VELLORE INSTITUTE OF TECHNOLOGY – AP

ENGINEERING CLINICS

PROJECT REPORT

BLIND AID STICK

Members Involved:

- 22BCE7212 – D.Abhijieeth - CSE
- 22BCE7525 –P. Vivek Vardhan- CSE
- 22BCE7798 – P.Akshay Siddhartha - CSE
- 22BCE7806- K.Harshitha- CSE
- 22BCE8716 – P. Hema Sri- CSE
- 22BCE8813- M.Mahadeep- CSE

UNDER THE GUIDANCE OF:

Prof.Nasseba Beebi

ABSTRACT:

Basically, it is difficult for blind people pass their day to day life with their disabilities. To make their stick smarter, we interfaced some system with their walking stick. In this system we interfaced some smart functions with their stick. Whenever the obstacle is detected on the way through ultrasonic sensor placed on the stick detects the object which is on the way. The detected object processed to identify the type of the object and then it is intimated as buzzer sound through speaker or via earphones connected with Raspberry pi Pico. So that blind can ableto identify the object in-front of them, if it is identified that is a human in their way, they can ask for any help. If there is a large obstacle like a car, they can be able to walk based on the object in-front of them. Additionally, by interfacing GPS with the system the person can find his live location.

CONTENTS:

INTRODUCTION

In the list of disabilities, blind people are facing more difficulties on their day-to-day life. Nowadays every technology becomessmarter, in the list of them, this technology makes walking stick smarter which has many applications such as walking stick indicator in case if they miss the stick through sound beeping. The proposed system also finds the current location details using GPS.

A blind aid stick, also known as a mobility cane, is a tool that helps people who are blind or visually impaired navigate and move around in public. The cane should reach from the user's chin or breastbone to the ground.

The stick helps the user select a path without obstacles. The user moves the stick in front of them to ensure there are no obstacles. The stick can also provide audio directions about the correct path.

Some limitations of a blind aid stick include: They can't detect hidden obstructions and They can't tell the size of an obstacle.

It helps the user to select the path free of any obstacle. The directions about right path are given via audio voice. Blind sticks have been designed earlier which can detect obstacles coming in front of blind people's way, but they were not designed to tell the size of obstacle .

PROCEDURE

1. Connect all the hardware components such as Ultra sonic sensor, GPS and Bluetooth module using the connecting wires.

2. Now connect the Aurdino Board chip to the system using a USB cable

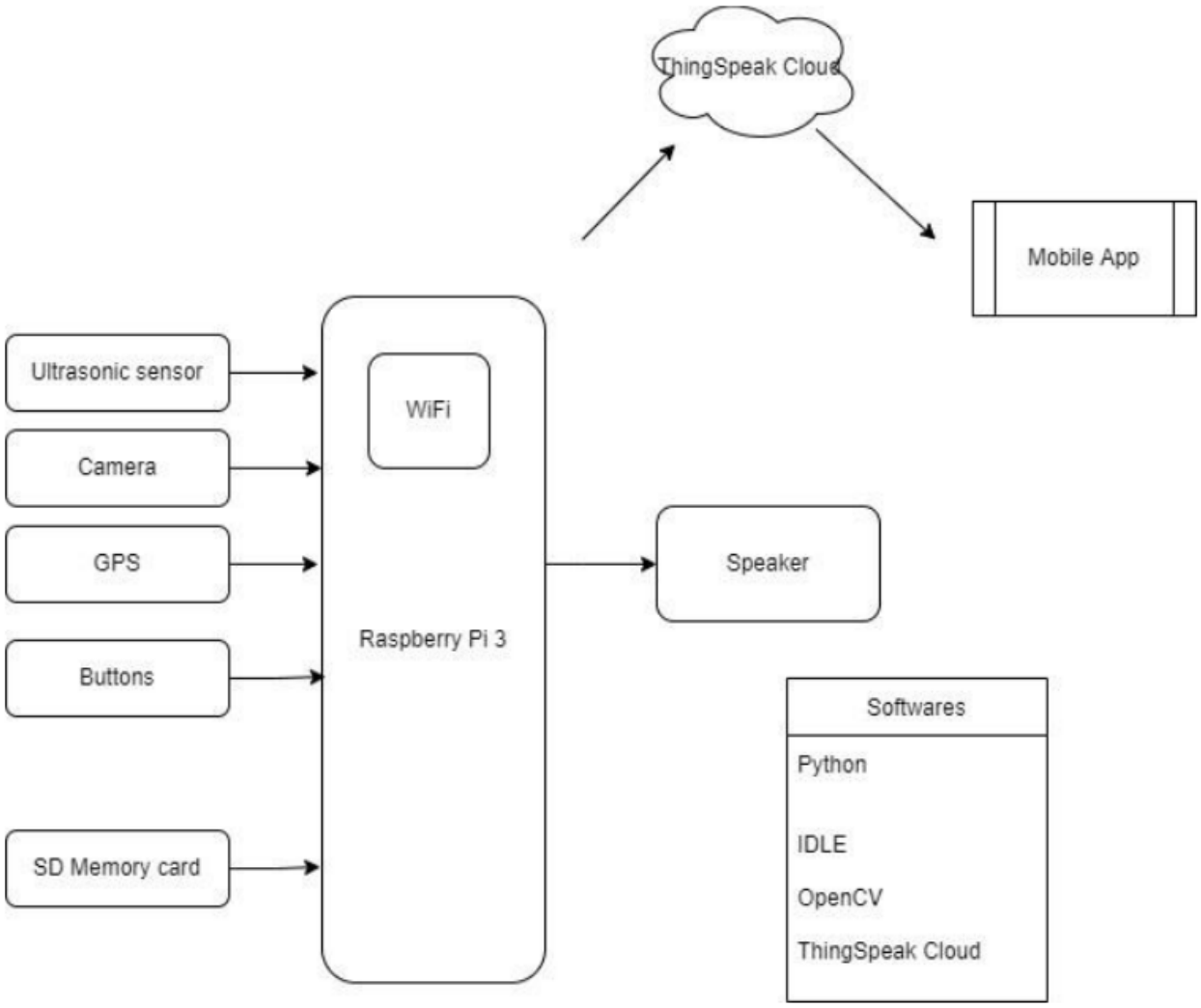
3. Write the program in a soft ware called thonny and connect it to the chip and run it.

4. Write a program such that the buzzer should when ultra-sonic sensor detects the obstacle in the given range.

5. And also the GPS location traced by the GPS should be transferred to mobile using the Bluetooth module.

6. Now attach all the components to a wooden board and place it to the stick which is light in weight .

BLOCK DIAGRAM



TOOLS REQUIRED

HARDWARE COMPONENTS:

- Ultrasonic sensor
- Aurdino Board
- Jumper wires
- Cable and Power bank
- LED
- Bluetooth module
- GPS

ULTRASONIC SENSORS:

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. 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. It has a maximum radius of 200 cm.

Aurdino Board:

Arduino has over the years released over 100 hardware products: boards, shields, carriers, kits and other accessories. In this page, you will find an overview of all active Arduino hardware, including the Nano, MKR and Classic families.

JUMPER WIRES:

Jumpers are typically small metal connections that are used to open or close circuitcomponents.

They have two or more connecting points that control a circuit board for an electrical system. They are responsible for setting up the motherboard and other computer devices.

Cable and Power bank:

A power or battery bank is a portable device that can supply energy and power from its built-in battery, typically through a USB port.

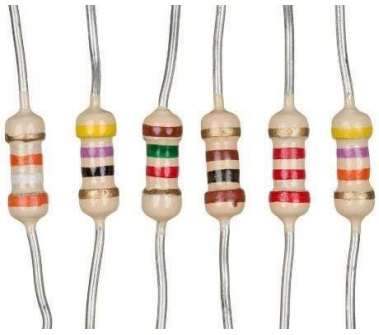
Connect the power bank and Aurdino board using USB cable.

We are using 10,000MAH battery, it lasts up to 2-3 day

RESISTORS:

A resistor is a type of electrical component used to regulate voltage and temperature. 1k ohm resister:

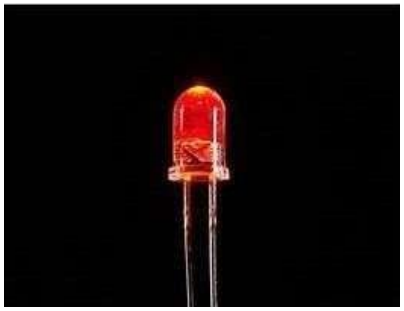
1k Ohm, 1/6th Watt, +/- 5% tolerance PTH resistors. Commonly used in breadboard The LEDs' current is limited by the 1K resistors to prevent overheating.
The common value for hard pull-up and pull-down resistors is 1K.



-

LED

A Super Bright 5mm LED is incredibly bright and has a broad beam angle, making it ideal for usage in models, illuminations, headlights, spotlights, and automobile lighting. Anywhereyou require dependable, high-intensity, low power lighting or indicator, the 5mm LED may be employed. They enter a breadboard fast.



GPS:

The signal contains data that a receiver uses to compute the locations of the satellites and to make other adjustments needed for accurate positioning. The receiver uses the time difference between the time of signal reception and the broadcast time to compute the distance, or range, from the receiver to the satellite.

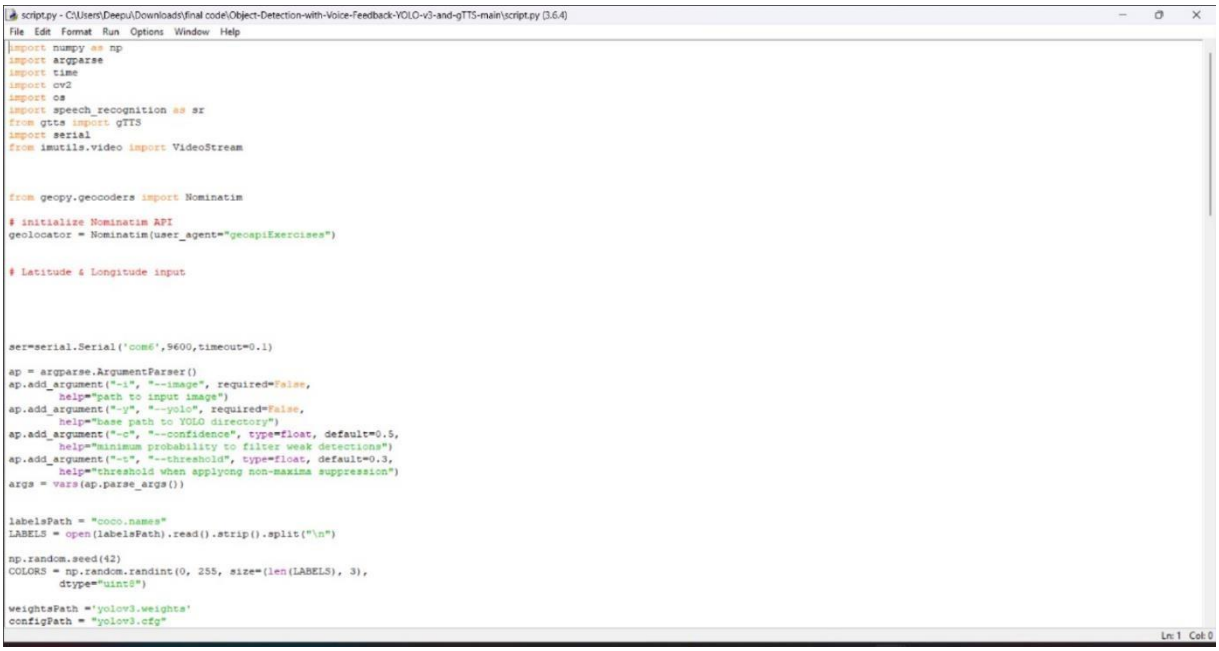
Bluetooth Module:

Bluetooth module (Bluetooth module) refers to the basic circuit set of the chip with integrated Bluetooth function, used for short-range 2.4G wireless communication module. For the end user, the Bluetooth module is a semi-finished product. It is used as Bluetooth Serial port prototype module. It can be easily programmed as master and slave. It is fully enhanced with data rate 3 Mbps modulation with complete 2.4 Ghz radio transceiver and baseband.

Software:

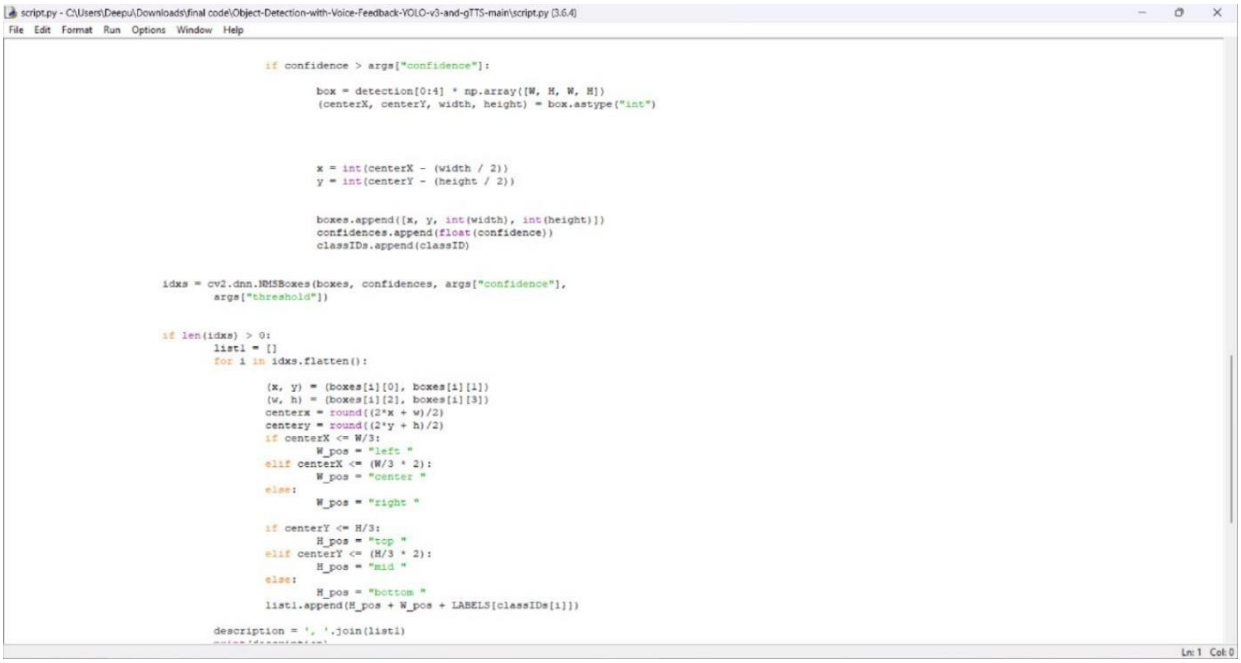
We need to install yolo software in the system and connect it to Aurdino board using cable and run the programmed

Code:



```
import numpy as np
import argparse
import time
import cv2
import os
import speech_recognition as sr
from gtts import gTTS
import serial
from imutils.video import VideoStream


from geopy.geocoders import Nominatim

# initialize Nominatim API
geolocator = Nominatim(user_agent="geoapiExercises")


# Latitude & Longitude input




ser=serial.Serial('com6',9600,timeout=0.1)

ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=False,
        help="path to input image")
ap.add_argument("-y", "--yolo", required=False,
        help="base path to YOLO directory")
ap.add_argument("-c", "--confidence", type=float, default=0.5,
        help="minimum probability to filter weak detections")
ap.add_argument("-t", "--threshold", type=float, default=0.3,
        help="threshold when applyong non-maxima suppression")
args = vars(ap.parse_args())


labelsPath = "coco.names"
LABELS = open(labelsPath).read().strip().split("\n")

np.random.seed(42)
COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),
        dtype="uint8")

weightsPath ='yolov3.weights'
configPath = "yolov3.cfg"
```

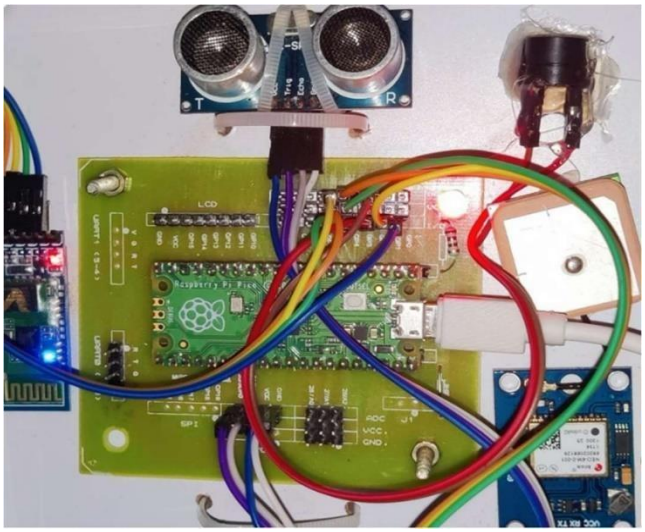
```
print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
vs=VideoStream(src=0).start()
time.sleep(2)
while(1):
        image = vs.read()
        cv2.imshow('in',image)
        key = cv2.waitKey(1)
        info=str(ser.readline().decode())
        if(len(info)>0):
                inf=info.split(',')
                print(inf[1])
                if(inf[1]=='object'):
                        (H, W) = image.shape[:2]

                        ln = net.getLayerNames()
                        ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]


                        blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),
                                        swapRB=True, crop=False)
                        net.setInput(blob)
                        start = time.time()
                        layerOutputs = net.forward(ln)
                        end = time.time()


                        print("[INFO] YOLO took {:.6f} seconds".format(end - start))


                        boxes = []
                        confidences = []
                        classIDs = []
                        ID = 0


                        for output in layerOutputs:

                                for detection in output:

                                        scores = detection[5:]
                                        classID = np.argmax(scores)
                                        confidence = scores[classID]
```


script.py - C:\Users\Deepu\Downloads\final code\Object-Detection-with-Voice-Feedback-YOLO-v3-and-gTTS-main\script.py (3.6.4)
File Edit Format Run Options Window Help
if confidence > args["confidence"]:
box = detection[0:4] * np.array([W, H, W, H])
(centerX, centerY, width, height) = box.astype("int")
x = int(centerX - (width / 2))
y = int(centerY - (height / 2))
boxes.append([x, y, int(width), int(height)])
confidences.append(float(confidence))
classIDs.append(classID)
idxs = cv2.dnn.NMSBoxes(boxes, confidences, args["confidence"],
args["threshold"])
if len(idxs) > 0:
list1 = []
for i in idxs.flatten():
(x, y) = (boxes[i][0], boxes[i][1])
(w, h) = (boxes[i][2], boxes[i][3])
centerx = round((2*x + w)/2)
centery = round((2*y + h)/2)
if centerX <= W/3:
W_pos = "left "
elif centerX <= (W/3 * 2):
W_pos = "center "
else:
W_pos = "right "
if centerY <= H/3:
H_pos = "top "
elif centerY <= (H/3 * 2):
H_pos = "mid "
else:
H_pos = "bottom "
list1.append(H_pos + W_pos + LABELS[classIDs[i]])
description = ', '.join(list1)
Ln: 1 Col: 0


script.py - C:\Users\Deepu\Downloads\final code\Object-Detection-with-Voice-Feedback-YOLO-v3-and-gTTS-main\script.py (3.6.4)
File Edit Format Run Options Window Help
elif centerX <= (W/3 * 2):
W_pos = "center "
else:
W_pos = "right "
if centerY <= H/3:
H_pos = "top "
elif centerY <= (H/3 * 2):
H_pos = "mid "
else:
H_pos = "bottom "
list1.append(H_pos + W_pos + LABELS[classIDs[i]])
description = ', '.join(list1)
print(description)
myobj = gTTS(text=description, lang="en", slow=False)
myobj.save("object_detection.mp3")
os.system("object_detection.mp3")
else:
Latitude=inf[1]
Longitude=inf[2]
print('LT:'+str(Latitude))
print('LG:'+str(Longitude))
location = geolocator.reverse(Latitude+","+Longitude)
address = location.raw['address']
# traverse the data
city = address.get('city', '')
state = address.get('state', '')
country = address.get('country', '')
code = address.get('country_code')
zipcode = address.get('postcode')
print('City : ', city)
print('State : ', state)
print('Country : ', country)
print('Zip Code : ', zipcode)
add=city +" " + state+" "+country
myobj = gTTS(text=add, lang="en", slow=False)
myobj.save("object_detection.mp3")
os.system("object_detection.mp3")
Ln: 1 Col: 0

Result:

When there is any obstacle Infront of the ultra-sonic sensor in the given range then the speaker will give a buzzer sounds that indicates that there is an obstacle so that the blind people can change direction or avoid the obstacle. In the code we have given the distance of 100 centimeters . and GPS tracs the exact location latitude and longitude value of the stick and transferred to mobile using the Bluetooth module.



Advantages:

- It will save the old/blind man from any type of obstacles.
- It will be very helpful to blind people.
- It also equipped with GPS, it will help us to find the exact location of the person.

CONCLUSION AND FUTURE SCOPE:

Blind aid Stick is a project which helps blind people with their problem in vision , when they are travelling the device will show the path to them. This prototype is user-friendly and also cost-effective. This prototype helps the user to move from one place to another and themindependent.

We can add special features like voiceassistant ,camera that recognizes the obstacles and alsonavigation software.

References:

- https://github.com/
- https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html
- https://www.geeksforgeeks.org/python-programming-language/

… THANK YOU …