PROJECT TITLE

EYEVOCE

(for visually challenged)

SUBMITTED BY:

21BCE8206 Shaik Farukh

21BCE9920 Chinnam Sri Harshitha

21BCE9921 Godugu Peerambe

21BCE9924 Avanigadda Sadhana

21BCE9931 Amasala Venkata Siva Sai Likhitha

21BCE9938 Bhimavarapu Tanu Sree Sai Durga

UNDER THE GUIDANCE OF:

Prof. Dr. Santosh Kumar Sahu

VELLORE INSTITUTE OF TECHNOLOGY,

AMARAVATI, AP-522 237, INDIA.

Abstract

In this project, we will be making a prototype of a working model that can help a visually challenged to detect the obstacles ahead. This will be enabled by object detection and distance detection features. We propose a Eyevoce with a obstacle detection system to help the blind in recognizing the objects in front of them. This system detects the obstacles within the range of 15 cm and gives an alarm as the person moves closer to the object. The result of the detection is informed to the blind person through a buzzer attached to the mobility cane. The proposed system was designed to be used in real-time and is equipped with a raspberry pico board, an ultrasonic sensor, a voice module and a speaker. The ultrasonic sensor attached to the lower end of the cane detects the objects that are around it within the range by sending the ultrasonic sound waves and sends signals to raspberry pico if any objects is detected. Thereby, the raspberry pico run the code stored in it. If the object detected is within the range of 15cm, it sends a signal to the voice module to give an alert. The voice module helps to converts the signal from the raspberry pico to the voice commands and sounds through the speaker. The voice can be given in our preferred language so that whenever it gets the obstacle signal, it says that there's an obstacle ahead. We used python for the model along with PIN and PWM libraries to get an artificial analog output.

INDEX

INTRODUCTION

The challenges faced by blind people in their everyday lives are not well understood. Blind people confront a number of visual challenges every day – from reading the label on a frozen dinner to figuring out if they're at the right bus stop. While many tools have been introduced to help address these problems using computer vision and other sensors (talking OCR, GPS, radar canes, etc.), their capabilities are dictated as much by the state-of-the-art in technology as they are by real human problems. A deeper understanding of the questions that blind people would like to ask in their day-to-day lives may help to direct innovation to solve them.

For visualising the Visual world to break and elucidate, which explains computer vision in computer technology. In classifying the objects' accuracy, machines use deep learning models17 and digital images such as cameras and videos. Visually impaired people find it hard to recognize the smallest detail with healthy eyes. The main problem with blind people is how to navigate their way to wherever they want to go. Such people need assistance from others with good eyesight.

This device allows us to identify and locate objects by sensor. With this kind of identification and localization, object detection can be used to count objects in a scene and determine and track their precise locations, all while accurately labelling them. It makes the work of Blind easy, efficient and reliable by sending wireless Voice based feedback whether the particular object is either too close to him or is it at a safer distance. It is not only useful for blind people but also for those who are facing issues like blurriness. This device is easy to carry and maintain as it is a part of mobility cane and one can carry this anytime.

Background:

In 1950s, demonstrations have already started in computer vision to identify the edges and align the simpler objects with falling under categories such as circles and squares by the techniques of first neural networks. In 1970s, Optional character recognition came into existence of computer vision, handwritten data on its primary trading tool. The illustrated data mainly used for the blind as a development.

Smart obstacle detector helps blind people in moving and allowing them to perform their work easily and comfortably. In normal cane/stick, detection is done by the sensor. However, it is not much efficient because the blind person does not know what type of things or objects come in front of him, what is the size of that object and how far is he from the object? So, it is difficult for blind person to move here and there.

We detect the object by the means of sensor. Here we have used ultrasonic sensor for efficient and fast detection of objects, stick measure distance between objects and stick by Ultrasonic sensor. Moreover, we have connected the voice module with Ultrasonic sensor. When objects come in range of ultrasonic sensor then we are intimated with a sound that conveys that there is object Infront of us. We can also use solar chip for operating the stick which is definitely included in the future enhancement of our research-based project.

Problem Definition:

The primary objective of this project is to help the blind in recognizing the object or obstacle by means of the device which contains an ultrasonic sensor and a voice module which identifies the object and intimates them to be careful. This acts as a personal assistant for the blind while travelling. We can add more features like intimation in our preferred language and notification is sent to our mobiles. We can also know the emotions of the opposite person by using this detector in further updated versions.

Objectives:

1. Help the blind to recognize the obstacle.
2. Object detection.
3. Prevents accidents.
4. Not only for blind people but also for the visually impaired.

Procedure:

- ➢ First, we have to purchase the required materials for the product.
- ➢ The required products for our product are:
    - Raspberry Pi Pico board
    - Ultrasonic sensors
    - Jumper wires
    - Voice module
    - External speaker
    - Cardboard (for base)
    - Stick

1.Raspberry Pi Pico board

Fig(a)

The Pico is a low-cost, high-performance microcontroller board built around the Raspberry Pi RP2040 chip. The Pico features flexible digital interfaces and can be easily programmed over USB using C/C++ or Micro-Python, thanks to a comprehensive SDK with software examples and full documentation.

2. Ultrasonic sensors

HC-SR04 Ultrasonic Sensor has two eyes like projects in the front which forms the transmitter and Receiver. The ultrasonic sensor uses sonar to determine the distance to an object like bats or dolphins do. This module is a transmitter, a receiver, and a control circuit in one single pack!! It offers excellent range accuracy and stable readings in an easy-to-use package.

Fig(b)

3.Voice Module

The Recording Module Voice Board is the real easy way to add Voice Recording (and Playback) to your project. The Module can be operated directly by using the 3 Push-Buttons or with every microcontroller (ex. Arduino). A microphone is implemented directly on the board, and you can connect any 8 Ohm Speaker. Your recordings are saved even without power due to the non-volatile storage on the recording module.

Fig(c)

4.Jumper Wires

A jump wire (also known as jumper, jumper wire, DuPont wire) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

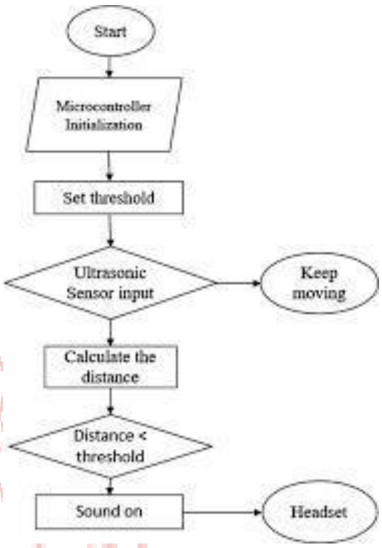
Fig(d)

5.USB Cable

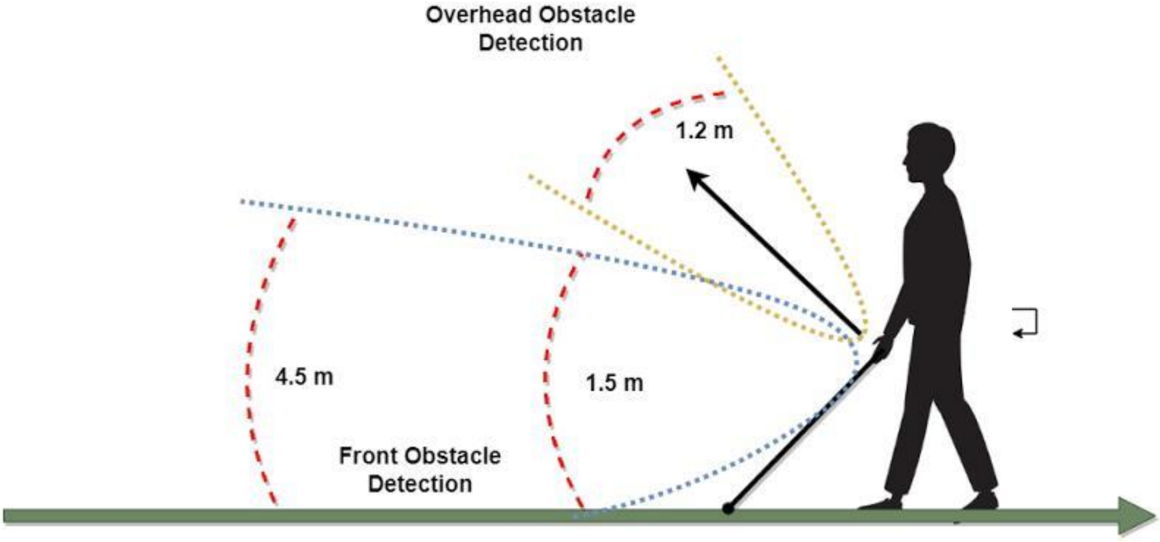
The term USB stands for "Universal Serial Bus". USB cable assemblies are some of the most popular cable types available, used mostly to connect computers to peripheral devices such as cameras, camcorders, printers, scanners, more.

Fig(e)

- Now, attach the cardboard to the stick for the base to attach all the components to the stick.
- Now attach the pico board *Fig(a)*, voice module *Fig(c)*, and external speaker to the cardboard with the help of glue.
- Now, we have to attach the ultrasonic sensors *Fig(b)* for the three sides of the stick in order to detect the object.
- Now, with the help of the jumper wires connect the ultrasonic sensors with the raspberry pi pico pins.
- Do the same with the voice module and external speaker(connect with the jumper wires *Fig(d)*
- Make sure all the wires are connected properly to all the pins of the respective boards.
- Now, connect the USB cable *Fig(e)* to the pico board.
- The device is ready to use.



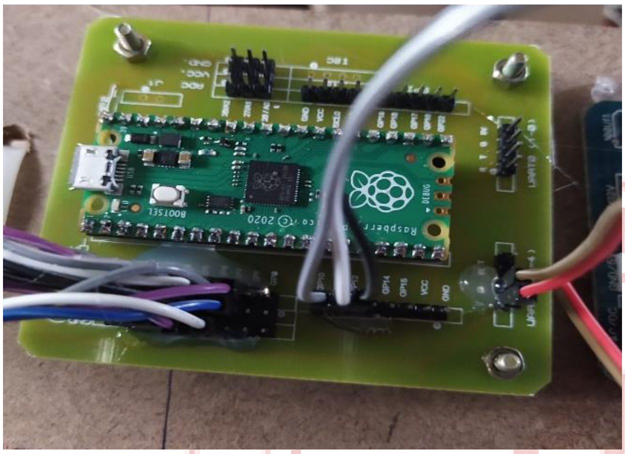
Fig(f).: Working flowchart



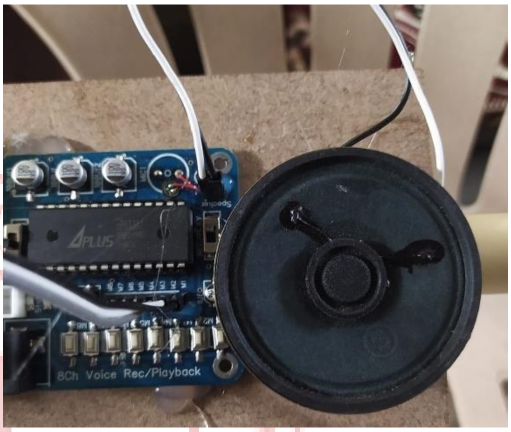
Fig(g).: a person using the Eyevoce cane

Results:

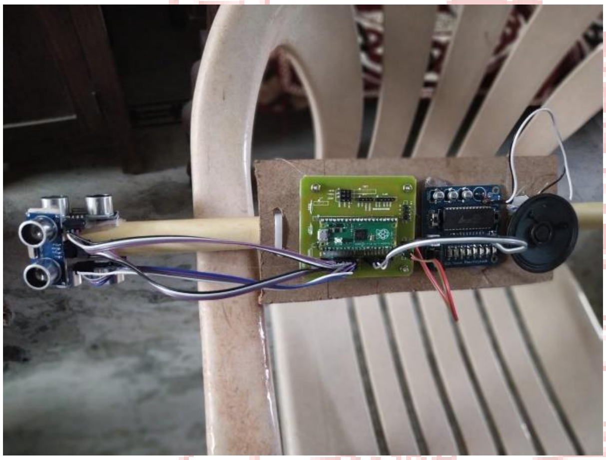
The following are few pictures of the final device after ensembling all the components:



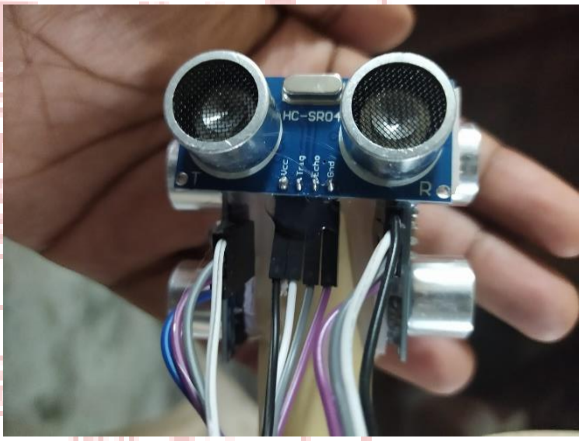
Fig(h)



Fig(i)

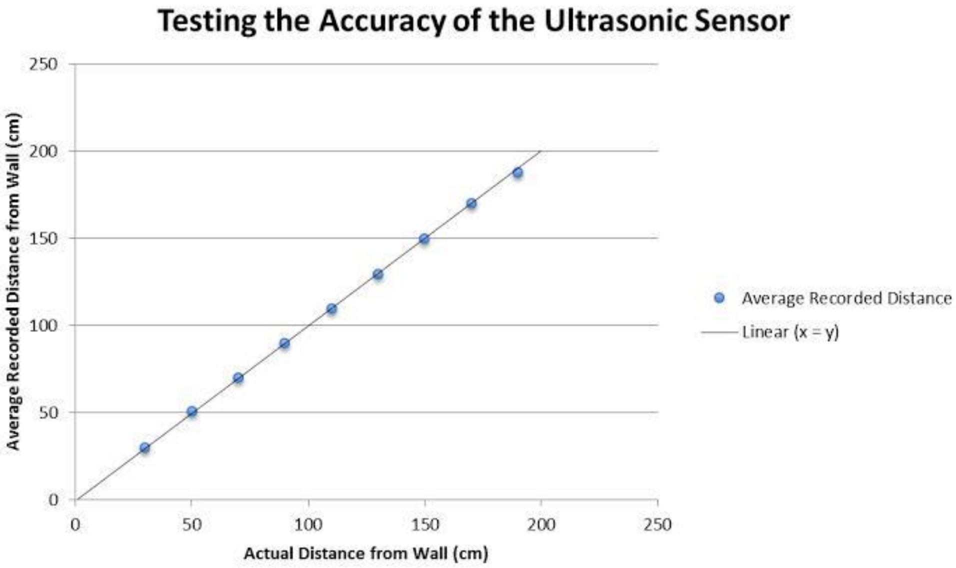


Fig(j)



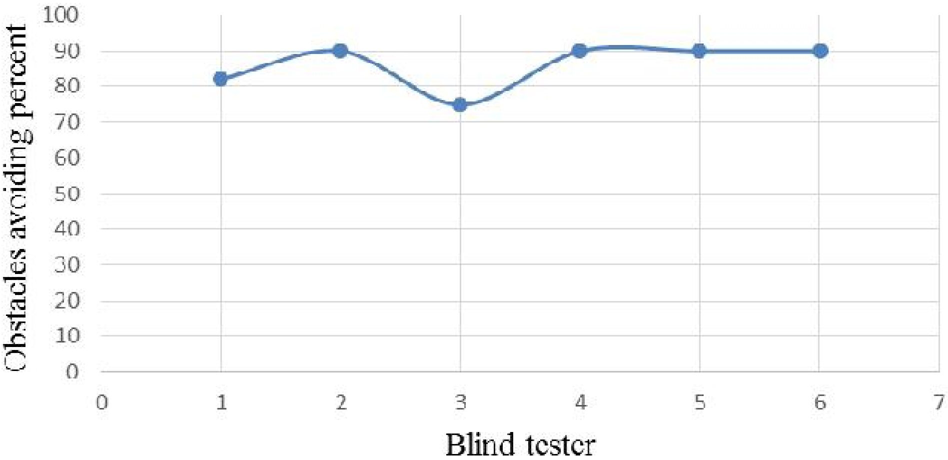
Fig(k)

From all the instances of the experiment, We get the accuracy stats of the device. As the main function of this device is to detect objects, the working of the Ultrasonic sensor matters the most. We recorded the average recoded distance by plotting the graph using Actual Distance from the wall/object(in cm) on x-axis against Average Recorded Distance from the wall/object(in cm) as shown in the *Fig(l)*.



Fig(l).:Testing the accuracy of the Ultrasonic Sensor

We performed experiments with the help of group of users in which they test the prototype to get the real experience of how the device works. We recorded the average obstacle avoidance percentage by plotting a graph of the users on x-axis and obstacle avoidance percentage on the y-axis *Fig(m)*. The average obstacle avoiding percentage is about 85.33% *(calculated from the values recorded in the experiment)*.



Fig(m).: Average obstacle avoidance percentage
(Calculated on the basis of six users experience)

Conclusion and Future Scope:

The possibilities are endless when it comes to future use cases for object detection. Herewith are some of the main useful applications of object detection: Vehicle’s Plates recognition, self-driving cars, Tracking objects, face recognition, medical imaging, object counting, object extraction from an image or video, person detection. This device is mainly focused to guide the visually paired, we can also modify this device according to the user requirements. We can also add the features like facial recognition to recognize the people around the cane using a camera attached to the device, can also add google assistant to guide the way using maps, detecting the expiry dates and name of the medicine

References:

Visited for the reference About [On]:

Working of Raspberry pi pico

https://www.instructables.com/Object-Detection-on-Raspberry-Pi/ [10-02-2023]

https://medium.datadriveninvestor.com/object-detection-with-raspberry-pi-and-python [10-02-2023]

Interfacing Ultrasonic sensor with raspberry pi pico

https://youtu.be/OTBIXnzcI34 [12-02-2023]

Object detection for blind

https://www.researchgate.net/publication/343682695_Smart_Blind_Stick_For_Obstacle_Detection_and_Navigation_System [10-02-2023]

https://techatronic.com/smart-blind-stick-using-arduino-and-ultrasonic-sensor/ [11-02-2023]

https://www.electronicsforu.com/electronics-projects/new-facial-recognition-smart-glasses-for-visually-challenged-person [11-02-2023]

Object detection and current trends

https://www.analyticsvidhya.com/blog/2022/03/a-basic-introduction-to-object-detection/ [15-02-2023]

https://www.frontiersin.org/articles/10.3389/frobt.2015.00029/full [16-02-2023]

Appendix:

```
from machine import Pin,PWM #importing PIN and PWM
import time #importing time
import utime
# Defining motor pins
v1=Pin(10,Pin.OUT)
v2=Pin(11,Pin.OUT)
v3=Pin(12,Pin.OUT)
v1.value(0)
v2.value(0)
v3.value(0)
# Defining  Trigger and Echo pins
trigger = Pin(3, Pin.OUT)
echo = Pin(2, Pin.IN)
trigger1 = Pin(5, Pin.OUT)
echo1 = Pin(4, Pin.IN)
trigger2 = Pin(7, Pin.OUT)
echo2 = Pin(6, Pin.IN)

# 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
    return dist
def get_distance1():
    trigger1.low()
    utime.sleep_us(2)
    trigger1.high()
    utime.sleep_us(5)
    trigger1.low()
    while echo1.value() == 0:
        signaloff = utime.ticks_us()
    while echo1.value() == 1:
        signalon = utime.ticks_us()
    timepassed = signalon - signaloff
    dist1= (timepassed * 0.0343) / 2
    return dist1
def get_distance2():
    trigger2.low()
    utime.sleep_us(2)
    trigger2.high()
    utime.sleep_us(5)
    trigger2.low()
    while echo2.value() == 0:
        signaloff = utime.ticks_us()
    while echo2.value() == 1:
        signalon = utime.ticks_us()
    timepassed = signalon - signaloff
    dist2= (timepassed * 0.0343) / 2
    return dist2
```

```python

while True:
    distance=get_distance() #Getting distance in cm
    print(distance)
    distance1=get_distance1() #Getting distance in cm
    print(distance1)
    distance2= get_distance2() #Getting distance in cm
    print(distance2)

    #Defining direction based on conditions
    if distance < 15:
        v1.value(1)
        time.sleep(2)
        v1.value(0)

    else:
         v1.value(0)

    if distance1< 15:
         v2.value(1)
         time.sleep(2)
         v2.value(0)

    else:
        v2.value(0)
        time.sleep(0.2)

    if distance2< 15:
        v3.value(1)
        time.sleep(2)
        v3.value(0)

    else:
        v3.value(0)
        time.sleep(0.2)


```

The code used for this device is a script written in MicroPython language for controlling three ultrasonic sensors and three motors. The script starts by importing the necessary libraries: **Pin**, **PWM**, **time**, and **utime**. Then, it defines three output pins for the three motors (**v1**, **v2**, and **v3**) and three sets of input and output pins for the ultrasonic sensors (**trigger** and **echo**, **trigger1** and **echo1**, and **trigger2** and **echo2**).

After that, there are three functions (**get_distance()**, **get_distance1()**, and **get_distance2()**) that take the distance measurements from the corresponding ultrasonic sensors using the **trigger** and **echo** pins. These functions return the distance values in centimeters.

Finally, there is an infinite **while** loop that continuously reads the distance measurements from the three ultrasonic sensors and based on the values, sets the direction of the three motors. If the distance value for a particular sensor is less than 15cm, then the corresponding motor (**v1**,

v2, or **v3**) will be turned on for 2 seconds using the **value()** method and then turned off. If the distance value is greater than or equal to 15cm, the corresponding motor will be turned off using the **value()** method.

Note that there is a **time.sleep(0.2)** after turning off motors **v2** and **v3**, but not for motor **v1**. It is not clear why this is the case, but it may be intentional or an oversight.

Overall, this script can be used as a starting point for a project that requires the control of multiple ultrasonic sensors and motors. However, it would be best to optimize the code for performance and accuracy.

*****
