श्रद्धावान् लभते ज्ञानम्

AMRITA
VISHWA VIDYAPEETHAM
UNIVERSITY
Established u/s 3 of UGC Act 1956

Center for Computational Engineering and Networking

Amrita Vishwa Vidyapeetham

2021-2023

A

Term Project Report On

"PROTOTYPE OF SELF DRIVING CAR WITH OBSTACLE DETECTION USING ARDUINO"

Submitted
By

KATIPALLY VIGHNESHWAR REDDY **CB.EN.P2DSC21012**

AMAL THOMAS **CB.EN.P2DSC21001**

ABSTRACT

Obstacle Detection(OD) systems have been used in wide range of different robotics areas and had extraordinary success in minimizing the risk of collisions. It is a critical requirement in building mobile robot systems where they all featured some kind of obstacle detection techniques in order to avoid two or more objects from colliding .In the project we will be dealing with a RC car which will be able to drive on it's own without any external controls and based on the obstacles in the path it will be changing it's path. The Ultrasonic Distance Sensor data will be processed and based on the processed data the Car will be trained to detect the obstacles and find the best alternative path to move forward without any collision. We will be making a maze structure with obstacles and let the car fine the correct way to reach the centre of the maze, this will done to demonstrate the working of the car with obstacle's detection feature.

Contents

List of Figures

Chapter 1

Introduction

1.1 Literature Review

Smart cars are also called wheeled robots, this are modern intelligent control system developed based on many technologies likes computer technology, electronic technology, sensor technology, communication technology, navigation technology, intelligent control technology and automatic control technology [1]. As a necessary function of Smart cars, obstacle avoidance as the basic problem of robot design. The so-called obstacle avoidance is to use the advanced range finding device in front of the smart car [2]. [3] has uses 8-bit AT89C51 to control the DC motor, and the pulse output was used as the control signal of the DC motor along with ultrasonic sensors and infrared photoelectric sensors were used to detect surrounding obstacles and black lines. A new method of measuring by rotating sensors was used in the design, and a better optimal control algorithm was obtained with the time space complexity[4]. The algorithm used in [5], compared with the multi-sensor face avoidance, the control method not only optimizes the motion control of the car, but also reduces the computation amount of the processor chip, helps to improve the stability and reliability of the system.

In [6], an optimal OA method, shortest path without collision, was presented to show that the mathematical model presented was quite accurate and effective.

[7]present an algorithm for performing collision avoidance in mobile robot that is based on ultrasonic and infrared sensors and some other modules, so that it can be easily used in real-time robotic applications.

1.2 Problem Statemen

the car must be able to accurately avoid obstacles encountered on the way in the process of tracking, so there are certain requirements for the detection distance. Taking into account the limitations of the speed of the trolley and the speed of the obstacle avoidance reactor during the obstacle detection process

1.3 Objectives

The car must be able to trace the way by avoiding obstacles and reach the final destination in the Maze.

Chapter 2

Theoretical background

2.1 Servo Motor

2.1.1 Understanding Servo Motor

Servo motor is an electronic device that converts rotational motion into linear motion. It is one of the main components used for industrial motor control. Also known as a servo, it uses a sensor in order to measure position, velocity, and torque of the motor at any given instance. The servo circuitry is built inside the motor unit and has a shaft fitted with a gear that can be positioned as needed.

2.1.2 Working of a Servo

Motion controller sends a signal to a servo and depending on the pulse width modulation (PWM), it rotates at a certain angle. When at rest, the output spline of a servo is 0 degree. The pulse width of 1.5ms makes it rotate by 90 degrees in one direction while that of 1ms makes it rotate 180 degrees backward to the starting position. The position of the output spline is measured by a potentiometer. When the desired position is reached, the power supply is cut and the motor holds that position until the next signal. Proportional operation is an important feature. If the motor stops at 180 degrees while it needs to be at 0 degrees, the movement is swift. However, if it stops closer to 0 degrees, it will move slowly to get there.

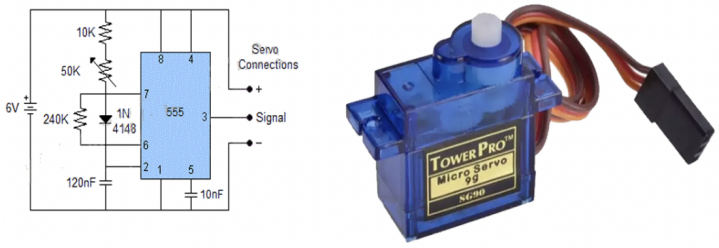

Figure 2.1: Servo Motor

2.1.3 Why robotics implements servo motor?

1. Servo motors are small and powerful. They can be easily programmed and allow almost perfect repeatability of motion.

2. They are easy to control. It is a combination of mechanical and electronic modules and has potentiometer to measure the feedback.

3. It is very easy to make your own servo with the help of gears, a motor, a basic comparator circuit, and some other components.

4. The precision delivered by a servo is incredible because of the ability to control angle based on a feedback system.

5. A servo can be used to make humanoids, bipeds, hexapods, and so on. They are also useful in controlling a camera position,

6. As compared to the stepper motors, they are more reliable and lighter. The only limitation is their range as only a few servos have a 360-degree non-continuous operation. Continuous rotation servos use encoders instead of potentiometer, which is a lined sheet of substance and photo-emitter detector pair.

2.2 Ultrasonic Sensor

The ultrasonic sensor (or transducer) works on the same principles as a radar system. An ultrasonic sensor can convert electrical energy into acoustic waves and vice versa. The acoustic wave signal is an ultrasonic wave traveling at a frequency above 18kHz. The famous HC SR04 ultrasonic sensor generates ultrasonic waves at 40kHz frequency.

Figure 2.2: Ultrasonic Sensor

Typically, a microcontroller is used for communication with an ultrasonic sensor. To begin measuring the distance, the microcontroller sends a trigger signal to the ultrasonic sensor. The duty cycle of this trigger signal is 10µS for the HC-SR04 ultrasonic sensor. When triggered, the ultrasonic sensor generates eight acoustic (ultrasonic) wave bursts and initiates a time counter. As soon as the reflected (echo) signal is received, the timer stops. The output of the ultrasonic sensor is a high pulse with the same duration as the time difference between transmitted ultrasonic bursts and the received echo signal.

The microcontroller interprets the time signal into distance using the following functions. Theoretically, the distance can be calculated using the TRD (time/rate/distance) measurement formula. Since the calculated distance is the distance traveled from the ultrasonic transducer to the object—and back to the transducer—it is a two-way trip. By dividing this distance by 2, you can determine the

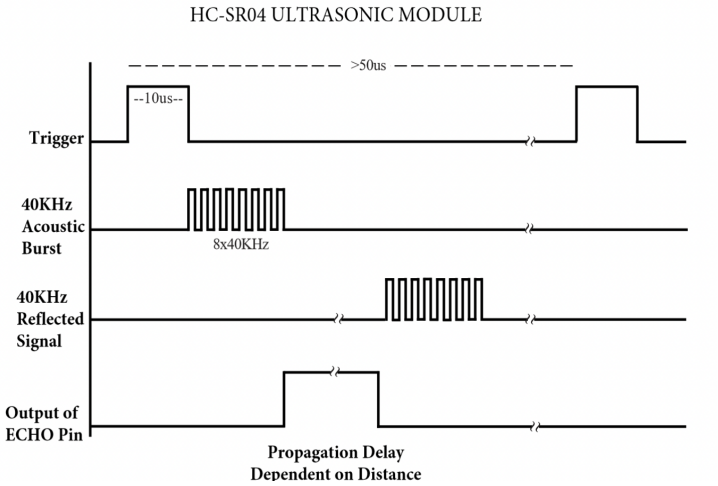

Figure 2.3: Waveforms

actual distance from the transducer to the object. Ultrasonic waves travel at the speed of sound (343 m/s at 20°C). The distance between the object and the sensor is half of the distance traveled by the sound wave.[iv] The following equation calculates the distance to an object placed in front of an ultrasonic sensor.

$$Distance\ (cm) = \frac{echo\ pulse\ width\ (uS)}{58}$$

$$Distance\ (inch) = \frac{echo\ pulse\ width\ (uS)}{148}$$

Figure 2.4: Distance Calculation formula

2.3 MicroController

1. Arduino UNO is a Microcontroller board designed by Arduino.cc in Italy.
2. It uses an Atmega328 Microcontroller which acts as the brain of this board.
3. Arduino Bootloader is installed on Atmega328 which makes it capable to work with Arduino Programming.
4. Arduino is an open-source platform so it has a lot of support from third-party developers.
5. Anyone can design its Libraries for different sensors and modules.

2.3.1 Pin Configuration

1. Arduino UNO has 20 input/output pins.

2. Among these 20 pins, we have 14 digital pins.
3. The remaining 6 pins are analog pins.
4. It also has 6 PWM pins which are used for Pulse Width Modulation.

Arduino UNO supports follow 3 communication protocols:

1. Serial Protocol
2. I2C Protocol
3. SPI Protocol

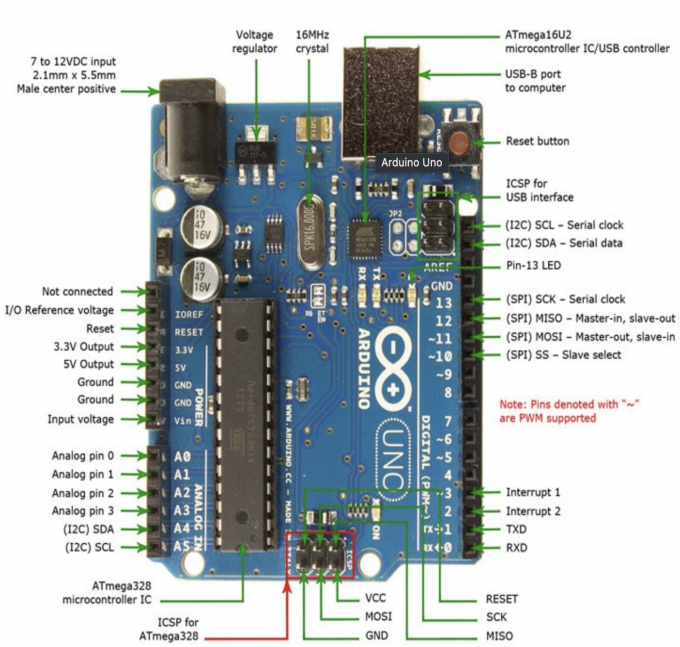

Figure 2.5: Arduino UNO

2.3.2 Arduino UNO Memory Features

1. It has a flash memory of 32Kb.
2. Arduino UNO has SRAM of 2KB.
3. EEPROM memory of UNO is 1Kb.
4. A bootloader of 2Kb is installed so we are left with 30kb Flash memory.

2.3.3 Arduino UNO Applications

1. Embedded Systems
2. Control Systems
3. Robotics

Chapter 3

Methodology

3.1 Components

The list of components uesd in this project are:

1. Ultrasonic Distance Sensor
2. Servo Motor
3. DC motor
4. Motor Shield Shield
5. Ardunio UNO
6. Battery Holder
7. RC Car Body

3.2 Working

The working of this project can be explained with the help of flowchart shown below fig 3.1 The servo motor is set to the 90 position which is front facing of the car, the ultra sonic sensor present on top of the servo motor will start the distance calculation. If the distance is less than 7 cm The car will stop and check for the obstracles at the left side by moving the servomotor to 180 position, if the left side distance is greater than 20 cm the car will turn left and revert back the servo motor to 90 position else (if the left distance is less than 20) it will move the servo position to 0 and will calculate the right side distance. If the right side distance is greater than 20 car will turn right and revert back the servo motor to 90 position, else (if the right distance is less than 20) this means that the car is having obstracle on front, left and right directions so the car will stop.

```
1 //This is the code which got dumped into Ardunio
2
3 #include <Servo.h>
4 Servo myservo;  // create servo object to control a servo
```

```
// twelve servo objects can be created on most boards
int Front_D = 0;
int Left_D = 0;
int Right_D = 0;
int m1a = 6;
int m1b = 5;
int m2a = 4;
int m2b = 3;
int temp;
//int pos = 93;    // variable to store the servo position
const int trigPin = 10;
const int echoPin = 11;
// defines variables
long duration;
int distance;
void setup() {
  Front_D = get_d();
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  pinMode(m1a, OUTPUT);
  pinMode(m1b, OUTPUT);
  pinMode(m2a, OUTPUT);
  pinMode(m2b, OUTPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  myservo.write(83);
  Serial.begin(9600);// Starts the serial communication

  delay(500);
  front();
  delay(200);
}


void loop() {
Front_D = get_d();
if (Front_D < 7) {
  stop1();
  delay(500);
  myservo.write(180);
  delay(1000);
  Left_D = get_d();
  if(Left_D > 20 ) {
    left();
    delay(500);
  }
  else {
   myservo.write(0);
   delay(1000);
   Right_D = get_d();
   if (Right_D < 20 ) {
```

```
      stop1();
      delay(500);
     myservo.write(83);
     Front_D = get_d();
     }
     else {
      right();
      delay(100);
     }
    }
}
}
int get_d() {
   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 microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
return distance;
}
void front() {
      digitalWrite(m1a, HIGH);
       digitalWrite(m1b, LOW);
       digitalWrite(m2a, HIGH);
       digitalWrite(m2b, LOW);
}
void stop1(){
      digitalWrite(m1a, LOW);
       digitalWrite(m1b, LOW);
       digitalWrite(m2a, LOW);
       digitalWrite(m2b, LOW);
}
void right(){

    digitalWrite(m1a, HIGH);
     digitalWrite(m1b, LOW);
     digitalWrite(m2a, LOW);
     digitalWrite(m2b, LOW);
     delay(700);
     front();
     myservo.write(83);
}
```

```
105 void left(){
106
107     digitalWrite(m1a, LOW);
108      digitalWrite(m1b, LOW);
109      digitalWrite(m2a, HIGH);
110      digitalWrite(m2b, LOW);
111      delay(675);
112      front();
113      myservo.write(83);
114 }
```

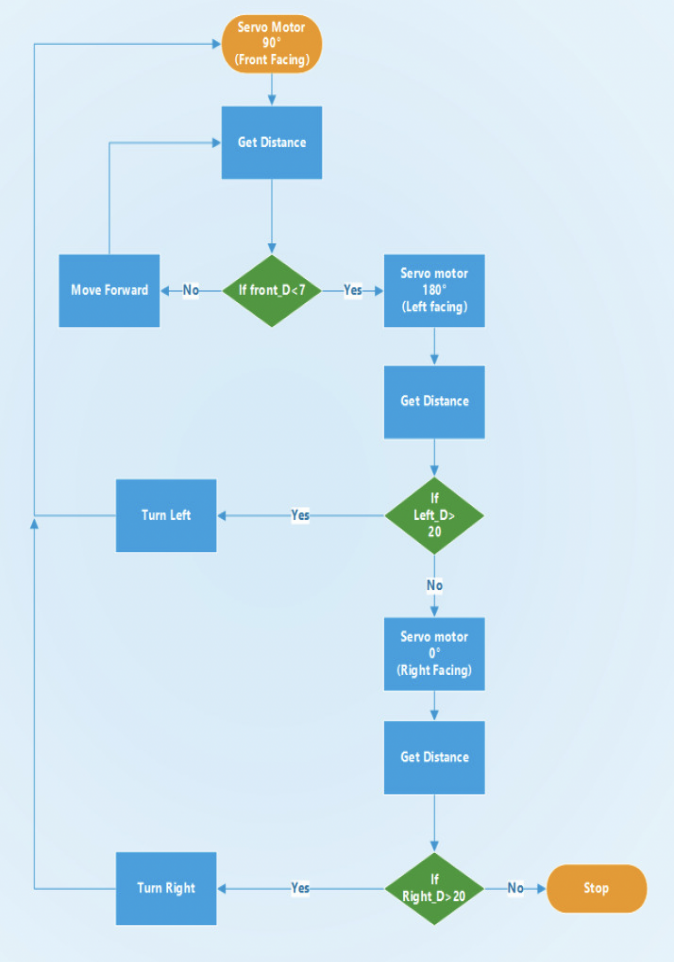

Figure 3.1: FlowChart

The connections for the above components can be seen from the fig 3.2

3.3 Outputs

This are the images of the working model of the self driving car

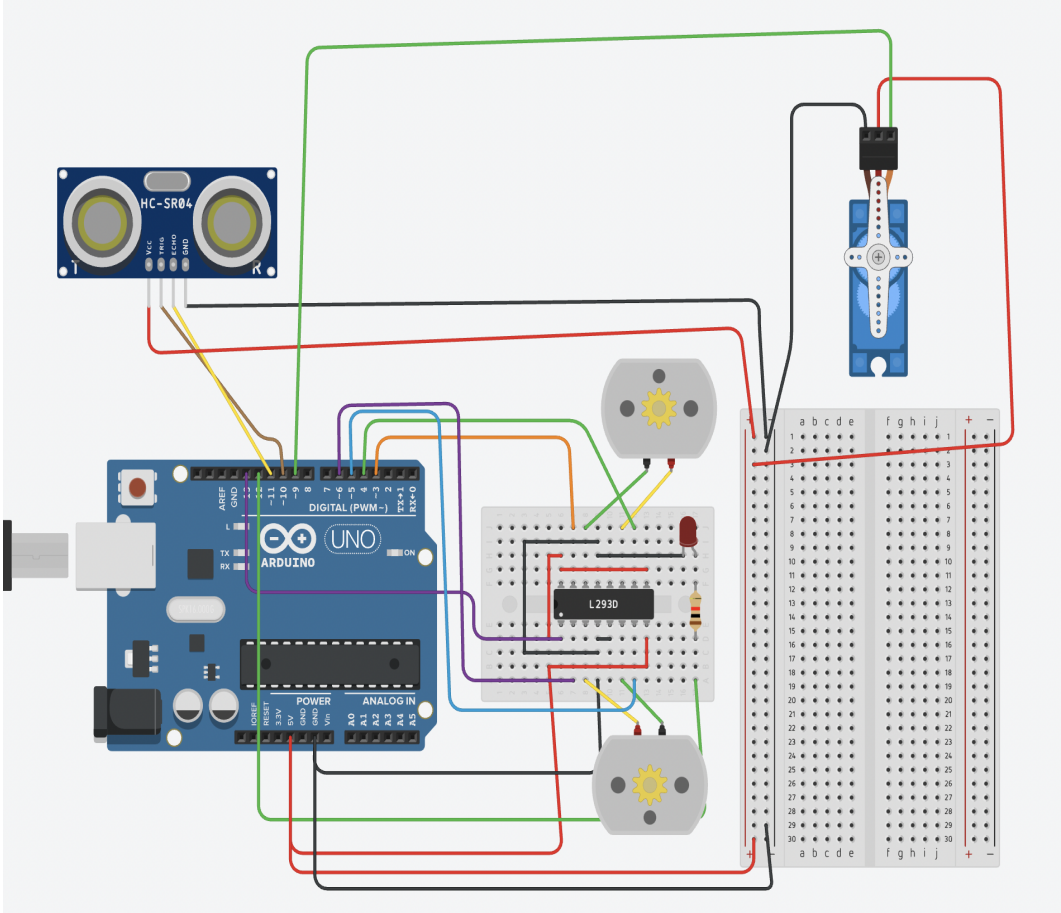

Figure 3.2: Circuit Diagram

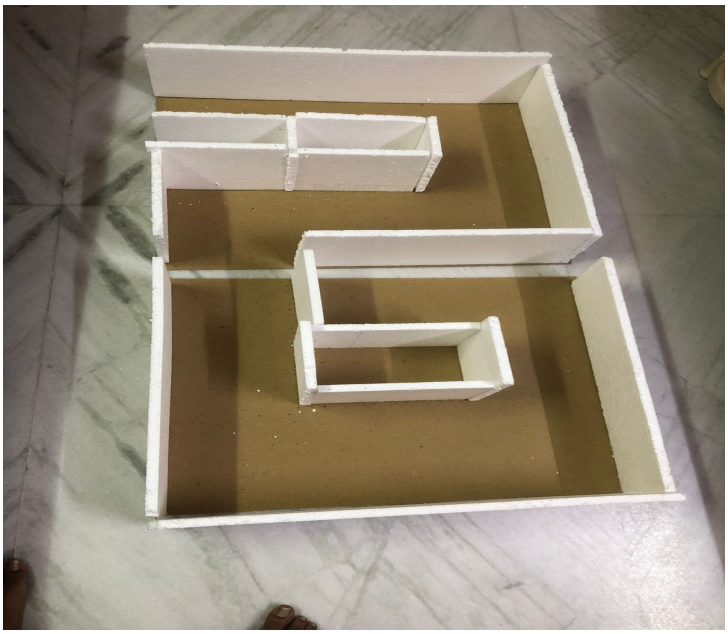

Figure 3.3: Track or Maze with obstracles

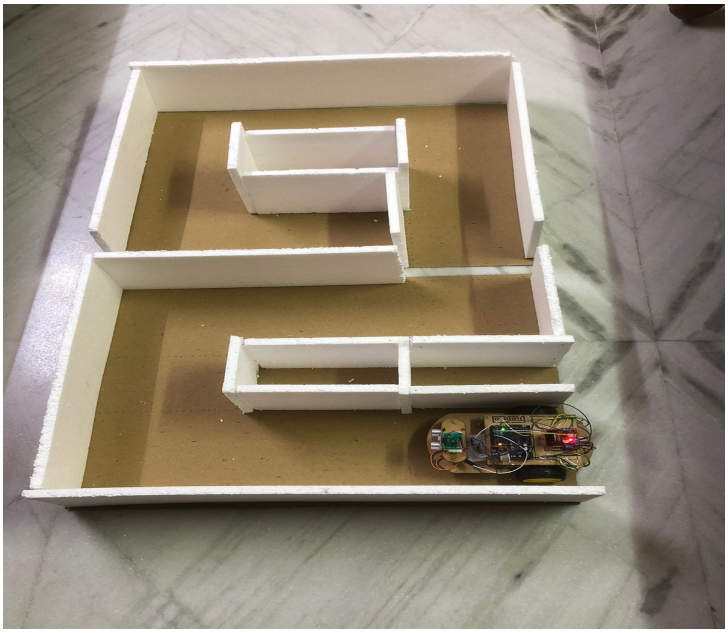

Figure 3.4: Car at the begining of the Maze

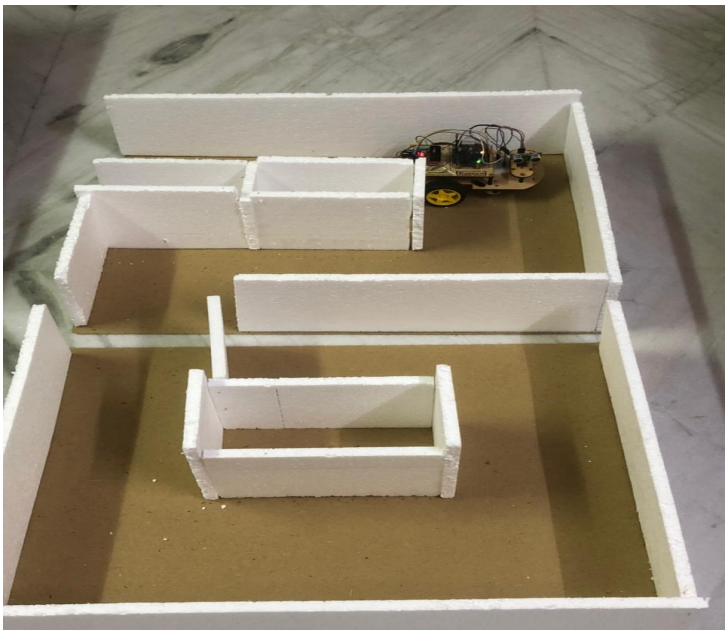

Figure 3.5: Car at the middle of the Maze

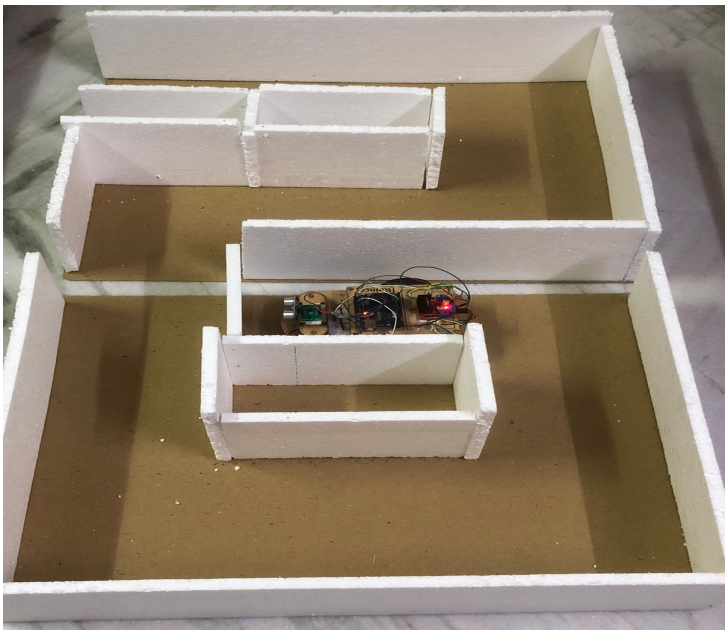

Figure 3.6: Car at the end of the Maze

Chapter 4

Conclusion and Future Scope

4.1 Conclusion

The Car was able to reach the destination from the start point of the Maze which had all possible instructions or Turns, by avoiding the obstracle. So the working model can choose the best path to travel by avoiding the obstracle in three directions which are front, left and right.

4.2 Future Scope

- Reverse option can also be used to make it even more realistic
- By adding camera module can gain addon features like traffic rules and many more

References

[1] Gelasius, R., Komoda, A., Gielen, S. C. Neural network dynamics for path planning and obstacle avoidance. Neural Networks, 8(1) (1995) 125-133

[2] Ziebinski A, Cupek R, Nalepa M. Obstacle Avoidance by a Mobile Platform Using an Ultrasound Sensor[C]// Conference on Computational Collective Intelligence Technologies and Applications. Springer, Cham, 238-248, 2017

[3] J. Nan, "Research on Robot Obstacle Avoidance System Based on Computer Path Planning," 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), 2021, pp. 909-913, doi: 10.1109/ICPECA51329.2021.9362699.

[4] Chaoxu Mu, Haibo He. Dynamic behavior of terminal sliding mode control. IEEE Transactions on Industrial Electronics, vol. 65, no. 4, pp. 3480-3490, 2018

[5] Y. Jin, S. Li, J. Li, H. Sun and Y. Wu, "Design of an Intelligent Active Obstacle Avoidance Car Based on Rotating Ultrasonic Sensors," 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 2018, pp. 753-757

[6] J. -H. Jhou, C. -L. Chen and C. -W. Sun, "Optimal Model of Obstacle Avoidance of Two-Wheeled Robots," 2020 International Symposium on Computer, Consumer and Control (IS3C), 2020, pp. 395-398, doi: 10.1109/IS3C50286.2020.00108.

[7] A. M. Alajlan, M. M. Almasri and K. M. Elleithy, "Multi-sensor based collision avoidance algorithm for mobile robot," 2015 Long Island Systems, Applications and Technology, 2015, pp. 1-6, doi: 10.1109/LISAT.2015.7160181.