**NAME: ADWAIT MILIND INGLE**

**ID: 20004054**

**PROJECT: OBSTACLE AVOIDING CAR**

**ABSTRACT**

In this project, we present the design and implementation of an obstacle-avoiding car using readily available components such as an Arduino Uno microcontroller, an L298 motor driver, and 60 RPM DC motors. The aim of this project is to create a cost-effective and versatile autonomous vehicle capable of navigating its environment while avoiding obstacles.

The core of the system is the Arduino Uno, which serves as the brain of the car, processing sensor data and making real-time decisions. Two DC motors with a speed of 60 RPM are employed to control the car's movement. The L298 motor driver provides the necessary control and power amplification to drive the motors effectively.

To achieve obstacle avoidance, we integrate ultrasonic distance sensors on the front of the car. These sensors emit ultrasonic waves and measure the time it takes for the waves to bounce back. By analyzing this data, the Arduino Uno can calculate the distance to any obstacles in its path. When an obstacle is detected within a predefined range, the car's control system initiates appropriate maneuvers to avoid collision. This includes stopping, reversing, and steering in a different direction.

The development of this obstacle-avoiding car offers several educational and practical benefits. It provides an excellent platform for students and hobbyists to gain hands-on experience with microcontroller programming, sensor integration, and motor control. Moreover, it has potential applications in various fields, such as surveillance, automation, and education, where autonomous navigation and obstacle avoidance are essential.

In conclusion, the obstacle-avoiding car described in this project demonstrates the feasibility of creating an autonomous vehicle using accessible components. By combining the power of the Arduino Uno, L298 motor driver, and 60 RPM DC motors with ultrasonic sensors, this project opens up possibilities for further innovation and exploration in the field of robotics and automation.

**INDEX**

|  |  |
| --- | --- |
| **Chapter 1:** | **Introduction**  **1.1 Objective**  **1.2 Problem statement** |
| **Chapter 2:** | **Methodology**  **2.1 Working**  **2.2 Circuit Diagram**  **2.3 Circuit Connections** |
| **Chapter 3:** | **Process & Requirement**  **3.1 Hardware Requirements**  **3.2 Component Description**  **3.2.1 Arduino UNO**  **3.2.2 Ultrasonic Sensor**  **3.2.3 L298 Motor Driver**  **3.2.4 9v Battery and Connecting wires** |
| **Chapter 4:** | **Codes** |
| **Chapter 5:** | **Conclusion** |

## 1. INTRODUCTION:

The development of autonomous vehicles has garnered significant attention in recent years, with applications ranging from self-driving cars to robotic systems. In this context, our project introduces an obstacle-avoiding car, a versatile and cost-effective autonomous vehicle solution. Powered by an Arduino Uno microcontroller and equipped with an L298 motor driver, this car incorporates two DC motors operating at 60 RPM. Its primary objective is to navigate its environment autonomously while detecting and avoiding obstacles.

The integration of ultrasonic distance sensors allows the car to measure the proximity of objects and make real-time decisions to circumvent potential collisions. This project serves as an educational and practical platform for those interested in exploring microcontroller programming, sensor integration, and motor control. Furthermore, its applications extend to various domains, including robotics, automation, and educational robotics, highlighting the relevance and significance of this endeavor in modern technology and innovation.

**1.1 OBJECTIVIES**:

1. Develop a self-guided car capable of autonomously maneuvering through various environments, including indoors and outdoors.

2. Implement robust obstacle detection mechanisms using sensors to identify objects or barriers in the car's path.

3. Create intelligent algorithms that enable the car to make immediate and appropriate decisions for obstacle avoidance.

4. Achieve precise control over the DC motors to ensure the car's movements are accurate and smooth during navigation.

5. Prioritize safety by incorporating fail-safe mechanisms to prevent collisions and protect the car and its surroundings.

6. Optimize the design for cost-efficiency while maintaining the functionality and reliability of the obstacle-avoiding system.

**1.2 PROBLEM STATEMENT:**

In the context of robotics and autonomous vehicles, the challenge lies in designing and implementing a cost-effective, reliable, and intelligent obstacle-avoiding car. This car must autonomously navigate its environment while detecting and successfully avoiding obstacles in real-time. Leveraging the Arduino Uno microcontroller, the L298 motor driver, and 60 RPM DC motors, the goal is to create a versatile and practical solution capable of addressing the complexities of varying terrains and obstacle scenarios.

### 2. METHODOLOGY

**2.1 WORKING:**

Firstly, assemble the hardware components, including the chassis, DC motors, L298 motor driver, Arduino Uno, and ultrasonic sensors. Next, program the Arduino to read data from the ultrasonic sensors and implement obstacle-detection algorithms. Configure the L298 motor driver to control the DC motors' speed and direction. Create a feedback loop where the Arduino continually processes sensor data, determining whether obstacles are within a defined range and deciding on appropriate motor control actions for obstacle avoidance. Test and fine-tune the system's performance, ensuring accurate obstacle detection and safe navigation. Finally, integrate the components into the car's chassis, power the system, and observe its autonomous obstacle-avoidance behavior in action, making adjustments as needed for optimal functionality and reliability.

**2.2 CIRCUIT DIAGRAM:**

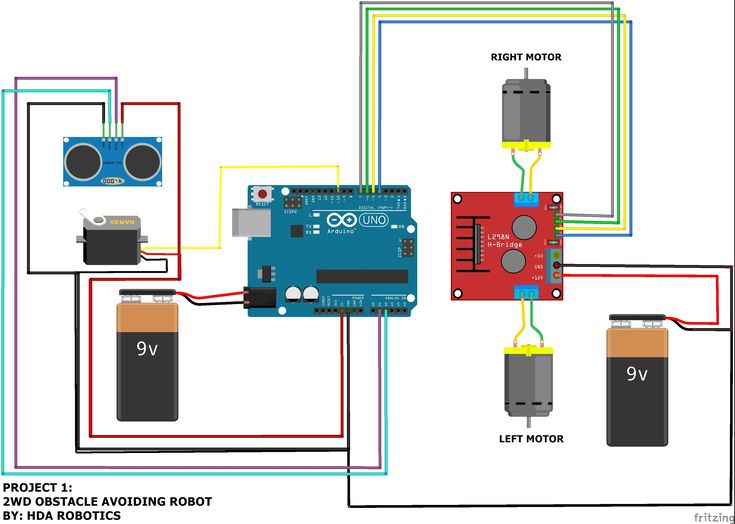


Figure 1: Obstacle Avoiding Car

[Courtesy: https://www.pinterest.com/pin/project-1-2wd-obstacle-avoiding-robot--853009985657937782/]

**2.3 CIRCUIT CONNECTIONS:**

1. Motor Connections:

- Connect the terminals of each DC motor to the motor outputs of the L298 motor driver. Typically, the motor terminals are labeled as "+," "-", and "S" (Signal). Connect the "+" and "-" terminals of each motor to the corresponding "+" and "-" outputs on the L298 motor driver.

- Connect the "S" (Signal) terminal of each motor to the "Out1" and "Out2" pins of the L298 motor driver for one motor and "Out3" and "Out4" for the other motor.

2. Power Supply:

- Connect the positive terminal of your power supply (e.g., a battery pack) to the "+12V" input of the L298 motor driver.

- Connect the negative terminal of your power supply to the "GND" (ground) of the L298 motor driver.

3. Arduino Uno to L298 Motor Driver:

- Connect the "ENA" (Enable A) and "ENB" (Enable B) pins on the L298 motor driver to two digital pins on the Arduino Uno (e.g., pins 5 and 6). These pins control the motor speed using PWM (Pulse Width Modulation).

4. Ultrasonic Sensor:

- Connect the VCC pin of the ultrasonic sensor to the 5V output on the Arduino Uno.

- Connect the GND (ground) pin of the ultrasonic sensor to the GND on the Arduino Uno.

- Connect the TRIG (trigger) pin of the ultrasonic sensor to a digital pin on the Arduino Uno (e.g., pin 7).

- Connect the ECHO pin of the ultrasonic sensor to another digital pin on the Arduino Uno (e.g., pin 8).

## 3. PROCESS AND REQUIREMENTS

**3.1 HARDWARE REQUIREMENTS:**

The following components are required to make Panic Alarm Circuit

| **S. NO** | **Component** | **Value** | **Qty** |
| --- | --- | --- | --- |
| 1. | Breadboard |  | 1 |
| 2. | Battery | 9v | 1 |
| 3. | Connecting Wires |  | 1 |
| 4. | Arduino UNO |  | 1 |
| 5. | Ultrasonic Sensor |  | 1 |
| 6. | L298 Motor Driver |  | 1 |
| 7. | DC Motors | 60 rpm | 2 |

**3.2 COMPONENT DESCRIPTION:**

* **ARDUINO UNO**
* **ULTRASONIC SENSOR**
* **L298 MOTOR DRIVER**
* **DC MOTORS (60 rpm)**
* **9V BATTERY AND CONNECTIONS WIRES**

### 3.2.1 ARDUINO UNO:

The Arduino Uno is a popular and versatile microcontroller board widely used in electronics and robotics projects. It boasts an ATmega328P microcontroller at its core, providing a reliable and easy-to-program platform for a wide range of applications. With a plethora of digital and analog input/output pins, the Arduino Uno enables seamless interfacing with sensors, actuators, and other components. Its open-source nature, along with a supportive community, allows users to access an extensive library of pre-written code and resources, simplifying the development process. The Arduino Uno's user-friendly IDE (Integrated Development Environment) makes programming accessible to both beginners and experienced engineers, making it an ideal choice for prototyping, automation, and educational projects.



Figure 2: Arduino UNO

[Courtesy: https://www.instructables.com/Obstacle-Avoiding-Robot-Arduino-1/]

**3.2.2 ULTRASONIC SENSOR:**

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

**3.2.3 L298 MOTOR DRIVER:**

The L298 motor driver is a popular integrated circuit designed for controlling and driving DC motors and stepper motors. Developed by SGS-Thomson Microelectronics (now STMicroelectronics), the L298 is widely used in robotics, automation, and other electronic projects where motor control is essential. This dual H-bridge motor driver offers several key features that make it a versatile choice for motor control applications. It can handle a wide range of voltage inputs, typically ranging from 7V to 46V, and provide a maximum current output of 2A per channel. This capability allows it to control a variety of motors, from small hobbyist motors to larger industrial-grade ones

The L298 features four input pins for each H-bridge, enabling precise control of motor direction and speed. By manipulating these input pins, you can make motors rotate clockwise, counterclockwise, or stop them altogether. Additionally, the L298 includes built-in flyback diodes, which help protect the circuit from voltage spikes generated by the motor when it is turned off.

One of the advantages of the L298 motor driver is its compatibility with various microcontrollers, such as Arduino, Raspberry Pi, and other popular development boards. This compatibility simplifies the process of integrating motor control into your projects, as you can easily interface the L298 with these platforms.

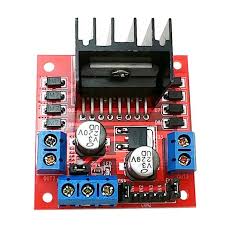


Figure 3: L298 Motor Driver

[Courtesy: https://www.instructables.com/Obstacle-Avoiding-Robot-Arduino-1/]

##### 3.2.4 9V BATTERY AND CONNECTING WIRES

The 9V battery and connecting wires are essential elements in a panic alarm system, providing the necessary power and interconnections to ensure the reliable operation of the system.

#### 4. CODES

#include <Servo.h> //Servo motor library. This is standard library

#include <NewPing.h> //Ultrasonic sensor function library. You must install this library

//our L298N control pins

const int LeftMotorForward = 7;

const int LeftMotorBackward = 6;

const int RightMotorForward = 4;

const int RightMotorBackward = 5;

//sensor pins

#define trig\_pin A1 //analog input 1

#define echo\_pin A2 //analog input 2

#define maximum\_distance 200

boolean goesForward = false;

int distance = 100;

NewPing sonar(trig\_pin, echo\_pin, maximum\_distance); //sensor function

Servo servo\_motor; //our servo name

void setup(){

pinMode(RightMotorForward, OUTPUT);

pinMode(LeftMotorForward, OUTPUT);

pinMode(LeftMotorBackward, OUTPUT);

pinMode(RightMotorBackward, OUTPUT);

servo\_motor.attach(10); //our servo pin

servo\_motor.write(115);

delay(2000);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

}

void loop(){

int distanceRight = 0;

int distanceLeft = 0;

delay(50);

if (distance <= 20){

moveStop();

delay(300);

moveBackward();

delay(400);

moveStop();

delay(300);

distanceRight = lookRight();

delay(300);

distanceLeft = lookLeft();

delay(300);

if (distance >= distanceLeft){

turnRight();

moveStop();

}

else{

turnLeft();

moveStop();

}

}

else{

moveForward();

}

distance = readPing();

}

int lookRight(){

servo\_motor.write(50);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(115);

return distance;

}

int lookLeft(){

servo\_motor.write(170);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(115);

return distance;

delay(100);

}

int readPing(){

delay(70);

int cm = sonar.ping\_cm();

if (cm==0){

cm=250;

}

return cm;

}

void moveStop(){

digitalWrite(RightMotorForward, LOW);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

digitalWrite(LeftMotorBackward, LOW);

}

void moveForward(){

if(!goesForward){

goesForward=true;

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

}

void moveBackward(){

goesForward=false;

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

**5. CONCLUSION:**

In conclusion, the obstacle-avoiding car project, powered by Arduino Uno, L298 motor driver, and 60 RPM DC motors, represents a successful integration of hardware and software to achieve autonomous navigation and obstacle avoidance. Through the collaboration of these components, the car has demonstrated its ability to detect obstacles, make real-time decisions, and navigate around them with precision and efficiency. This project not only showcases the potential of robotics and automation but also serves as an educational platform for enthusiasts and students to explore the intricacies of microcontroller programming, sensor integration, and motor control. The obstacle-avoiding car's practicality extends to various domains, including surveillance, automation, and educational robotics, highlighting its relevance in the ever-evolving world of technology and innovation. As a testament to the power of creativity and engineering, this project inspires further exploration and development in the field of autonomous vehicles and robotics.