EEE F411
Internet of Things

A
REPORT
ON

SMART STICK FOR VISUALLY IMPAIRED

By Group 24

Nishtha Pareek	**2019B1A81044P**
Nehal Gupta	**2019B1A80990P**
Devesh Mittal	**2020A8PS0570P**
Isha Srivastava	**2019B1A31034P**
Richa Prasad	**2019B5A80732P**

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE,
PILANI (Rajasthan)

(December, 2022)

ACKNOWLEDGEMENT

We'd like to thank Prof. Vinay Chamola for giving us the opportunity and resources to work on this project which aims to build a smart stick for visually impaired people. The lectures and hands-on labs that Dr. Chamola conducted, helped us immensely in making this smart device and learning even further than what we'd learnt in class. His constant suggestions and guidance in clearing our doubts throughout the semester and in this project are priceless. We'd also like to thank the institute for providing us the hardware and infrastructure due to which this project was possible.

TABLE OF CONTENTS

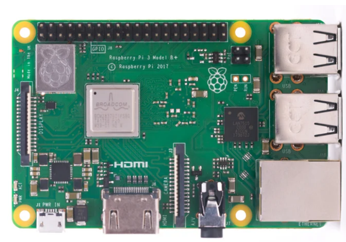
INTRODUCTION

Any disability, with itself brings a whole range of challenges for a person. Advancements in technology have made it possible now to deal with few of those challenges, making a little positive change in their lives. In this project, we aim to create a device for visually impaired people to navigate more conveniently in their normal life. The impaired vision puts them in a vulnerable position every time they go outside. Everyday we get to know about the abundant cases of accidents, robberies and other crimes against visually impaired people when they step outside their homes. Presently Blind people use a white stick as a tool for detection, when they move or walk. But sometimes, they might not be able to know what's ahead of them with a normal blind stick or might drop the stick due to a variety of reasons. Here we have tried to develop a smart tool which can serve as a blind stick being more efficient and helpful than the conventional one.

Components and Technologies

Raspberry Pi-4

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT). In this project, we are making use of Raspberry pi-4.

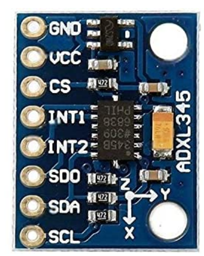


Ultrasonic Sensor (HC - SR04)

HC-SR04 is an ultrasonic ranging module that provides a 2 cm to 400 cm non-contact measurement function. The ranging accuracy can reach up to 3mm and effectual angle is < 15°. It can be powered from a 5V power supply.

Accelerometer Module (ADXL345) :

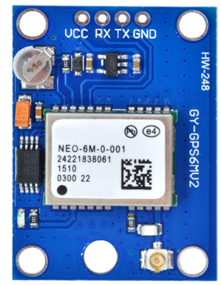
The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface.



Webcam(logitech):

Logitech C270 Digital HD Webcam with Widescreen HD Video Calling. It has

GPS Module (NEO-6M):



It can track up to 22 satellites over 50 channels and achieve the industry's highest level of tracking sensitivity i.e. -161 dB, while consuming only 45 mA current. Unlike other GPS modules, it can perform 5 location updates in a second with 2.5m horizontal position accuracy.

3-Pin Buzzer:

Push Button:

Twilio

Twilio is a customer engagement platform which makes sending and receiving SMS easy.

Features and Working

1. Object Detection and identification of objects once it is closer than 100m.
2. User is informed about the identity and distance of the object via earphones.
3. Alarm system in case of the stick falling on the ground until the user picks it up. This results in ease of location of the stick.
4. SMS alert sent to close ones in times of inconvenience(stick falling) or danger..
5. GPS location of person sent to emergency contact in case of stick falling.

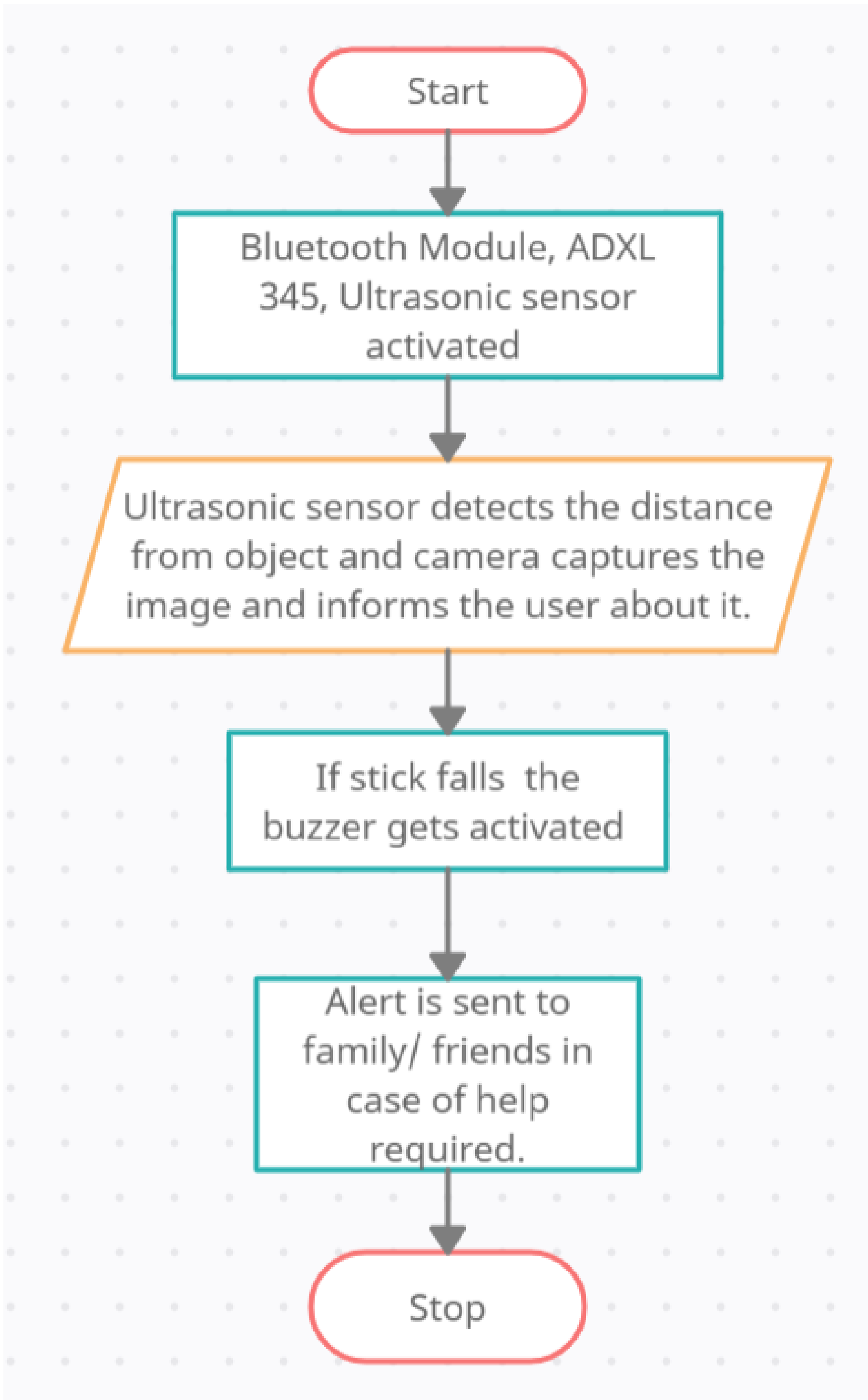
The Smart Stick will be integrated with all the components mentioned earlier in the hardware section

1. The webcam and ultrasonic sensor will be on throughout the period of the smart stick being on.
2. The ultrasonic sensor will send a pulse every few milliseconds and if there's an obstacle nearby in the range of 100 meters, the webcam will be turned on. The object detection module will capture and identify the object and the audio corresponding to the same will be transmitted on the earphones connected to the Raspberry Pi.
3. Once the object is closer than a specified distance limit, the buzzer will alert the user that there is an obstacle in close

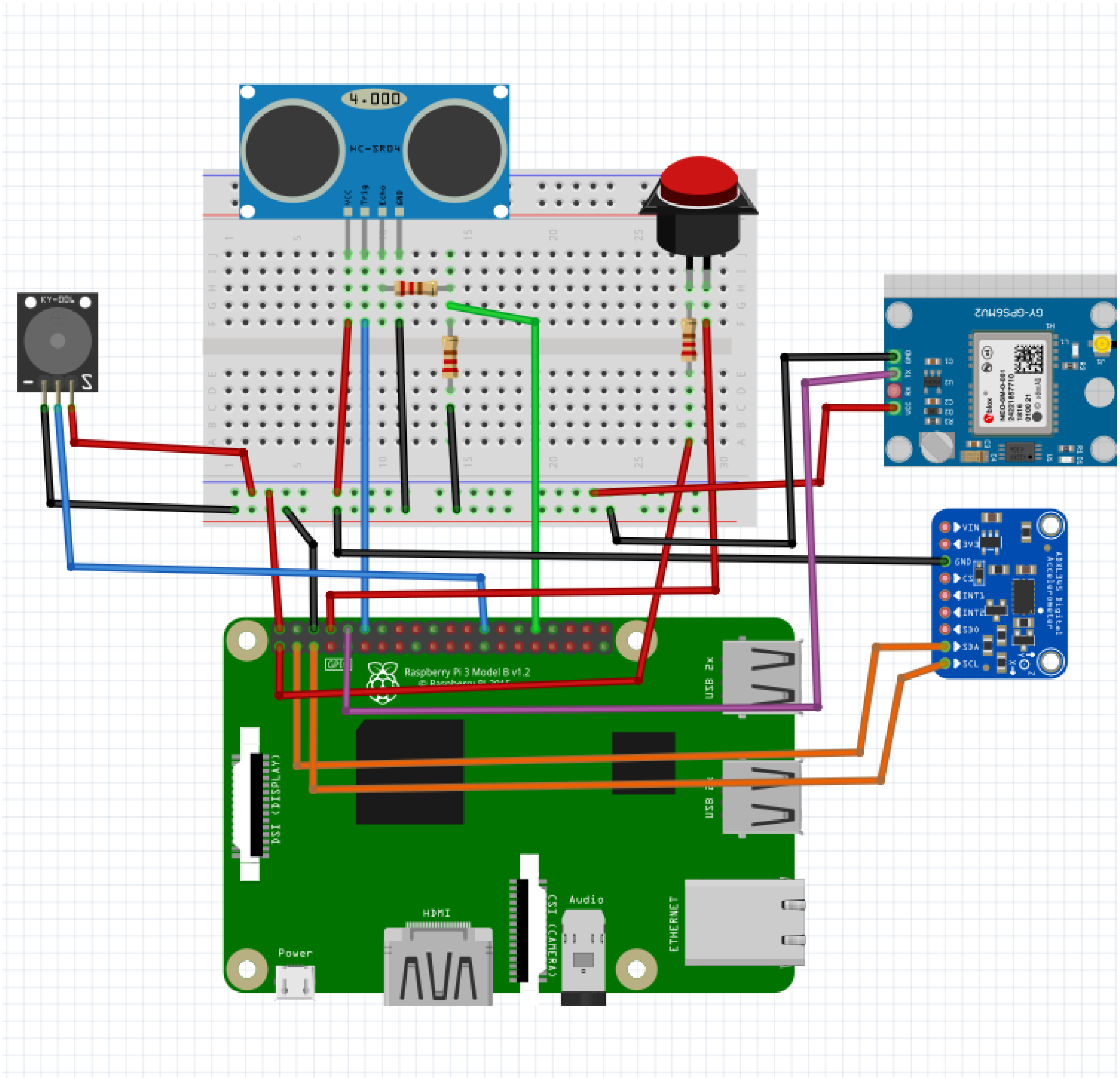
proximity, while the object detection will inform them about the nature of the object.

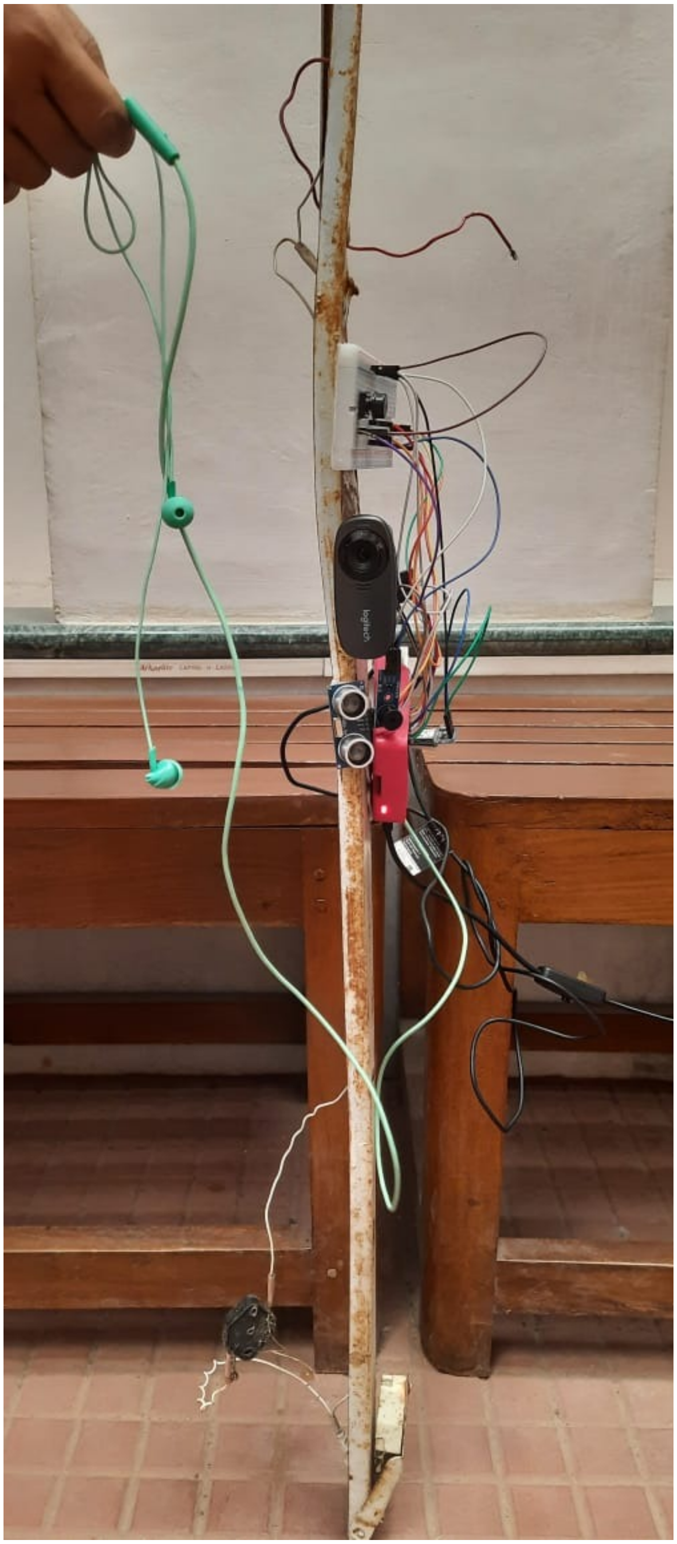
4. In case the user drops the stick, the accelerometer will detect that the stick has fallen and the buzzer will keep buzzing until the stick has been picked up.
5. In addition an alert will be sent to an emergency contact's phone via the **Twilio App.**
6. GPS location for the same will also be shared with the person of contact.

Schematic/Block Diagram



Circuit Diagram and Final Design





Code

```
import RPi.GPIO as GPIO

from time import sleep

import time

import os

from twilio.rest import Client

import gps as loc


#Disable warnings (optional)

buttonPin = 19

GPIO.setmode(GPIO.BCM)  # choose BCM or BOARD numbering schemes. I
use BCM

GPIO.setup(13, GPIO.OUT)# set GPIO 25 as an output. You can use any
GPIO port

GPIO.setup(buttonPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)


account_sid = "AC174a0802c2e63396a3807e4d26b6bc88"

auth_token = "32f7d8cf2a5d76bc4cdd2504d3df0765"

client = Client(account_sid, auth_token)


def buz():

 n =1

 p = GPIO.PWM(13, 200000)    # create an object p for PWM on port 25
at 50 Hertz

 p.start(50)             # start the PWM on 70 percent duty cycle
```

```
 print("Buzzer working")
 p.ChangeFrequency(20000)
 latitude, longitude = loc.gps_start()
 message = client.messages.create(
 body="Person in danger. Call to check\nlatitude:"+ latitude
+"\nlongitude:"+longitude+"\n",
 from_="+19498326594",
 to="+918529691031" )
 print(message.sid)

 while n==1:
   buttonState = GPIO.input(buttonPin)
   if buttonState == False:
      message = client.messages.create(
      body="Help not needed",
      from_="+19498326594",
      to="+918529691031" )
      print(message.sid)
      p.stop()

      n=0

def stop():
   p.stop()
```

```
import serial

import time

import string

import pynmea2


def gps_start():

      port="/dev/ttyAMA0"

      ser=serial.Serial(port, baudrate=9600, timeout=0.5)

      dataout = pynmea2.NMEAStreamReader()

      newdata=ser.readline()


      if newdata[0:6] == "$GPRMC":

            newmsg=pynmea2.parse(newdata)

            lat=newmsg.latitude

            lng=newmsg.longitude

            gps = "Latitude=" + str(lat) + "and Longitude=" +
str(lng)

            print(gps)
```

Cost

Raspberry Pi-4 (4GB RAM)- Rs 4969
Breadboard- Rs 95
Logitech Webcam-Rs 1995
Ultrasonic sensor-Rs 72
Accelerometer-Rs 125
GPS Module NEO-6M- Rs 265

Total Cost= Rs 7521

Results and Conclusions

Thus we made a smart device with the following features:

- Is able to identify the obstacle present in front of the visually impaired person and also tell how far from the person it is.

- Is able to alert emergency contacts in times of need such as when the stick has fallen.

- Is able to indicate to the user when the stick has fallen so the person can locate it.

This project was a great learning experience for all the five of us. We put to use what we had learned in our lecture and lab sessions and make something that is meaningful to us and others. Through this project, not only we learnt what we implemented, but also what could be implemented. It broadened our perspective and gave us a broader vision about the versatility of the Internet of Things. Other than the technical aspect, we learnt how teamwork is crucial and extremely important for taking these projects to the next level.

Challenges faced

One of the major challenges we faced was the accuracy of these sensors.

The Ultrasonic Detector is sensitive and can make an alert even for relatively small objects which might not actually be an obstacle.

The object detection library from tensorflow has been quite functional but sometimes it makes erroneous identifications.

The webcam response has a bit of a lag even after multiple attempts to get rid of it.

References

https://sparklers-the-makers.github.io/blog/robotics/use-neo-6m-module-with-raspberry-pi/

https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/

https://pimylifeup.com/raspberry-pi-accelerometer-adxl345/

https://fritzing.org/parts/

https://forums.raspberrypi.com

https://www.tensorflow.org/hub/tutorials/object_detection