the door to enhancing robotics research, leading to more efficient, intelligent transportation and logistics systems.

2. Components Used :-

1)**Arduino Uno**: The microcontroller that serves as the brain of the project.

2)Ultrasonic Sensor (HC-SR04): Measures the distance to objects in front of the car.

3)Servo Motor: Rotates the ultrasonic sensor to scan the surroundings.

4)Motor Driver (L298N or similar): Controls the movement of the car's motors.

5)DC Motors (2 or 4): Provide forward, backward, and turning movement to the car.

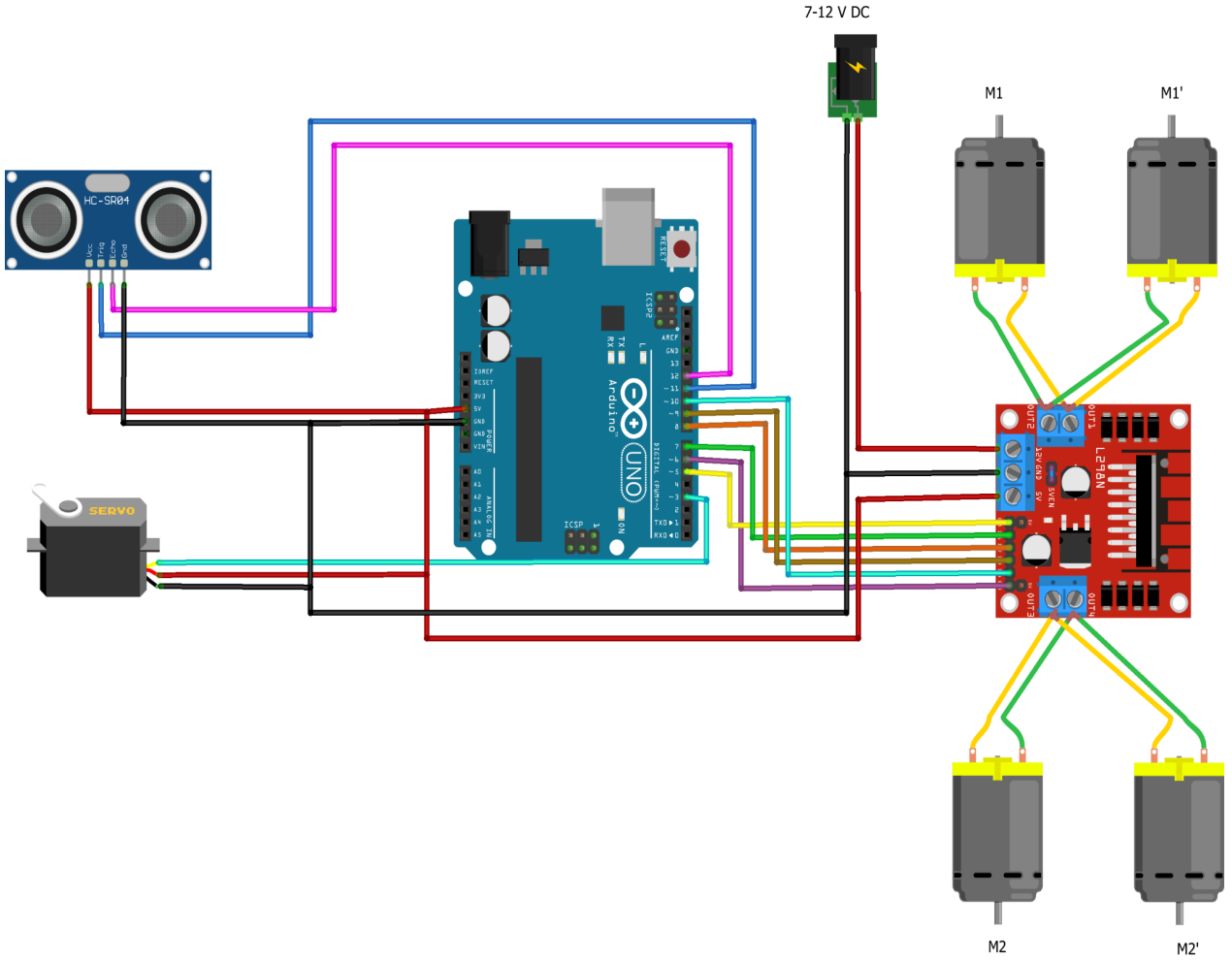
6) lithium battery:-

3. Key Features:-

1. ***Obstacle Detection****:* The ultrasonic sensor continuously scans the area in front of the car. If an object is detected within a set distance (e.g., 15 cm), the car stops, reverses, and checks for a clear path on either the left or right side.

2. ***Servo-Driven Scanning:*** A servo motor moves the ultrasonic sensor from left to right to scan the surroundings, helping the car determine the best direction to turn when an obstacle is detected.
3. ***Autonomous Navigation:*** The car automatically makes decisions on whether to turn left or right based on the available space, allowing it to navigate through complex environments without external control.

4. Circuit Diagram:-



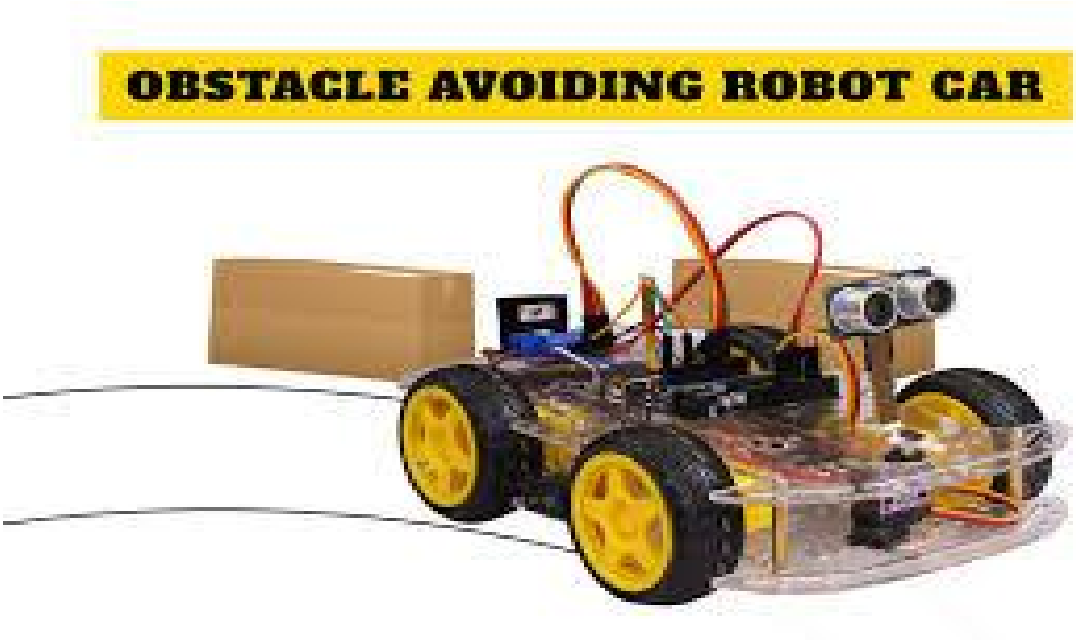
4.

Ultrasonic Sensor connected to digital pins for trigger and echo.

Servo Motor attached to a PWM pin for directional control of the sensor.

Motor Driver controlling the motors via digital pins for movement (forward, backward, and turning).

#Modal:-



5. Working Principle:-

1. ***Continuous Forward Movement****:* By default, the car moves forward in a loop.
2. ***Obstacle Detection:***

   When an obstacle is detected at a certain distance (e.g., 15 cm or less), the car:

   - Stops.
   - Reverses for a short duration.
   - Uses the servo motor to scan both left and right directions.
3. ***Turning Decision****:*
   - Based on the ultrasonic sensor readings, the car determines whether to turn left or right, depending on which direction has more space.
4. ***Resuming Movement****:* After turning, the car resumes its forward motion, repeating the process for each new obstacle.

6. Code Overview:-

***#Ultrasonic Sensor Control**:* Used to measure the distance to obstacles and feed this information into the car's movement logic.

#Motor Control: The motor driver is used to move the car forward, backward, or turn based on the obstacle detection logic.

#Servo Control: The servo motor rotates the ultrasonic sensor to scan both left and right, helping the car make decisions.

7. Code :-

```
#include <Servo.h>

int trigPin = 9; // Ultrasonic Sensor Pins

int echoPin = 10;


int in1 = 2;

int in2 = 3;      // Motor Pins


int in3 = 4;

int in4 = 5;

Servo myservo;


long duration;

int distance;

boolean goesForward = false;

const int MAX_SPEED = 255;
```

```
void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  myservo.attach(11);
  myservo.write(115);  // Start with servo looking straight ahead
  delay(2000);
}

void loop() {
  distance = readPing();

  if (distance <= 15) {
    moveStop();
    delay(100);
    moveBackward();
    delay(300);
    moveStop();
    delay(200);

    int distanceR = lookRight();
    delay(200);
    int distanceL = lookLeft();
    delay(200);
```

```
    if (distanceR >= distanceL) {
      turnRight();
      moveStop();
    } else {
      turnLeft();
      moveStop();
    }
  } else {
    moveForward();  // Move forward if no obstacle is detected
  }
}

// Function to look right with servo
int lookRight() {
  myservo.write(50);  // Turn servo to the right
  delay(500);
  int distance = readPing();
  delay(100);
  myservo.write(115);  // Return to center position
  return distance;
}
 int lookLeft() {        // Function to look left with servo

  myservo.write(170);  // Turn servo to the left
  delay(500);
  int distance = readPing();
```

```
  delay(100);
  myservo.write(115);  // Return to center position
  return distance;
}

int readPing() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  int cm = (duration * 0.034) / 2;
  if (cm == 0) cm = 250;  // Assume no object detected within max range
  return cm;
}
void moveStop() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
void moveForward() {
  if (!goesForward) {
    goesForward = true;
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
```

```
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(in1, MAX_SPEED);
    analogWrite(in3, MAX_SPEED);
  }
}
void moveBackward() {
  goesForward = false;
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(in2, MAX_SPEED);
  analogWrite(in4, MAX_SPEED);
}
void turnRight() {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(500);
  moveForward();  // Resume forward after turn

void turnLeft() {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, HIGH);
```

```
  digitalWrite(in4, LOW);
  delay(500);
  moveForward();  // Resume forward after turn
}
```