Project

BlindSight

Liam MARTIN-ESCOFFIER

Bachelor (Hons) of Software & Electronic Engineering

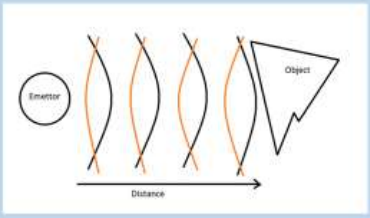
Galway-Mayo Institute of Technology

2019/2020

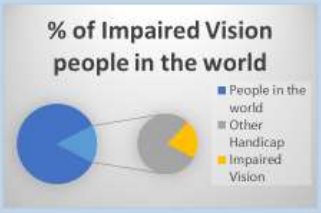
BlindSight

By Liam MARTIN-ESCOFFIER – G00364974

BlindSight is my 4th year college project.
It is a Smart Jacket with a built in system to help users, *who are blind or with vision impairment,* avoid obstacles when moving around.



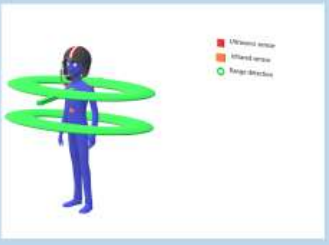
This project was made to facilitate the life of visually-impaired people and help them in their ordinary life.
I have based my research on Biomimicry, "Biomimicry is the imitation of the systems of nature for the purpose of solving complex human problems."



This project was made under Python with the use of an Arduino Uno Mega and a Raspberry Pi 4.

The Avoidance system use Ultrasonic sensor to determine if an object is under a threshold distance and if so alert the user

My plan is to develop an even better version with the intent to sell it to a health care so it can be produced on a larger scale, and be use for the highest number of people possible



Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Software & Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

_____

Table of Contents

Summary

BlindSight is a project made to help people with impaired vision, it's designed to help them with the difficulties they encountered in their daily life especially in a place with a lot of obstacles.

This project can interact with the user using Voice command it can signal to the user the direction of the obstacle that is under a certain threshold and communicate it to the user via vibration on the appropriate direction.

The main approach for this project was the use of echolocation by different animals to detect their prey in situation were the visibility is low, this is a principle called biomimicry.

The main approach for this project was to try to think of the different possibilities to communicate a stimuli that everybody have to a person that can barely have it or to someone that cannot process this stimuli.

The technology used are Python, Arduino, Raspberry Pi.

I have successfully created the jacket however it exists still some minor correction to make.

This project is something that will help a lot of people, and I hope to sensibilize the others to how the life of impaired vison people can be sometimes difficult.

Introduction:

For this final year, we need to create a project, that will “concentrate” all the different aspect of our formation, that’s mean software engineering.

For my project I choose to find a new and innovative way of helping people so I research, how I can help with different handicap at my level, I choose to try to accommodate the life of people with impaired vision.

For doing so I first try to found different way of simulate vision but the user interface was an obstacle for it, so I designed this project based on how nature have done it, indeed the dauphin and bat use the principle of echolocation to help them navigate in the 3D world.

Inspired by this I try to understand the limit of my system and build around them.

So, I will in this report describe how I managed to create this project.

The inspirations for this project

This project was designed to help people with impaired vision be able to detect some higher than ground obstacle. Because even if walking stick and dogs are quite efficient it can happen that they didn’t recognize taller obstacle.

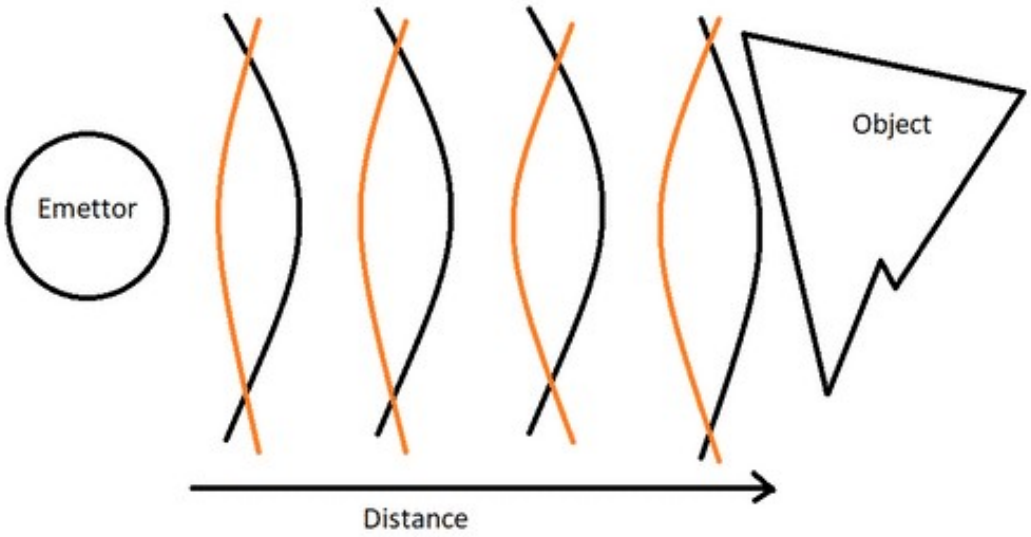
So, for helping them I research how can I help them to accommodate with this issue.

By doing so I read about biomimicry and was inspired by it.

The principle of Biomimicry is how the human design a system based on the imitation of the already existing system present in nature.

In the case of this project I was inspired by the ability that dauphin and bats use to detect their preys.

Indeed, using ultrasound they can know the distance that is between them and their prey. And if we apply this to my project it means that I can know the distance between them and an obstacle and inform the user of it.



Architecture of the System

This project is composed of different parts:

- A vest to hold the different element together
- An Arduino Board for the detection program
- A Raspberry pi 4 for the high-level functions
- 3 Ultrasonic sensors use for detection
- 1 IR distance sensor
- 4 Vibration motor use as an interface
- 1 Microphone for the voice command

Each of this part work together and communicate between each over, to create the main system, for example:

If a Ultrasonic sensor detect a distance below a certain threshold (80 cm) , the Arduino mega will pick it up and then send it to the Raspberry pi then the raspberry will store it, during this time the Arduino will activate the vibration motor corresponding to the sensor.

And on another program the raspberry will be listening to certain voice command and interact with the user if the commands are correct.

The different parts are designed to work in a certain way, for example the Ultrasonic sensor have been chose because of the area of detection, contrary to IR sensor that just have a very precise but it’s far too narrow for the use of the project.

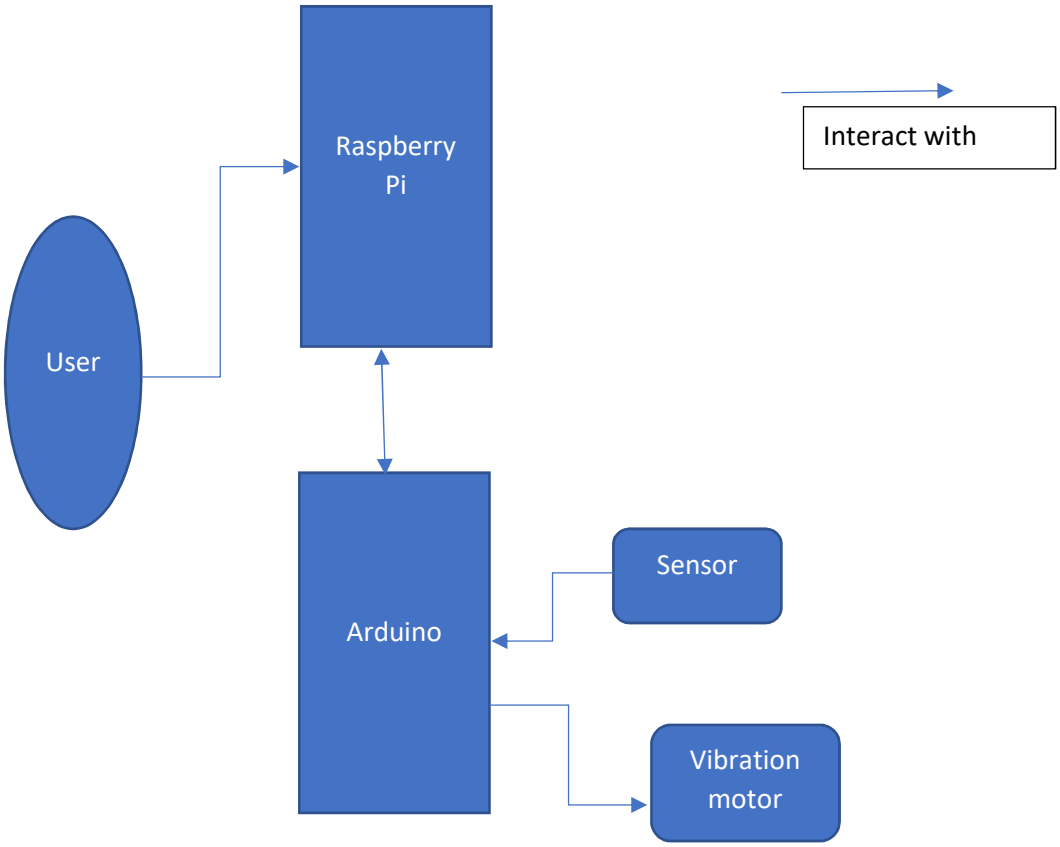
I have chosen to divide how the project work in 3 different programs:

- The first one is the Arduino one, it checks the data of the sensor each loop and send it to another program on the raspberry pi

- The second Program host on the Raspberry pi will receive the number of the sensor that have been activated and stored it in a file.

- The third program will work on the “smart” part of this project using Speech to Text and Speech to Text it will activate different functions when asked for it.

The 3 programs will then work together to create the whole project.



What are the Technologies?

Arduino & Raspberry

Arduino

The Arduino system is a software made by the company of the same name, they design for the most part single board computer for the use of embedded, this software is open source that mean that everyone can use and modify it.

In this project I use the Arduino Mega, a board that is different from the “general” use of the Uno Rev3.

Let’s compare both with the use we need for them:

We have 3x HR SO4 ultrasonic sensors, with each of them having 4 pin we need at least 12 digital pins plus the 3 for the IR distance sensor and the 4 vibrator motors we will need a total of 12 + 3 + (4*2) =23 pins. 22 of them as digital ones and 1 as analogical one.

23 Pins

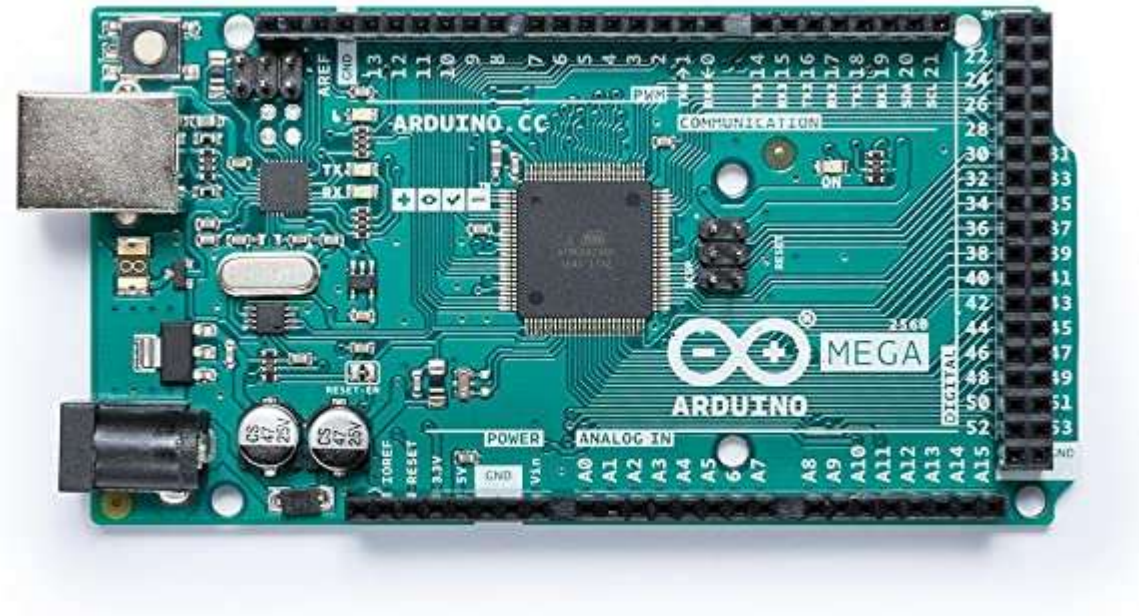
Arduino Uno Rev3:

- Digital I/O Pins: 14
- Analog Pins: 6
- Total Pins: 20

Arduino Mega Rev3:

- Digital I/O Pins: 54
- Analog Pins: 16
- Total Pins: 16

So, it’s better to use the Arduino Mega because of the number of I/O we will need in this project.



Raspberry Pi

The raspberry pi is a single board Computer designed by the raspberry foundation, designed to the use of teaching and learning computer science, it had developed far beyond expectation and the first model became very popular.

Nowadays they existed a lot more than 1 or 2 models especially with the release in 2019 of the raspberry Pi B (the model we used in this project), this model can have until 4 GB of ram, it’s power supply by USB-C and have 4 different USB port (2 USB port 2 and 2 USB 3).

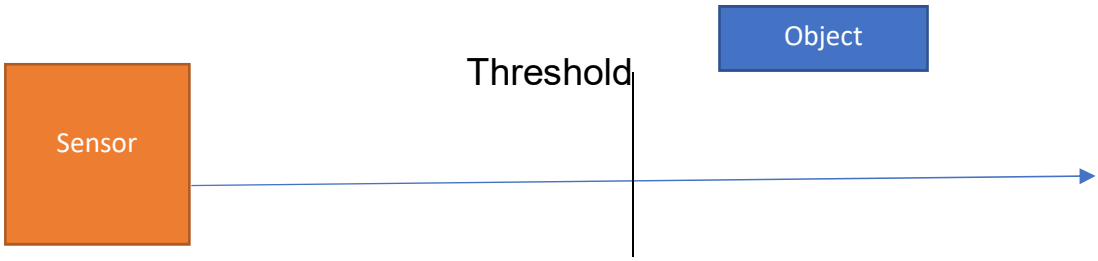
In this project the Raspberry will help us create the high-level function like Geolocation, send SMS, …

The model we use will be the Raspberry Pi 4 B, 4GB.

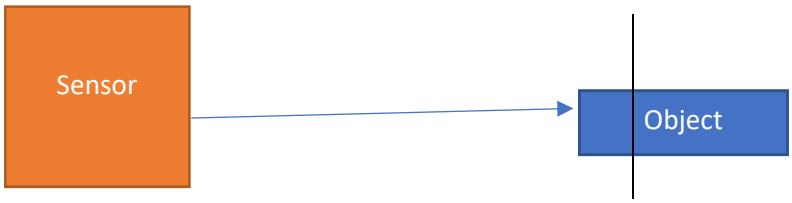
The different code for Arduino and Raspberry

Arduino

The code for the Arduino is quite simple, we need something that will send a signal if a sensor detects something under a threshold for example:



In this case we will not send a signal because the object is away from the threshold.



The object is now under the distance threshold we can send a signal to the board.

For this we will use a “if” statement as follow:

```
if (measured distance is inferior to threshold) then
      do something ….
```

This will become in code:

```
if(distance<30){
      string_to_send += ‘L’;
      Motor(1, ”on”);
}
else {
      string_to_send += ‘B’;
      Motor(1, ”off”);
}
```

The previous code means that if the distance if less than 30 cm we will send add to a string ‘L’ to lesser or ‘B’ to bigger and depending on the condition we will make the Motor 1 vibrate or not.

The program will work like this:

- We define our variable, Input and Output
- We create an empty string
- We enter a loop
- We ask the sensor if something happen and add the answer to the string
- We make the motor vibrate if needed
- We finally send the string to the raspberry Pi

For declaring our Sensor as an Input or as an Output, the variable will be declared as follow:

```
void setup() {
pinMode(trigPinl, OUTPUT);
pinMode(echoPinl, INPUT);
```

In the setup part we use the pinMode() function and we put in 2 arguments, the first one will be the number of the pin use, the second is INPUT or OUTPUT or INPUT_PULLUP.

For creating a String it's: `String variable_name = "…";`

We can also choose to not iniate it: `String variable_name;`

The loop will be automatically built by the Arduino IDE and look as follow:

```
void loop() {


}
```

It’s the main part of our program where everything will be.

In this part, first the “Virtual Ground” and “Virtual VCC” will be declared, it’s use is to allow us to have more GND and VCC pins on the board.

```
digitalWrite(VCC3, HIGH);
digitalWrite(GND1, LOW);
```

After everything is declared like above, we will go into the main Part of the program, it’s divided in 3 different functions both having different purposes. The first one is made to measure the distance on the Ultrasonic sensor (measureDist) while the second one is for the vibration motor to be activated (motorStatus) the last function is for measuring the distance on the IR sensor (irDist).

```
////////////////////////////////////////////////////////////////
int measureDist(int triggerPin, int echoPin,int motorPin){
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance1= duration*0.034/2;
    if(distance1<30){
      msg_send+='L';
      motorStatus(motorPin,"on");
    }
    else{
      msg_send+='B';
      motorStatus(motorPin,"off");
    }
}

////////////////////////////////////////////////////////////////
```

This function work as in two parts, the first one measure the distance while the second compare the measured value to the threshold here 30 cm.

If the value is bigger than we add a ‘B’ to the string that we are going to send later, and we make sure the motor is off.

If the value is lower than the threshold than we add ‘L’ to the string, and we put the motor on.

```
void motorStatus(int motorPin, String cmd){
  if(cmd == "on"){
    digitalWrite(motorPin,HIGH);
  }
  else if (cmd == "off"){
    digitalWrite(motorPin,LOW);
  }
}
```

The motorStatus function will take 2 arguments one is the pin Number the other one is the status of the motor ‘on’ or ‘off’ and based on that we activate or not the motor.

The function for the IR sensor will be roughly the same except that we will work in Analog and not digital.

After all of that we will go back into main to send to the Raspberry the string containing the data (a string of ‘L’ or ‘B’).

Raspberry Pi

For this part we will have 2 different python code, 1 to save the data from the Arduino, the other to manipulate the “High-level” function.

The first Program

For this program we will use a library called “serial”, this library will help us connect the raspberry to the Arduino and let them talk to each other, it will work as follow:

- Import all the library we need
- Create some Variables
- Give a path to the file
- Create the link between the 2 boards using serial.Serial()
- Read the value given
- Put them into the file with a timestamp if a ‘L’ appears else discard the data
- Add a number to the value pass in
- Check if the number is bigger than 200
- If it is bigger than delete the file and recreate it
- Else continue the previous work

This part is designed to save some of the value read while if the file become too big and impact the performance of the system dispose of it.

```
import serial
from datetime import datetime
 import os
now = datetime.now()

list_detect = ['N','N','N','N','N','N','N','N']
count = 0
send_data=""
arduino = serial.Serial('COM1', 9600, timeout=.1)
f= open("Save.txt","w+")
f.close()
while(True):
      f=open("Save.txt", "a+")
      string_from_serial = arduino.read(10)
      if 'L' in string_answer_from_serial:
         if i==200:
            if os.path.exists("Save.txt"):
              os.remove("Save.txt")
         TimeStamp = now.strftime("%m/%d/%Y, %H:%M:%S")
         f.write("Time stamp: "+TimeStamp+" "+list_detect % (i+1))
```

We will talk a bit more about the different library use in this program:

Serial:

This module is made to access the serial port from Python, it allow to use the connection made as an API with the help of command like “read”,” write”.

The module is easy to install with a single command “*python -m pip install pyserial*”.

DateTime:

This module helps with the manipulation of date, here it’s use as a timestamp.

OS:

This module was made to allow the program to access the operating system, it has a multitude of command for manipulating the os and get information out of it.

The Second Program

First let's look at the full program, then we will explain bit after byte.

```
#!/usr/bin/env python3.6.8

import os,random
import subprocess
import speech_recognition as sr
from twilio.rest import Client
import requests
import webbrowser
import pyttsx3
engine = pyttsx3.init()
r = sr.Recognizer()
import wikipedia

import shlex
account_sid ="AC6a6f62dbaf928dab9d849e25d1412efb"

auth_token ="25380b677c458cbfca4abc0eea9bdce0"


Number_list = ["+33617966728","+33617966728"]

print(">>>")
engine.say("I'm listening")
engine.runAndWait()
#subprocess.call(["espeak","I'm Listening"])

print(sr.Microphone.list_microphone_names())
```

```
def Wikipedia():
    print(">>>")
    engine.say("What subject ?")
    engine.runAndWait()
    r = sr.Recognizer()
    with sr.Microphone() as source:
        #r.adjust_for_ambient_noise(source)
        audio = r.listen(source)
    print("Analizying...")
    gg= r.recognize_google(audio, language = "en-US")
    print(wikipedia.summary(gg,sentences=3,auto_suggest=True, redirect=True))
    try:
        engine.say(wikipedia.summary(gg,sentences=3,auto_suggest=True, redirect=True))
        engine.runAndWait()
    except wikipedia.DisambiguationError as e:
        print("disambiguation")
    return
```

```
def SMS():


    print(">>>")
    engine.say("What do you want to send ?")
    engine.runAndWait()
    r = sr.Recognizer()
    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)
    print("Analizying...")
    gg= r.recognize_google(audio, language = "en-US")
    ggfr = r.recognize_google(audio, language = "fr-FR")
    client = Client(account_sid, auth_token)
    for i in Number_list:


        message = client.api.account.messages.create(

        to=str(i),

        from_="+447588675953",

        body="This is a message send to you via the Smart Jacket -->"+gg+" or you have said in French :"+ggfr)


    return
```

```
def emergency():
    engine.say("EMERGENCY PROTOCOL ACTIVATED !")
    engine.runAndWait()
    def display_position():
        """  Function To Print GeoIP Latitude & Longitude """
        ip_request = requests.get('https://get.geojs.io/v1/ip.json')
        my_ip = ip_request.json()['ip']
        geo_request = requests.get('https://get.geojs.io/v1/ip/geo/' +my_ip + '.json')
        geo_data = geo_request.json()

        a=("latitude "+geo_data['latitude'],"longitude "+geo_data['longitude'])

        str = ' --  '.join(a)
        return str



    client = Client(account_sid, auth_token)
    for i in Number_list:


        message = client.api.account.messages.create(

        to=str(i),

        from_="+447588675953",

        body="Jacket alert system SEND LAST COORDONATE : --  "+display_position())
    return
```

```
while True:

    mic=sr.Microphone()
    with mic as source:
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)



    Commande = r.recognize_google(audio, language = "en-US")
    print("Analizying...")
    if(Commande == "send SMS"):
        SMS()
    elif(Commande == "emergency"):
        emergency()
    elif(Commande == "Wikipedia"):
        Wikipedia()
    elif(Commande == "stop"):
        break
```

So, the first part is the different module and library used in the program, following is a list of them with a small summary of their functions.

Random:

This library allows the use of pseudo random numbers, here it was use as a test in a previous version were the sms send was random (to test purpose).

SubProcess:

In case, the TTS engine doesn’t work we can replace it with subprocess and use the espeak module of Linux to send data to the speaker.

Speech_Recognition:

This library enables us to perform speech recognition, using different API, here we use the google one.

We also need PyAudio because we use a microphone.

So, the highest version of python we can use, is 3.6 due to PyAudio not being updated yet.

Twilio:

This library is an easy to use module to send SMS, it needs a Twilio account to work, but it’s easy to make a free one.

It can be installed via the command “pip install twilio”.

Request:

This module is made to allow an easy access to HTTP resource, here it will help to get back json resource.

WebBrowser:

This module allow python to display web pages, here it’s not use anymore because the system doesn’t have any visual information anymore.

Pyttsx3:

This is the Text to speak engine we put a sentence in it and then we make the computer where the program is speak using the output of the system

Wikipedia:

This module is made to fetch the subject from Wikipedia using different option, you can have the title, a brief summary, choose in which language you want it and a lot of other option.

Shlex:

This module is made for a simpler lexical analysis, it can be used in many cases, but in this program, it was use in case the TTS engine doesn’t work and we need to pass through command line.

Now that we have define the different modules, we can concentrate on the different functions of the program and how does everything work together.

Following the declaration of the module and library come the declaration of some variable especially the identification needed for sending SMS, along that come the beginning of the program where we ask what do we want to do, we will declare the 3 different functions used, SMS, Wikipedia, Geolocation. And then we check what the user says and link it to one of the different functions.

Now let's see in detail how the Wikipedia function work:

```
def Wikipedia():
    print(">>>")
    engine.say("What subject ?")
    engine.runAndWait()
    r = sr.Recognizer()
    with sr.Microphone() as source:
        #r.adjust_for_ambient_noise(source)
        audio = r.listen(source)
    print("Analizying...")
    gg= r.recognize_google(audio, language = "en-US")
    print(wikipedia.summary(gg,sentences=3,auto_suggest=True, redirect=True))
    try:
        engine.say(wikipedia.summary(gg,sentences=3,auto_suggest=True, redirect=True))
        engine.runAndWait()
    except wikipedia.DisambiguationError as e:
        print("disambiguation")
    return
```

First, we make the TTS engine ask for a subject, and then we launch the STT engine without the adjustment for the ambient noise because depending on the microphone use we can already have a noise cancelation mode, we then process the sound input into the google recognition and we define the language to American English but we can change it by modifying the language option, for example “fr-FR” will make the recognition in French.

Following that, for debugging purpose we print what we get, and then make the TTS engine says it while into a try except so we only keep the disambiguation error(expl: if we ask for New York and the micro didn’t pick it up well we will have a disambiguation error).

The SMS function is as follow:

```
def SMS():

    print(">>>")
    engine.say("What do you want to send ?")
    engine.runAndWait()
    r = sr.Recognizer()
    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)
    print("Analizying...")
    gg= r.recognize_google(audio, language = "en-US")
    ggfr = r.recognize_google(audio, language = "fr-FR")
    client = Client(account_sid, auth_token)
    for i in Number_list:


        message = client.api.account.messages.create(

        to=str(i),

        from_="+447588675953",

        body="This is a message send to you via the Smart Jacket -->"+gg+" or you have said in French :"+ggfr)

    return
```

First we will still ask the user what does he want to send, we take the Input from the STT engine and process it in English or in French then using the Twilio API we iterate thought a array of phone number link to the account, and send the message from the phone number register in the website with a body containing the 2 sentences English and French.

The Emergency Function as a similar part than the SMS function because it will still send a SMS but this time we will send Geolocation of the user take by the network is on (I recommend using mobile data as sometimes the network can be host in a different city).

```
def emergency():
    engine.say("EMERGENCY PROTOCOL ACTIVATED !")
    engine.runAndWait()
    def display_position():
        """  Function To Print GeoIP Latitude & Longitude """
        ip_request = requests.get('https://get.geojs.io/v1/ip.json')
        my_ip = ip_request.json()['ip']
        geo_request = requests.get('https://get.geojs.io/v1/ip/geo/' +my_ip + '.json')
        geo_data = geo_request.json()

        a=("latitude "+geo_data['latitude'],"longitude "+geo_data['longitude'])

        str = ' --  '.join(a)
        return str


    client = Client(account_sid, auth_token)
    for i in Number_list:


        message = client.api.account.messages.create(

        to=str(i),

        from_="+447588675953",

        body="Jacket alert system SEND LAST COORDONATE : --  "+display_position())
    return
```

So we first connect to using the request module to a website having only the json of our IP address, that using this IP address we fill the blank to access the Geolocation data of our IP address and the we send it via SMS to everyone who is link to our Twilio account.

Finally, the “Main” is a simple loop where we ask the STT engine to listen and if the User say something related to one of the three functions than we can access the function related.

Ultrasonic sensor

In this project, we used 3 Ultrasonic sensors HR-SO4, they have a measure range between 2cm and 400 cm (0.02m to 4m), they have a power input of 5V so it can be powered via an Arduino.

The Effective angle is around 15°, they are quite small in size 45mm by 20mm by 15mm.

HR-SO4

Vibration Motors

We also use 3 Vibrations motors, they have a power input of 3VCC They will vibrate when trigger and have only 2 pins (VCC & GND), they are also small enough to fit properly into the jacket with a height of 2mm and around 1 cm in length.

Vibration motors

Infrared Sensor

Sharp IR Distance Sensor

This is an analog Distance sensor that can detect a distance between 10 and 80 cm, it has 3 different pins, 1 for the data, and the 2 others for The power input and output.

It operates at 5 V and deliver analog data to the connected board.

Difficulties and The Next Step

The Difficulties encounter

This project was more difficult to realise than I expected, indeed I encountered some issues along the way, some were logistic and other purely from software.

First let's talk about the software issues that I encounter and then following this, we will talk about the logistic problems.

So, what does "a software issue" mean, it's when the software or program I was working on or working with doesn't work as expected due to programming error, version issue, wrong file name, etc.

One of the main difficulty I have encountered was working with the "speech_recognition" library, indeed this library for working correctly need to use PyAudio (due to the use of the microphone) but PyAudio in the other end was not compatible with Python 3.7 so when I first try the program nothing was working at all, and so I start trying to figure out how to change the version of python on the IDE I use (thonny). But then the library was not installed and would not install in a python 3.6 file, so I delete every library, and every version of python before trying anew.

Another Software issue was with the text to speak engine, it was directly linked to the ALSA library (is what make the raspberry pi able to produce

sound on the speaker, etc) because of some incompatibility with other software, after looking on internet for some answer I try to reinstall the ALSA library ant this time it work after a delay of 3 to 5 seconds.

Now for the more Logistical issues, the first one was I didn’t receive all that I need for my project especially the Jacket, the Ultrasonic sensor on time and so I was becoming late on my schedule. This was resolve very quickly but deliver to the wrong address in France.

But the biggest trouble was to relocate myself in mid-March due to the Covid-19 pandemic, so some of the thing I have could not be used or take back in France, and I need to found other way does make everything works again.

The Next Step

So, for what I will do next on this project, first I plan to ameliorate some part of it to create an even better version of it, a more good-looking version too. Then I will discuss with healthcare company or/and the social security in France to help me distinguish what is the best approach to create a viable product that everyone can afford while still making benefits. For this I will maybe make my own company and then work on how to help people with non-invasive technology while still applying the principle of Biomimicry.

Conclusion

To conclude, this project is something that I found very important to me due to several reasons, the time I put on it, the knowledge I gain from it, and also the ability it has to help people.

I have learned indeed a lot of things working on this project, especially working with the unexpected and what happen with it, I have learned to think outside of the box, while still applying my current knowledge to what I do.

Finally, with this project answering the goal it was made for, I think it has developed into something that I want to do because working with a purpose it something that feel great and I hope I can do a it more project like this one.

Source

- https://www.python.org/doc/
- https://pypi.org/
- https://en.wikipedia.org/wiki/Biomimetics
- https://www.raspberrypi.org/forums/
- https://stackoverflow.com/
- https://www.pololu.com/product/136
- https://www.robotshop.com
- https://www.electroschematics.com/hc-sr04-datasheet/

Code

Arduino:

```
void setup() {
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);

  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);

  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin3, INPUT);

  pinMode(VCC1, OUTPUT);
  pinMode(GND1, OUTPUT);

  pinMode(VCC2, OUTPUT);
  pinMode(GND2, OUTPUT);

  pinMode(VCC3, OUTPUT);
  pinMode(GND3, OUTPUT);

  pinMode(VCC4, OUTPUT);
  pinMode(GND4, OUTPUT);

  pinMode(GND_vibra1, OUTPUT);
  pinMode(GND_vibra2, OUTPUT);
  pinMode(GND_vibra3, OUTPUT);

  pinMode(Vibra1, OUTPUT);
  pinMode(Vibra2, OUTPUT);
  pinMode(Vibra3, OUTPUT);

  Serial.begin(9600); // Starts the serial communication
}

//////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////

void loop() {

  digitalWrite(VCC1, HIGH);
  digitalWrite(VCC2, HIGH);
  digitalWrite(VCC3, HIGH);
  digitalWrite(VCC4, HIGH);
  digitalWrite(GND1, LOW);
  digitalWrite(GND2, LOW);
  digitalWrite(GND3, LOW);
  digitalWrite(GND4, LOW);
  digitalWrite(GND_vibra1, LOW);
  digitalWrite(GND_vibra2, LOW);
  digitalWrite(GND_vibra3, LOW);

measureDist(trigPin1,echoPin1,Vibra1);

measureDist(trigPin2,echoPin2,Vibra2);

measureDist(trigPin3,echoPin3,Vibra3);

irDist();

}


////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////
int measureDist(int triggerPin, int echoPin, int motorPin) {
  long duration1;
  int distance1;
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration1 = pulseIn(echoPin, HIGH);
  distance1= duration1*0.034/2;
    if(distance1<=30){
      digitalWrite(motorPin, HIGH);
      delay(250);
      digitalWrite(motorPin, LOW);

      msg_send+='L';
    }
    else{
      msg_send+='B';

    }
}

////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////

int irDist() {
  int distance2 = sensor.getDistance();
  if (distance2 <= 3) {

    digitalWrite(Vibra3, HIGH);
    delayMicroseconds(150);
    digitalWrite(Vibra3, LOW);
    msg_send += 'L';
  }
  else {

    digitalWrite(Vibra3, LOW);
    msg_send += 'B';
  }
  Serial.println(distance2);
  return distance2;

}
```

Python File:

```
#!/usr/bin/env python3



import os,random
import speech_recognition as sr
from twilio.rest import Client
import requests
import webbrowser

r = sr.Recognizer()
import wikipedia
import pyttsx3
import urllib.request
import json



engine = pyttsx3.init()
voices =
engine.setProperty('voice','HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\V
oices\Tokens\TTS_MS_EN-US_ZIRA_11.0')
account_sid ="XXXXXXXXXXXXXXXXXXXX"
auth_token ="XXXXXXXXXXXXXXXXX"


Number_list = ["+XXXXXXXXXX"]


engine.say("I'm listening")
engine.runAndWait()




def Wikipedia():
    print(">>>")
    engine.say("What subject ?")
```

```
    engine.runAndWait()
    r = sr.Recognizer()
    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)


    try:
        gg= r.recognize_google(audio, language = "en-US")
    except Exception as e:
        print("Error: Could not understand    Example: New york City")
        gg= "New York city"
    try:
        print(wikipedia.summary(gg,sentences=3,auto_suggest=True,
redirect=True))
        engine.say(wikipedia.summary(gg,sentences=3,auto_suggest=True,
redirect=True))
        engine.runAndWait()
    except wikipedia.DisambiguationError as e:
        wikipedia.random(pages=1)
    return
def SMS():
    print(">>>")
    engine.say("What do you want to send ?")
    engine.runAndWait()
    r = sr.Recognizer()
    with sr.Microphone() as source:


        audio = r.listen(source)
    try:
        print("Google Speech Recognition thinks you said in English: -  "
+ r.recognize_google(audio, language = "en-US"))
        print("Google Speech Recognition thinks you said in fr: -  " +
r.recognize_google(audio, language = "fr-FR"))


    except Exception as e:
```

```
        print("Error: " + str(e))
    try:
        gg= r.recognize_google(audio, language = "en-US")
    except Exception as e:
        print("Error:  Could not understand  Test Example = Wikipedia")
        gg = "Project Demonstartion"
    client = Client(account_sid, auth_token)
    for i in Number_list:

        message = client.api.account.messages.create(

        to=str(i),

        from_="+12057513314",

        body="This is a message send to you via the Smart Jacket -->"+gg)

    engine.say("SMS send")
    engine.runAndWait()
    return

def emergency():
    engine.say("Trigger emergency signal")
    engine.runAndWait()

    with urllib.request.urlopen("https://geolocation-db.com/json") as url:
        data = json.loads(url.read().decode())
        string_data = ""+str(data)

    client = Client(account_sid, auth_token)
    for i in Number_list:
```

```
        message = client.api.account.messages.create(


        to=str(i),


        from_="+12057513314",


        body="Jacket alert system SEND LAST COORDONATE : --
"+string_data)
    return



while(True):


    print(">>>")
    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)



    try:
            print("Google Speech Recognition thinks you said in English:
-  " + r.recognize_google(audio, language = "en-US"))


    except sr.UnknownValueError:
        print("Google Speech Recognition could not understand audio")
    except sr.RequestError as e:
        print("Could not request results from Google Speech Recognition
service; {0}".format(e))


    try:
        Commande = r.recognize_google(audio, language = "en-US")
    except Exception as e:
        print("Error:  Could not understand  Test Example = Wikipedia")
        Commande="Wikipedia"
```

```
    if(Commande == "send SMS")or(Commande == "SMS"):
        SMS()

    elif(Commande == "emergency"):
        emergency()

    elif(Commande == "Wikipedia"):
        Wikipedia()
    elif((Commande == "Stop")or(Commande == "stop")):
        engine.say("Shuting down")
        engine.runAndWait()
        break
    else:
        pass
print(">>>")
```