Object Avoidance Robot

First Member:
Ahmed Mohamed Mounir
ID: 224643
MSA University
Third Year

Second Member:
Amr Khaled Abdeltawab
ID: 224363
MSA University
Third Year

May 19, 2024

Supervised by Prof. Ahmed ElShafee

Abstract

This project involves the creation of an object avoidance moving small vehicle using Arduino Uno, motors, motor shield, wheels, batteries, and an ultrasonic sensor. The vehicle is designed to navigate around obstacles by detecting empty directions and moving towards them. The main objective is to develop a reliable and efficient robotic system capable of autonomous navigation. The outcomes include successful implementation of the robot and demonstration of its capabilities.

Contents

Chapter 1

Introduction

1.1 Overview

The objective of this project is to design and implement a small autonomous vehicle capable of avoiding obstacles. Utilizing an Arduino Uno microcontroller, the vehicle employs an ultrasonic sensor to detect obstacles and a servo motor to determine the optimal path for navigation.

1.2 Importance and Relevance

Autonomous robots are becoming increasingly important in various applications, ranging from industrial automation to household chores. This project provides a practical introduction to robotic interfacing and programming, highlighting essential concepts in autonomous navigation.

1.3 Objectives

The main objectives of this project are:

- To design a small vehicle that can navigate autonomously.
- To implement obstacle detection and avoidance using an ultrasonic sensor.
- To program the vehicle's movement logic using Arduino.

Chapter 2

Project Requirements

2.1 Description

The project requirements as outlined in the course include the following components:

- Arduino Uno microcontroller: the main control unit for the robot.
- L298N Motor Driver Shield: to control the motors.
- Two DC motors: for driving the wheels.
- Wheels: for mobility.
- AA batteries and battery holder: to power the motors and the Arduino.
- HC-SR04 Ultrasonic Sensor: for obstacle detection.
- Servo motor: to rotate the ultrasonic sensor for scanning.
- Switch: to turn the power on and off.
- Connecting cables: for all necessary electrical connections.

2.2 Specifications

The vehicle is required to:

- Detect obstacles within a specified range.

- Change direction based on the detected obstacles.
- Move autonomously without human intervention.

Chapter 3

Literature Review

3.1 Existing Solutions

Various object avoidance robots exist, ranging from simple sensor-based robots to advanced AI-driven models. Many use ultrasonic or infrared sensors for obstacle detection.

3.2 Proposed Improvements

Our project focuses on simplicity and cost-effectiveness, making use of readily available components and straightforward programming to achieve reliable performance.

Chapter 4

Design Methodology

4.1 Design Process

The design process involved:

- Selecting appropriate components (Arduino, motors, sensors).
- Designing the circuit diagram.
- Writing the control algorithm in Arduino IDE.

4.2 Circuit Diagrams and Descriptions

4.3 Logic Design

The robot's logic includes forward movement, obstacle detection, and direction change based on sensor input.

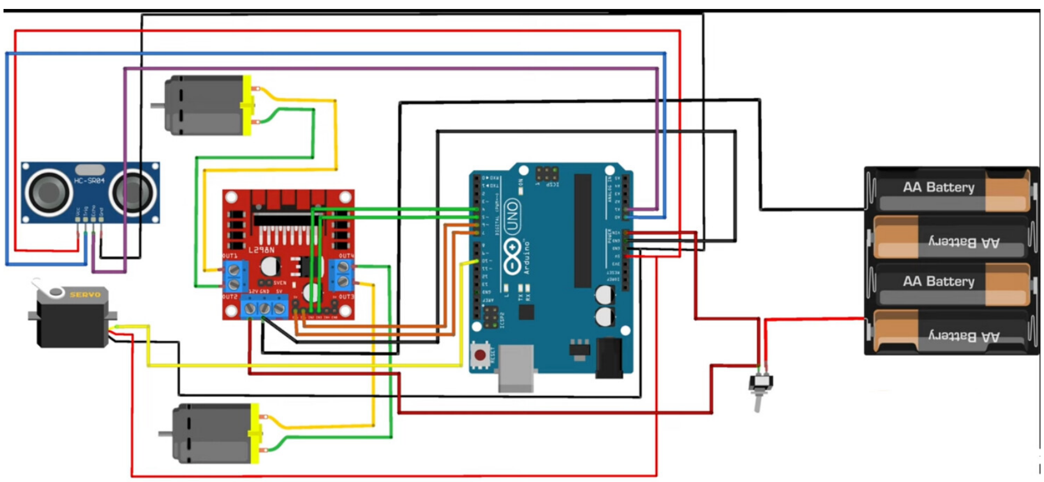

Figure 4.1: Circuit Diagram

Chapter 5

Implementation

5.1 Step-by-Step Implementation

- Assembling the hardware components.
- Writing and uploading the Arduino code.
- Testing the robot's functionality and making adjustments as needed.

5.2 Challenges and Solutions

Challenges included fine-tuning the motor speeds and sensor accuracy. These were addressed by iterative testing and code adjustments.

5.3 Screenshots and Photographs

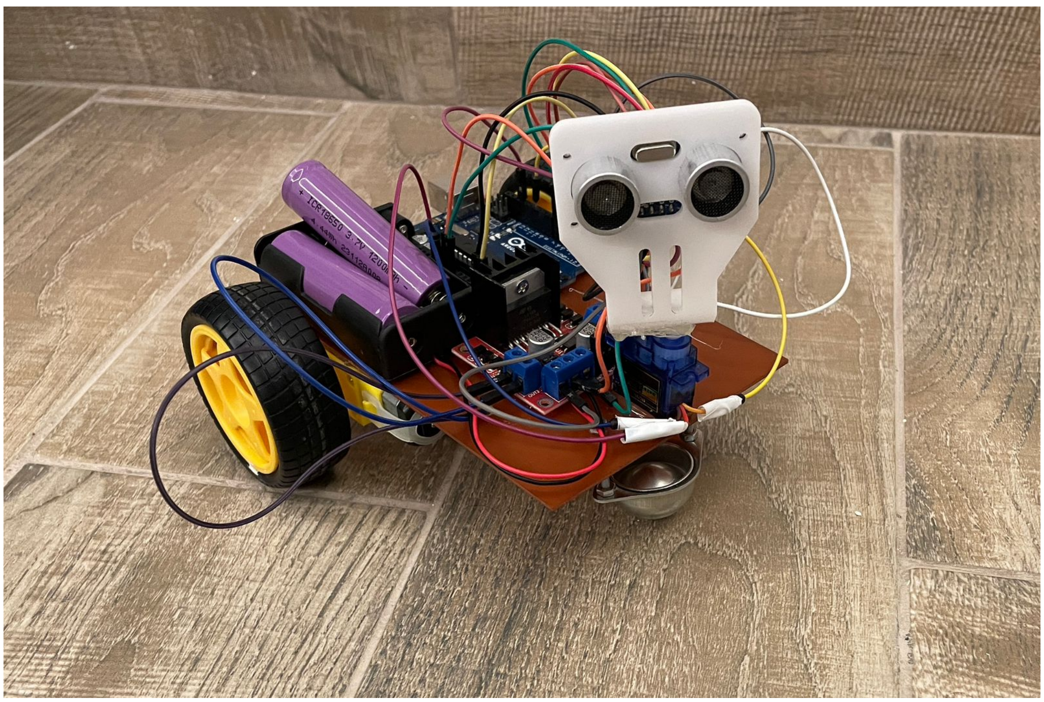

Figure 5.1: Robot during testing

Chapter 6

Testing and Results

6.1 Testing Methods

The robot was tested in various environments to ensure reliable obstacle detection and avoidance.

6.2 Results

The results indicate that the robot can effectively navigate around obstacles, as demonstrated by successful tests.

Chapter 7

Discussion

7.1 Interpretation of Results

The robot's performance met the initial objectives, showing effective obstacle avoidance.

7.2 Assessment

The project demonstrates the feasibility of building a cost-effective autonomous vehicle using basic components.

7.3 Comparison

Compared to initial objectives, the project achieved its goals of autonomous navigation and obstacle avoidance.

Chapter 8

Conclusion

8.1 Summary

This project successfully created an object avoidance robot using Arduino. The major findings include the effectiveness of ultrasonic sensors in obstacle detection and the reliability of the control algorithm.

8.2 Future Work

Future improvements could involve enhancing sensor accuracy and implementing advanced navigation algorithms.

Chapter 9

References

- Arduino documentation: `https://www.arduino.cc/en/Guide/Introduction`
- Ultrasonic sensor datasheets: `https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf`
- Related work: `https://www.youtube.com/arafamicrosystems`

Chapter 10

Appendices

10.1 Code Listings

Listing 10.1: Arduino Code

```
#include <Servo.h>
#define in1 3  //Right motor pole 1
#define in2 2  //Right motor pole 2
#define in3 7  //left motor pole 1
#define in4 4  //left motor pole 2
#define ena 5  //Right motor speed
#define enb 6  //Right motor speed
Servo myservo;  // create servo object to control a servo
int pos = 0;    // variable to store the servo position
const int trigPin = A4;//Trigger Pin of Ultrasonic
const int echoPin = A5;//Echo Pin of Ultrasonic
long duration;//used by ultrasonic Function
int distance,Cdistance,Rdistance,Ldistance;//Cdistance=center Distance
void setup()
{
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  pinMode(ena,OUTPUT);
  pinMode(enb,OUTPUT);
```

```
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  myservo.attach(11);  // attaches the servo on pin 9 to the servo obje
  myservo.write(85);  // Initial Pos of Servo at center
}

void loop()
{
  Cdistance=ultrasonic();//take ultrasonic distance value
  if(Cdistance>30)//if greater than 30 Cm
  {
    forward();//go to forward
  }else
  {
    if(Cdistance<=10&&Cdistance>2)reverse();// if close to object rever
    off();                    //stop robot
    myservo.write(35);        //Turn servo Right
    delay(300);
    Rdistance=ultrasonic();   //get ultrasonic value at right
    myservo.write(135);       // Turn servo Left
    delay(300);
    Ldistance=ultrasonic();   //get ultrasonic value at left
    myservo.write(85);        //return to center pos
    delay(300);
    comparison(Rdistance, Ldistance); //go to Comparasion function
    off();                     // stop after execute Comparison function
    delay(150);
  }
  // reset all variables
  Rdistance=0;
  Ldistance=0;
  Cdistance=0;
}

int ultrasonic(void)//get distance captured by ultrasonic sensor
{
  distance=0;
  // Clears the trigPin
```

```
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microsecon
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance= duration*0.034/2;
  return distance;
}

void comparison(int r, int l)//compare values of right and left Servo P
{
  if(r>25||l>25)
  {
   if(r>l || r==l)
   {
     Tright90();
   }else if(l>r)
   {
     Tleft90();
   }
  }else if(r<25&&l<25)
  {
    Tleft180();
  }
}
void forward()
{
  analogWrite(ena,130);
  analogWrite(enb,130);
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  digitalWrite(in3,LOW);
  digitalWrite(in4,HIGH);
}
```

```
void off()
{
  digitalWrite(in1,LOW);
  digitalWrite(in2,LOW);
  digitalWrite(in3,LOW);
  digitalWrite(in4,LOW);
}
void Tleft90()
{
  analogWrite(ena,150);
  analogWrite(enb,150);
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  delay(650);
}
void Tright90()
{
  analogWrite(ena,150);
  analogWrite(enb,160);
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
  digitalWrite(in3,LOW);
  digitalWrite(in4,HIGH);
  delay(750);
}

void Tleft180()
{
  analogWrite(ena,150);
  analogWrite(enb,150);
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  delay(1500);
}
```

```
void reverse()
{
  analogWrite(ena,140);
  analogWrite(enb,140);
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  delay(450);
}
```

Chapter 11

Acknowledgments

We would like to thank our professor and classmates for their support and guidance throughout this project.