A Mini Project Report on

Obstacle Avoidance Car

T.E. - I.T Engineering

Submitted By

Yatish Gharat	**21104050**
Anurag Gupta	**21104109**
Praniv Warungshe	**21104031**

Under The Guidance Of

Ms. Charul Singh

DEPARTMENT OF INFORMATION TECHNOLOGY

A.P.SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic year : 2023-24

CERTIFICATE

This to certify that the Mini Project report on **IR Empowered Robo Car** has been submitted by Yatish Gharat (21104050), Anurag Gupta (211040109) and Praniv Warungshe (21104031) who are a Bonafede students of A. P. Shah Institute of Technology, Thane, Mumbai, as a partial fulfilment of the requirement for the degree in **Information Technology**, during the academic year **2023-2024** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Ms. Charul Singh
Guide

Dr. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)
1.
2.

Place: A.P.Shah Institute of Technology, Thane
Date:

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Ms. Charul Singh** Expressing gratitude towards our HoD, **Dr. Kiran Deshpande**, and the Department of Information Technology for providing us with the opportunity as well as the support required to pursue this project.

TABLE OF CONTENTS

CHAPTER 1

Introduction

The field of robotics has seen significant advancements with the development of autonomous vehicles, particularly in the creation of obstacle avoidance robot cars. These are self-navigating systems designed to detect and maneuver around obstacles in their path. This report focuses on the design and construction of an obstacle avoidance robot car, with a special emphasis on the use of the Raspberry Pi Pico.

The Raspberry Pi Pico is a microcontroller board that serves as the brain of our robot car. It is responsible for processing the data from the sensors, making decisions based on this data, and controlling the motors to steer the car. The choice of the Raspberry Pi Pico is due to its affordability, ease of use, and extensive community support.

The primary function of an obstacle avoidance robot car is to provide a seamless, autonomous navigation system that can detect and avoid obstacles in real-time. This is achieved through a combination of sensors, actuators, and control algorithms. The sensors, such as ultrasonic or infrared sensors, detect the presence and distance of obstacles. The control algorithms then process this data and command the actuators, typically motors, to steer the car away from the detected obstacles.

Despite the challenges in development, the potential applications of obstacle avoidance robot cars are vast. They can be used in automated transportation systems, delivery services, and even in hazardous environments where human intervention is risky. Furthermore, the principles and technologies used in these cars are fundamental to the development of more complex autonomous systems, such as self-driving cars.

In addition to these applications, the obstacle avoidance robot car serves as an excellent educational tool. It provides a hands-on experience in robotics, programming, and hardware interfacing. The process of building and programming the car offers invaluable insights into the workings of sensors, microcontrollers, and motor control.

This report aims to provide a comprehensive understanding of the design, construction, and operation of an obstacle avoidance robot car using a Raspberry Pi Pico. It will delve into the technical details of the sensors, control algorithms, and the interfacing of the various hardware components.

CHAPTER 2

Review of Literature

Sr No.	Title	Author(s)	Year	Key Findings	Relevance to Project
1	Ultrasonic Sensor-Based Obstacle Avoidance System for Autonomous Vehicles	Sharma, R., Singh, A.	2022	Improved obstacle detection and avoidance algorithms leading to enhanced safety and efficiency in autonomous vehicles.	Relevant for understanding ultrasonic sensor application in obstacle avoidance, providing insights into algorithm development for our project.
2	Design and Implementation of an Autonomous Car using Raspberry Pi with Ultrasonic Sensor	Chen, Y., Wang, L.	2023	Successful integration of Raspberry Pi with ultrasonic sensors for obstacle detection and avoidance in a self-driving car. Enhanced navigation accuracy and reliability demonstrated.	Directly applicable to our project as it specifically addresses the integration of Raspberry Pi with ultrasonic sensors for obstacle avoidance in a car.

3	Development of an Intelligent Vehicle Navigation System Using Raspberry Pi Pico and Ultrasonic Sensors	Patel, K., Shah, S.	2021	Efficient real-time obstacle detection and avoidance system implemented using Raspberry Pi Pico and ultrasonic sensors.	Provides insights into utilizing Raspberry Pi Pico with ultrasonic sensors for intelligent navigation, relevant for implementing similar functionalities in our project.

CHAPTER 3

Problem Statement

Designing an Ultrasonic Sensor-Based Obstacle Avoidance Car Using Raspberry Pi Pico: The project aims to develop a compact and intelligent car capable of autonomously navigating through environments while avoiding obstacles. Utilizing Raspberry Pi Pico microcontroller along with ultrasonic sensors, the system will be designed to detect obstacles in its path and maneuver around them effectively. The solution should demonstrate a seamless integration of hardware components and algorithmic intelligence to enable the car to navigate safely and efficiently in real-world scenarios.

a. Motivation

In an era marked by rapid technological advancements, there arises an increasing demand for autonomous systems that can navigate complex environments with precision and reliability. Our ultrasonic sensor-based obstacle avoidance car powered by Raspberry Pi Pico rises to meet this demand by providing a flexible solution applicable across a spectrum of industries. Whether it's optimizing warehouse operations, facilitating medical procedures, or enhancing agricultural processes, our car promises to deliver heightened efficiency and safety. Through its seamless integration of obstacle detection technology and intelligent algorithms, this compact yet powerful car represents a cornerstone in the evolution of autonomous navigation systems, poised to revolutionize various real-world applications where autonomous mobility is paramount.

b. Objectives

- Designing a Compact Object Avoidance Car: Create a hardware and software setup for a compact car utilizing Raspberry Pi Pico and ultrasonic sensors for precise path tracking.
- Implementing Obstacle Detection: Integrate ultrasonic sensors to enable accurate obstacle detection and navigation around encountered obstacles.
- Developing Integrated Algorithms: Design algorithms for seamless integration of line tracking and obstacle avoidance, facilitating smooth navigation.
- Ensuring Robustness: Conduct thorough testing to ensure the system's reliability across diverse environmental conditions.

CHAPTER 4

System Architecture

a. State Diagram/Workflow

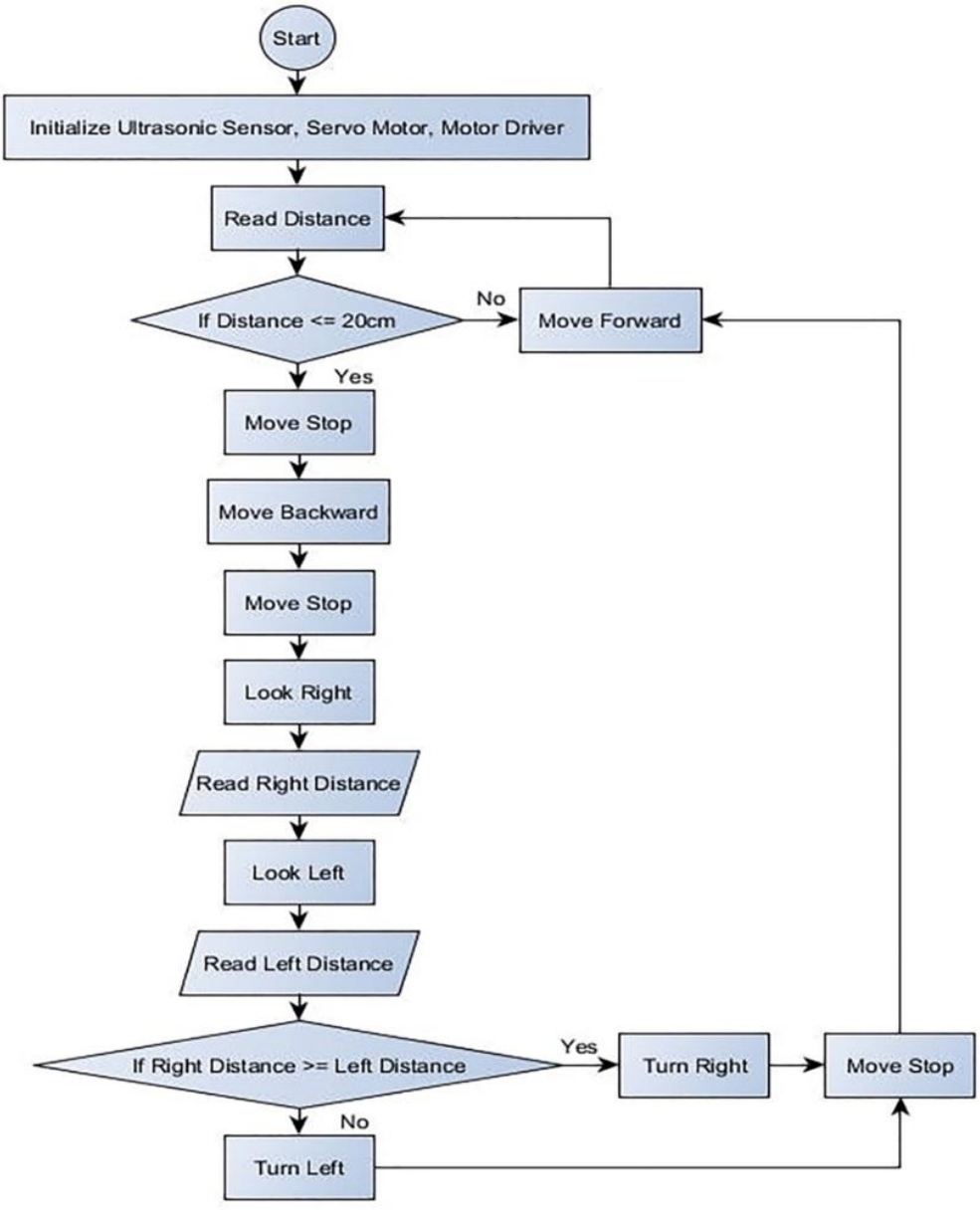


Figure 4.1: Workflow

b. Circuit Diagram

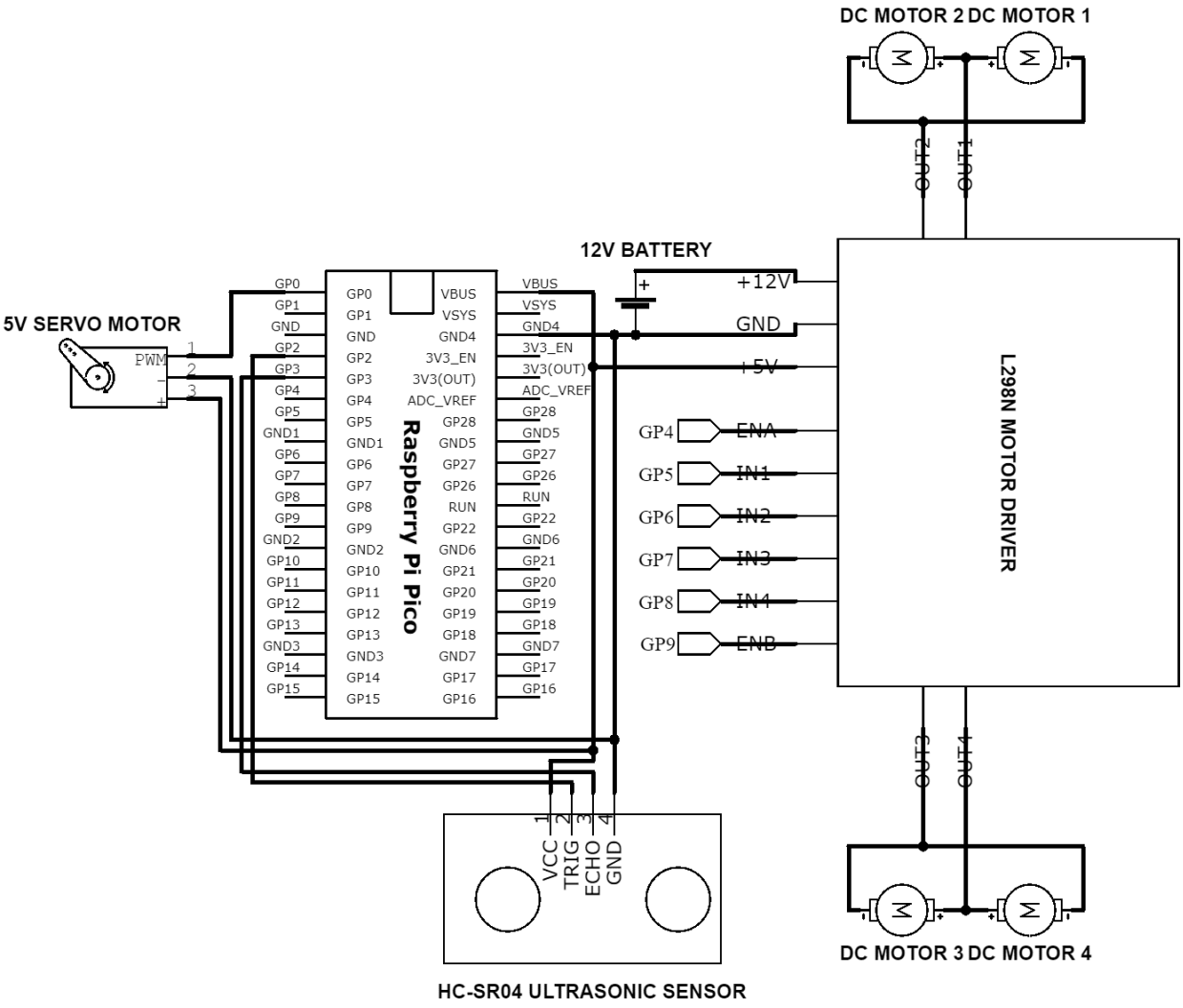


Fig 4.2: Circuit Diagram

CHAPTER 5

Project Timeline

Sr No.	Group Members	Time Duration	Work to be Done
1	Yatish Gharat Anurag Gupta Praniv Warungshe	3rd and 4th week of January	Topic finalization and requirement gathering
2		1st and 2nd week of February	Implementing the circuit design on software
3		End of February and 1st week of February	Connecting the components
4		By the end of March	Final testing and resolving issues(if any)

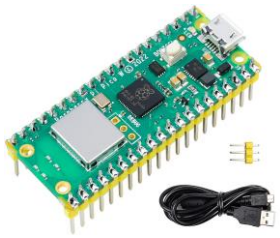
CHAPTER 6

Implementation

a. Hardware and Software Requirements

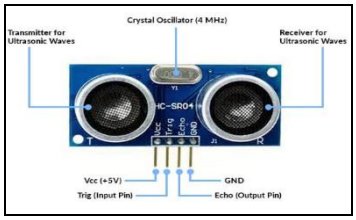
- **Raspberry Pi Pico:**

  The Raspberry Pi Pico is a microcontroller board developed by the Raspberry Pi Foundation. It features the RP2040 microcontroller chip, which is designed by Raspberry Pi and incorporates dual-core ARM Cortex-M0+ processors. The Pico offers a versatile platform for embedded projects, with features such as GPIO pins, ADC, PWM, SPI, I2C, UART, and USB connectivity. It supports various programming languages including MicroPython, C/C++, and CircuitPython, making it accessible for both beginners and advanced users in the embedded systems and DIY electronics communities.



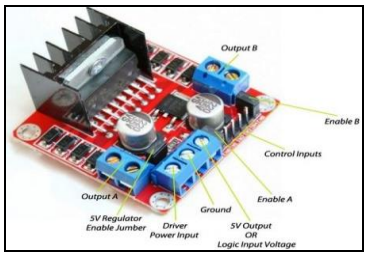
- **Ultrasonic Sensor:**

  A sensor used for measuring distance by emitting ultrasonic waves and calculating the time taken for the waves to bounce back.Typically consists of a transducer that emits ultrasonic waves and a receiver that detects the reflected waves. Distance is calculated based on the time difference between transmission and reception.



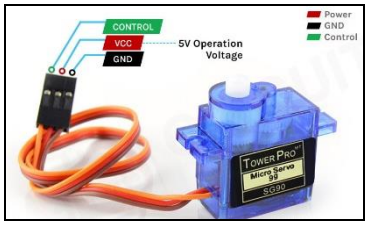
- **Motor Driver:**

  An electronic device or module that controls and manages the operation of an electric motor. It serves as an interface between a microcontroller or other control system and the motor itself, enabling precise control of the motor's speed, direction, and other parameters.Typically includes H-bridge circuits to control motor direction, current sensing for protection, and input pins for receiving control signals from the Raspberry Pi Pico.



- **Servo Motor:**

  A motor that allows precise control of angular position, commonly used for moving robotic arms, sensors, or other components.Contains a DC motor, gears, and a feedback control system that adjusts the motor's position based on input signals. Rotation range is typically limited to a specific angle.



- **Bo Motor:**

  Bo motor (Battery Operated) is a lightweight DC geared motor that produces high torque and rpm at low voltages.Consists of a motor body, shaft, and terminals for electrical connection. Rotation direction and speed are controlled by varying the voltage and polarity applied to the terminals.

- **65mmWheels:**

  Wheels with a diameter of 65mm, used for providing traction and mobility to the robot.Typically made of durable materials like rubber or plastic, with a hub for attachment to the motor shaft and a tread pattern for grip.

- **Switch:**

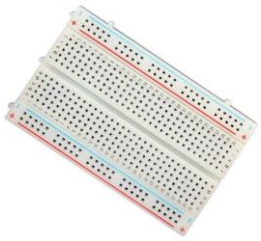
  A mechanical or electronic device used to interrupt or divert the flow of electrical current, allowing for safe startup and shutdown procedures.

- **Jumper Wires:**

  Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

- **Breadboard:**

  A breadboard is a construction base used for building semi-permanent prototypes of electronic circuits. It allows you to place components and connections on the board to make circuits without soldering. The holes in the breadboard hold onto parts or wires where you put them and electrically connect them inside the board. Breadboards are reusable, making them popular for prototyping and educational purposes1. However, they are not ideal for long-term use or high-frequency circuits. They come in many sizes to fit projects large and small.



Hardware	Quantity	Cost
Raspberry Pi Pico	1	Rs 650/-
Ultrasonic Sensor	1	Rs 100/-
Breadboard	1	Rs 100/-
Motor Driver	1	Rs 150/-
Servo Motor	1	Rs 140/-
Bo Motor	4	Rs 400/-
5mm Wheels	4	Rs 160/-
3.7V li-ion Battery	4	Rs 400/-
Switch	1	Rs 10/-
Jumper Wires	20-30	Rs 50/-
Total		Rs 2210/-

a. Principle and Working:

- **Principle:**

Our project centers on the development of an object-avoiding robot car using Raspberry Pi Pico. The fundamental principle underlying our design involves the seamless integration of hardware and software components to enable autonomous navigation while effectively detecting and circumventing obstacles in real-time.

At its core, our robot car employs ultrasonic sensors strategically positioned to continually scan the environment for potential obstructions. When an obstacle is detected within the predefined detection range, the system initiates a dynamic deviation maneuver. This maneuver entails dynamically adjusting the motor rotations to steer the car away from the obstacle while preserving its intended path.

Key to the success of our design is the implementation of intelligent algorithms capable of processing sensor data in real-time and making rapid decisions regarding obstacle avoidance maneuvers. These algorithms leverage the capabilities of the Raspberry Pi Pico microcontroller to analyze sensor readings, calculate optimal deviation trajectories, and adjust motor controls accordingly.

Furthermore, our design prioritizes the seamless integration of obstacle detection and navigation functionalities, ensuring that the robot car can swiftly and efficiently navigate through dynamic environments while avoiding collisions with obstacles. By employing robust hardware components, sophisticated software algorithms, and meticulous system integration, our project aims to demonstrate the feasibility and effectiveness of autonomous object avoidance in real-world applications..

- **Working:**

To ensure the optimal performance of our object-avoiding robot car, a systematic wiring arrangement is implemented. The Raspberry Pi Pico serves as the central control unit and is connected to various components as follows:

- Ground (GND) Connection:The GND pin (Pin 34) of the Raspberry Pi Pico is linked to multiple components, including the ultrasonic sensor, motor drivers, and sensors, establishing a common ground reference for stable operation.

- Power Supply Connection:The 5-volt output pin (Pin 36) of the Raspberry Pi Pico supplies power to components like the ultrasonic sensor, motor drivers, and sensors, ensuring consistent voltage supply for their operation.

- Power Supply for Motors:The positive terminal of the power supply is connected to the VCC pin of the motor drivers, while the negative terminal is connected to the GND (Ground) of the motor drivers, providing power to the motors.

- Motor Control Pins:The control pins of the motor drivers, including input pins for motor control and enable pins, are connected to specific GPIO pins of the Raspberry Pi Pico. For example, IN1 (Pin 11), IN2 (Pin 12), IN3 (Pin 13), IN4 (Pin 15), and EN (Enable Pin - Pin 16) are connected to control the 4WD motors.

- Ultrasonic Sensor Connection:The trigger (Trig - Pin 2) and echo (Echo - Pin 3) pins of the ultrasonic sensor are connected to dedicated GPIO pins of the Raspberry Pi Pico for distance measurement and obstacle detection.

- Servo Motor Connection:The signal pin of the servo motor is connected to a GPIO pin of the Raspberry Pi Pico. For example, if using Pin 18 for servo control, the signal pin of the servo motor would be connected to this pin.

This systematic wiring arrangement ensures proper communication and coordination between the Raspberry Pi Pico and the various hardware components, facilitating seamless operation of the robot's object-avoidance capabilities while navigating through its environment.

Code:

```
from machine import Pin,PWM #importing PIN and PWM
import time #importing time
import utime

# Defining motor pins
motor1=Pin(14,Pin.OUT)
motor2=Pin(13,Pin.OUT)
motor3=Pin(12,Pin.OUT)
motor4=Pin(11,Pin.OUT)
# Defining enable pins and PWM object
enable1=PWM(Pin(15))
enable2=PWM(Pin(10))

# Defining    Trigger and Echo pins
trigger = Pin(16, Pin.OUT)
echo = Pin(17, Pin.IN)

# Defining    Servo pin and PWM object
servoPin = Pin(0)
servo = PWM(servoPin)
duty_cycle = 0 # Defining and initializing duty cycle PWM

# Defining frequency for servo and enable pins
servo.freq(50)
enable1.freq(1000)
enable2.freq(1000)

# Setting maximum duty cycle for maximum speed
```

```
enable1.duty_u16(16000)
enable2.duty_u16(16000)

# Forward
def move_forward():
    motor1.low()
    motor2.high()
    motor3.high()
    motor4.low()
    print("forword")

# Backward
def move_backward():
    motor1.high()
    motor2.low()
    motor3.low()
    motor4.high()
    print("backword")

#Turn Right
def turn_right():
    motor1.low()
    motor2.high()
    motor3.low()
    motor4.high()
    print("right")

#Turn Left
def turn_left():
    motor1.high()
```

```
        motor2.low()
        motor3.high()
        motor4.low()
        print("left")

#Stop
def stop():
        motor1.low()
        motor2.low()
        motor3.low()
        motor4.low()
        print("stop")

# Defining function to get distance from ultrasonic sensor
def get_distance():
    trigger.low()
    utime.sleep_us(2)
    trigger.high()
    utime.sleep_us(5)
    trigger.low()
    while echo.value() == 0:
            signaloff = utime.ticks_us()
    while echo.value() == 1:
            signalon = utime.ticks_us()
    timepassed = signalon - signaloff
    dist = (timepassed * 0.0343) / 2
    print(dist)
    return dist
#Defining function to set servo angle
def setservo(angle):
```

```
    duty_cycle = int(angle*(7803-1950)/180) + 1950
    servo.duty_u16(duty_cycle)

setservo(90)
try:
    while True:
        distance=get_distance() #Getting distance in cm

        #Defining direction based on conditions
        if distance < 15:
            stop()
            move_backward()
            time.sleep(1)
            stop()
            time.sleep(0.5)
            setservo(30) #Servo angle to 30 degree
            time.sleep(1)
            right_distance=get_distance()
            print("right distance: ",right_distance)
            time.sleep(0.5)
            setservo(150) #Servo angle to 150 degree
            time.sleep(1)
            left_distance=get_distance()
            print("left dist: ",left_distance)
            time.sleep(0.5)
            setservo(90)
            if right_distance > left_distance:
                turn_right()
                time.sleep(0.5)
                stop()
```

```
                else:
                    turn_left()
                    time.sleep(0.5)
                    stop()
            else:
                move_forward()
                time.sleep(0.5)

except Exception as e:
    print(e)
    stop()
```

CHAPTER 7

Conclusion

In conclusion, our object-avoiding robot car project has laid a solid foundation for future advancements in autonomous robotics. By integrating additional sensors, optimizing path planning algorithms, and exploring new capabilities such as autonomous exploration and cloud integration, we can enhance the robot's functionality and versatility. Furthermore, collaboration between multiple robot cars and the development of user-friendly interfaces will further extend the project's potential for real-world applications. With ongoing innovation and refinement, our project has the potential to contribute significantly to fields such as search and rescue, environmental monitoring, and warehouse automation, paving the way for a future where autonomous robots play a crucial role in navigating complex environments and solving real-world challenges.

CHAPTER 8

Future Scope

Moving forward, there are exciting avenues for enhancing our object-avoiding robot car project. Integrating additional sensors like LiDAR or infrared cameras can improve obstacle detection, enhancing navigation safety and efficiency. Optimizing path planning algorithms can further enhance navigation by dynamically rerouting around obstacles and prioritizing paths. Additionally, expanding the robot's capabilities to autonomously explore and map unknown environments using SLAM algorithms can enable deployment in diverse applications, from search and rescue missions to environmental monitoring. Integrating the robot with IoT devices and cloud services can enable remote monitoring, control, and data analysis, enhancing its functionality. Exploring collaboration between multiple robot cars and developing user-friendly interfaces can further extend the project's potential for real-world applications, empowering users to navigate complex environments seamlessly.

CHAPTER 9

References

[1] D. Floreano and J. Urzelai. "Evolutionary Robots with Online Self-Organization and Behavioral Fitness

[2] Oussama Khatib. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots

[3] Marija Seder. "Hierarchical Path Planning of Mobile robots in Complex Indoor Environments

[4] https://circuitdiagrams.in/obstacle-avoidance-robot-using-raspberry-pi/

[5] https://forums.raspberrypi.com/viewforum.php?f=143

[6] https://www.youtube.com/watch?v=hyFZKBXhAwI&t=2s&ab_channel=Robojax