Smart Chin Brace for Secondary Dystonia

Project - 3

ME-GY:6933 Advanced Mechatronics – Raspberry Pi and Arduino

Meet Doshi mnd338
Harsh Shah hcs367
Joshua Adewolu ija236

Spring 2020

Abstract

Cervical dystonia is a neurological disorder that causes the muscles in your neck to abnormally contract. Secondary Dystonia develops mainly as the result of environmental factors that provide insult to the brain. A patient can recover very fast using some sensory tricks and also muscle sensors can be used to prevent it from happening in the first place. In this project we aim to develop a backup option in-case the first techniques don't work.

Literature Survey

It is found that some people respond to different sensory tricks than others, this can be a problem for the **smart chin brace,** since it addresses only one sensory trick and though it works for the majority of the people, it might not for everyone. This is a problem for patients who are paralyzed and cannot apply the sensory tricks themselves. They need assistance in helping them recover if the device we made does not work for them. Sending an alert to the patient's close personal so that they can reach out to them and help them was needed. According to our research we found out that people respond to texts more than they to emails or interacting with apps.

Introduction

Cervical dystonia is the most common form of focal dystonia. It is a neurological disorder that causes the muscles in your neck to abnormally contract1. These muscle contractions cause involuntary movements and awkward positions of the head, neck, and sometimes shoulders2. The major forms of Cervical Dystonia head posture include;

Torticollis: Rotation of head, neck and chin towards the side about the X-axis.

Laterocollis: Tilting of the head, neck and chin with the ear touching the shoulder about the Z-axis.

Anterocollis: Falling of head, neck and chin about the Y-axis.

Retrocollis: Upwards movement of the head, neck and chin about the Y-axis.

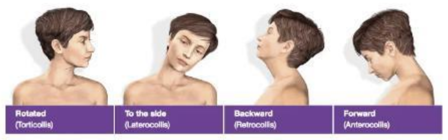


Figure 1: Types of Dystonia (Source: Paci c Neuroscience Institute, 2019)

Muscles usually work in pairs. We have the agonistic muscles and the antago-nistic muscles. Agonist muscles are those that, during any movement, apply the force that allows the movement to occur while Antagonistic muscles are at rest while the movement is being performed11. During neck exion, the sternoclei-domastoid muscle

(SCM) is the agonist muscle while the splenius capitis muscle (SPL) is the antagonistic muscle. During a dystonic movement, both the agonist and antagnostic muscles are co-contracting which causes the strange deviation of the head.

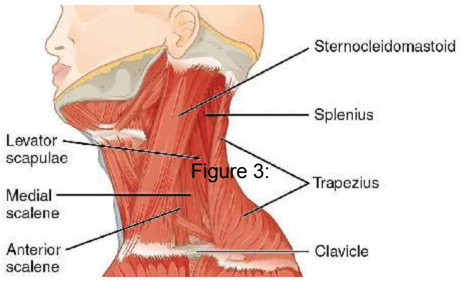


Figure 2: Major Neck Muscles (Source: Thedoctorsofpt, 2020)

Primary and Secondary Dystonia

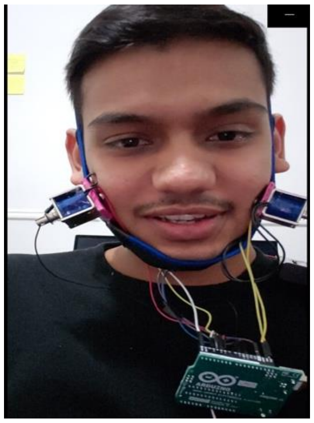
Primary Dystonia is preserved to be genetic while Secondary Dystonia develops mainly as the result of environmental factors that provide insult to the brain. Spinal cord, head, and peripheral injury are also recognized contributors to dystonia6. Many secondary dystonia are associated with approximately 50 neurological and metabolic diseases such as stroke. Some maneuvers called Sensory Trick or Geste Antagonistes were discovered that enable patients to alleviate dystonic movements. It is an episodic and specific maneuver that ameliorates dystonia in a manner that is not easily physiologically perceived as necessary to counteract the involuntary movement. They include motor tricks, imaginary tricks, forcible tricks and reversible sensory tricks. According to re-search carried out by Wissch et al in “Trick Maneuvers in Cervical Dystonia: Investigation of Movement and Touch-Related Changes in Polymyographic Activity", in patients with idiopathic dystonia, sensory tricks may reduce or even relieve muscle activity. For secondary dystonia, regular sensory trick isn’t sufficient enough to alleviate the dystonia and hence we have to use forcible tricks. Forcible tricks are like sensory tricks but necessitates the use of force and are always antagonistic to the direction of the dystonia. Forcible Sensory Tricks:

- Counter pressure applied to the cheeks.
- Holding the chin Massaging the neck.
- Counter pressure on the temple.

Vesper Fe Marie Llaneza Ramos et al in their article, \Tricks in dystonia: ordering the complexity" postulated that the careful analysis of sensory tricks may lead to the fabrication of e ective mimicry devices which was an objective of this project.

Previous projects:

Project 1 Project 2



Using Arduino our aim was to help the patients recover after they had suffered from dystonia.

We further built up on the previous project using Propeller. Our aim was to prevent dystonia before it happens and actuating the previous system as a response to prevent dystonia.

Project 3:

Aim for this project:

Objective for this project is to address the issue of our device failing to work. In this project, we take into consideration the possibility of the Smart Chin Brace being unable to help some patients recover or prevent from Dystonia, and keeping this in mind we aim to add another feature in our device, which would send an emergency message to the close personal of the patient if the he/she has not recovered.

Components:

-Raspberry Pi 3:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera

- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

-Arduino Micro:

- Microcontroller: ATmega32u4
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 20
- PWM Channels: 7
- Analog Input Channels: 12
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega32u4) of which 4 KB used by bootloader
- SRAM: 2.5 KB (ATmega32u4)
- EEPROM: 1 KB (ATmega32u4)
- Clock Speed: 16 MHz

-Transistors: Connecting the Solenoids to 12V source.

-Two 12V Push pull Solenoids

-Six Vibrating Motors

-Diodes

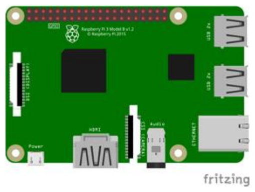
-220 Ohms Resistors

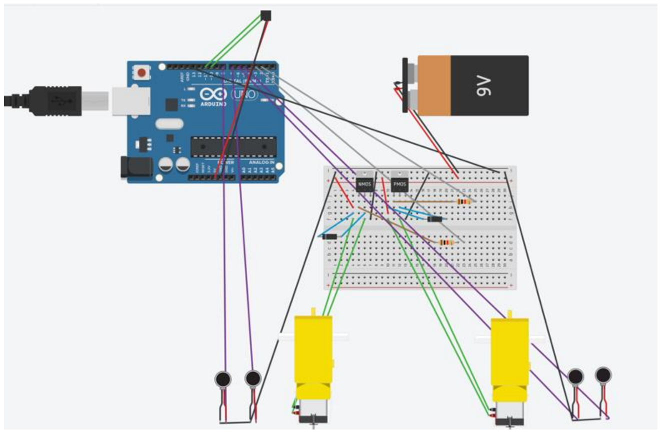
-Memsic 2125 Dual-Axis Accelerometer

BILL OF MATERIAL:

ITEM	QUANTITY	AMOUNT($)
Arduino Micro	1	50
Raspberry Pi	1	70
12v Push Pull solenoid	2	12.5
Chin Brace	1	15
Vibrating motor	6	12
Cables	Lots	2.25
Battery Pack	1	5
Battery	8	0.42
Transistor	2	5
Memsic 2125	1	10
TOTAL		205

Circuit Diagram:





Working:

- The Chin Brace setup is connected to the Arduino, which controls the solenoids actuate depending on the position the neck has tilted to, from the values from the accelerometer.

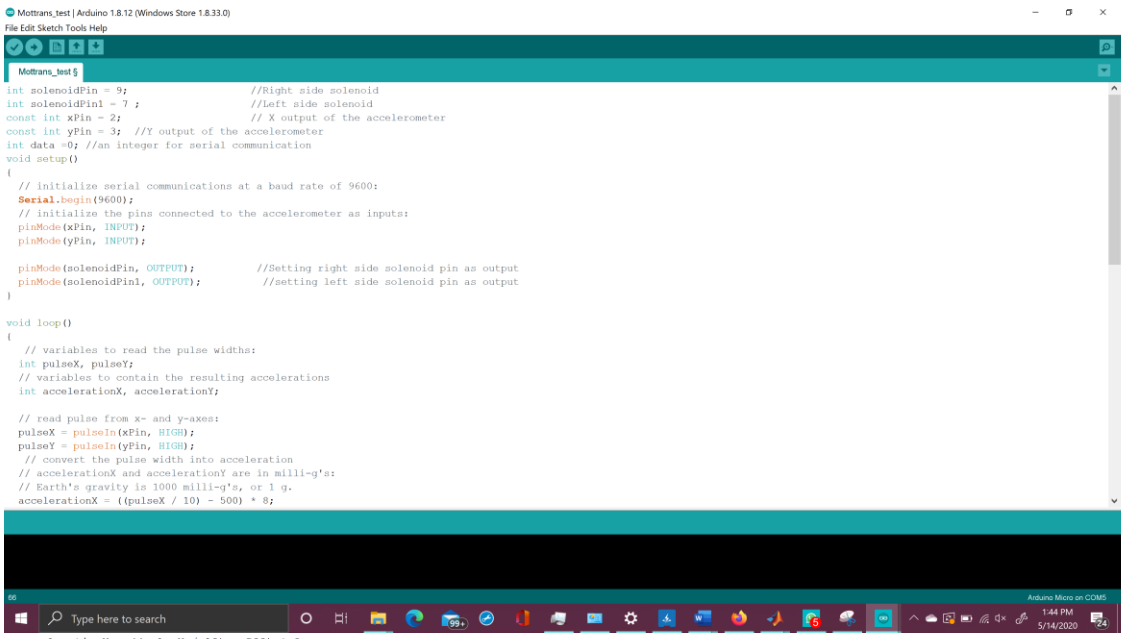
- The Arduino is connected to the Raspberry Pi 3 using a micro USB cable. The code for the Smart Chin Brace to work efficiently is uploaded in the Arduino micro through its IDE installed in the Raspberry Pi.

- The Raspberry Pi then communicates with the Arduino using a python code, where it looks for the print statement of the Arduino's serial monitor.

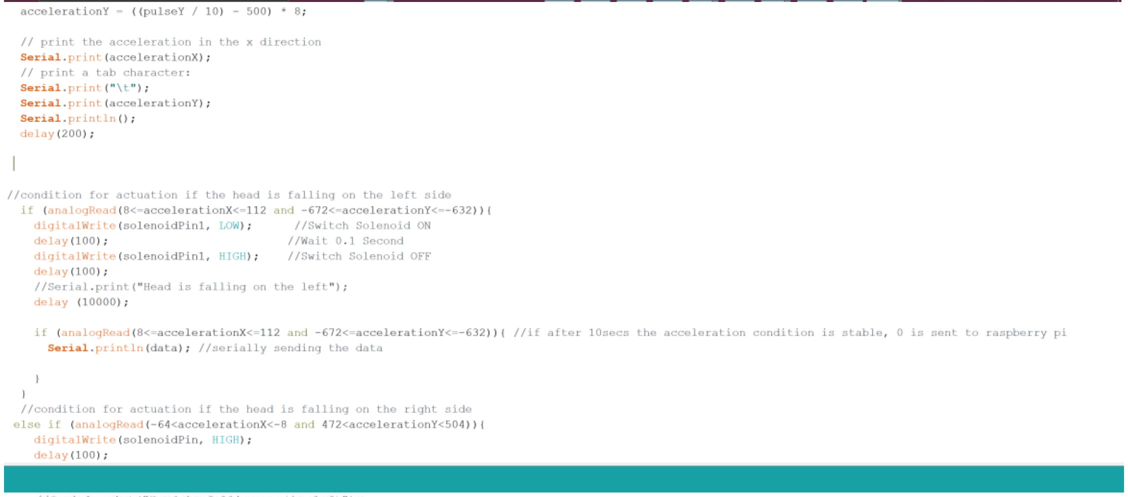
- Whenever the head tilts in either direction a statement is printed in the serial monitor of Arduino. If this statement is being printed for more than 10 seconds, It means that the head is still tilted and that the sensory trick did not help the patient, this gets detected by the Raspberry Pi, using the code in python.

- Once this is detected the Raspberry Pi executes the rest of the code to send the message to the phone numbers added in the emergency contacts of the patient, Saying " EMERGENCY"

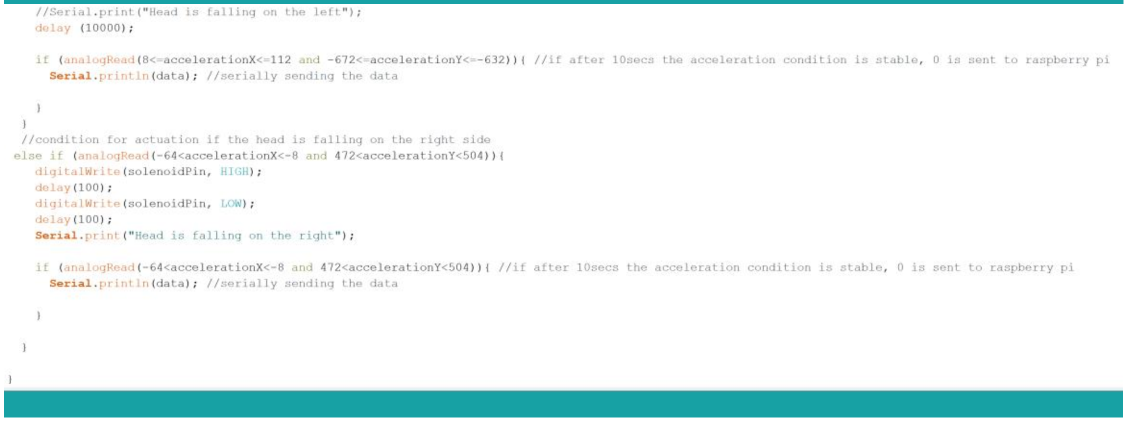
Code:


Mottrans_test | Arduino 1.8.12 (Windows Store 1.8.33.0)
File Edit Sketch Tools Help
Mottrans_test §
66
Arduino Micro on COM5
Type here to search
1:44 PM
5/14/2020

```
int solenoidPin = 9;                    //Right side solenoid
int solenoidPin1 = 7 ;                  //Left side solenoid
const int xPin = 2;                     // X output of the accelerometer
const int yPin = 3;  //Y output of the accelerometer
int data =0; //an integer for serial communication
void setup()
{
  // initialize serial communications at a baud rate of 9600:
  Serial.begin(9600);
  // initialize the pins connected to the accelerometer as inputs:
  pinMode(xPin, INPUT);
  pinMode(yPin, INPUT);

  pinMode(solenoidPin, OUTPUT);          //Setting right side solenoid pin as output
  pinMode(solenoidPin1, OUTPUT);          //setting left side solenoid pin as output
}

void loop()
{
   // variables to read the pulse widths:
  int pulseX, pulseY;
  // variables to contain the resulting accelerations
  int accelerationX, accelerationY;

  // read pulse from x- and y-axes:
  pulseX = pulseIn(xPin, HIGH);
  pulseY = pulseIn(yPin, HIGH);
   // convert the pulse width into acceleration
  // accelerationX and accelerationY are in milli-g's:
  // Earth's gravity is 1000 milli-g's, or 1 g.
  accelerationX = ((pulseX / 10) - 500) * 8;
```



```
  accelerationY = ((pulseY / 10) - 500) * 8;

  // print the acceleration in the x direction
  Serial.print(accelerationX);
  // print a tab character:
  Serial.print("\t");
  Serial.print(accelerationY);
  Serial.println();
  delay(200);



//condition for actuation if the head is falling on the left side
  if (analogRead(8<=accelerationX<=112 and -672<=accelerationY<=-632)){
    digitalWrite(solenoidPin1, LOW);      //Switch Solenoid ON
    delay(100);                          //Wait 0.1 Second
    digitalWrite(solenoidPin1, HIGH);    //Switch Solenoid OFF
    delay(100);
    //Serial.print("Head is falling on the left");
    delay (10000);

    if (analogRead(8<=accelerationX<=112 and -672<=accelerationY<=-632)){ //if after 10secs the acceleration condition is stable, 0 is sent to raspberry pi
      Serial.println(data); //serially sending the data

    }
  }
  //condition for actuation if the head is falling on the right side
 else if (analogRead(-64<accelerationX<-8 and 472<accelerationY<504)){
    digitalWrite(solenoidPin, HIGH);
    delay(100);
```



```
    //Serial.print("Head is falling on the left");
    delay (10000);

    if (analogRead(8<=accelerationX<=112 and -672<=accelerationY<=-632)){ //if after 10secs the acceleration condition is stable, 0 is sent to raspberry pi
      Serial.println(data); //serially sending the data

    }
  }
  //condition for actuation if the head is falling on the right side
 else if (analogRead(-64<accelerationX<-8 and 472<accelerationY<504)){
    digitalWrite(solenoidPin, HIGH);
    delay(100);
    digitalWrite(solenoidPin, LOW);
    delay(100);
    Serial.print("Head is falling on the right");

    if (analogRead(-64<accelerationX<-8 and 472<accelerationY<504)){ //if after 10secs the acceleration condition is stable, 0 is sent to raspberry pi
      Serial.println(data); //serially sending the data

    }

  }

}
```

Future Developments

- We can use Fast Fourier transform to differentiate between voluntary and involuntary muscle activity by taking the least mean square and extracting the mean wave.
- Accommodation of more types of cervical dystonia such as Torticollis, Anterocollis and Retrocollis.
- Sending the daily reports of the patient using the muscle sensor to the doctors for monitoring.
- Adding GPS module and sending the location embedded in the emergency text message.

References

1. Familydoctor.org Editorial. \What Is Cervical Dystonia? - Cervical Dystonia Treatment." Familydoctor.org, 13 Nov. 2018, familydoctor.org/condition/cervical-dystonia/.
2. \Cervical Dystonia." Dystonia Medical Research Foundation, dystonia-foundation.org/what-is-dystonia/types-dystonia/cervical-dystonia/.
3. DystoniaSociety. \Botulinum Toxin Injection." The Dystonia Society, www.dystonia.org.uk/botulinum-toxin-injection.
4. DystoniaSociety. \Denervation." The Dystonia Society, www.dystonia.org.uk/denervation.
5. DystoniaSociety. \Deep Brain Stimulation." The Dystonia Society, www.dystonia.org.uk/deep-brain-stimulation.
6. Deuschl G, Heinen F, Kleedorfer B, et al. Clinical and Polymyographic Investigation of Spasmodic Torticollis. J Neurol. 1992 Jan;239(1):9{15.
7. Ramos V.F.M.L, Karp, B.I. et al. Tricks in dystonia: ordering the complexity. J Neurol Neurosurg Psychiatry. 2014 Sep; 85(9): 987{993.
8. Ochudlo S, Drzyzga K, Drzyzga LR, et al. Various patterns of gestes antagonistes in cervical dystonia. Parkinsonism Related Disorders 2007 Oct;13(7):417{20.
9. Schramm A, Reiners K, Naumann M. Complex mechanisms of sensory tricks in cervical dystonia. Movement Disorders 2004 Apr;19(4):452{8.
10. Wissel J MD, Muller J MD, et al. Trick Maneuvers in Cervical Dysto-nia: Investigation of Movement- and Touch-Related Changes in Polymyographic Activity. Movement Disorders 1999 :994{999.
11. Trifocus. \What Is the Di erence Between Agonist and Antagonist Muscle?" Trifocus Fitness Academy, 22 Jan. 2019, www.trifocus tnessacademy.co.za/blog/di erence-agonist-and-antagonist-muscle/.
12. https://maker.pro/raspberry-pi/tutorial/how-to-connect-and-interface-raspberry-pi-with-arduino
13. https://www.twilio.com/docs/sms/tutorials/how-to-send-sms-messages-python
14. https://core-electronics.com.au/tutorials/solenoid-control-with-arduino.html

Smart Surveillance System for Patients with Paroxysmal dystonia, Epileptic Seizures or Seizure-related Attacks

Doshi Meet · Shah Harsh · Adewolu Joshua

mnd338 · hcs367 · ija236

ME-GY 6933: Advanced Mechatronics – Term Project Report

May 12, 2020

Abstract

Paroxysmal dystonia (historically known as tonic spasms or tonic seizures) is a type of fluctuating dystonia characterized by repetitive and patterned twisting movements and abnormal postures lasting seconds to hours. Epilepsy is associated with a high rate of premature mortality from direct and indirect effects of seizures, epilepsy, and antiseizure therapies. This project aims at developing a smart surveillance system that can detect seizure attacks amongst patients and inform necessary personnel so that appropriate actions can be taken.

Keywords: I2C, Seizures, Paroxysmal, SDA, SCL, SUDEP

Background

Epilepsy is associated with a high rate of premature mortality from direct and indirect effects of seizures, epilepsy, and antiseizure therapies. Sudden unexpected death in epilepsy (SUDEP) is the second leading neurologic cause of total lost potential life-years after stroke. Among neurologic disorders in the United States, sudden unexpected death in epilepsy (SUDEP) is the leading cause of lost years of life after stroke. Because the incidence of epilepsy is greatest among the young, SUDEP, along with status epilepticus, accidents, drownings, and suicide, the most common epilepsy-related causes of death, together lead to substantial premature mortality.

It is estimated that up to 50,000 deaths occur annually in the U.S. from status epilepticus (prolonged seizures), Sudden Unexpected Death in Epilepsy (SUDEP), and other seizure-related causes such as drowning and other accidents. Sudden Unexpected Death in Epilepsy (SUDEP) accounts for 34% of all sudden deaths in children.

Paroxysmal dystonia (historically known as tonic spasms or tonic seizures) is a type of fluctuating dystonia characterized by repetitive and patterned twisting movements and abnormal postures lasting seconds to hours. Paroxysmal dystonia may affect the face, arm, or leg and is often precipitated by tactile stimulation, voluntary movement, loud noise, or hyperventilation.

A split second is all it takes for a patient to lose a life and thus the need for constant monitoring.

Figure 1: A Person Experiencing a Seizure Attack (Source: The Epilepsy Network, 2019)

Someone experiencing seizures cam serious damage or commit unintentional suicide due to the seizures if not attended to.

<u>I2C Communication</u>

I2C allows a single master to control multiple slaves as well as multiple masters control a single or multiple slaves. It uses two wires to transmit data between devices. The line for the master and slave to send and receive data is the Serial Data Line (SDA) while the line that carries the clock signal is the Serial Clock Line (SCL). I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line). Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master.

With I2C, data is transferred in *messages.* Messages are broken up into *frames* of data.
Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame.

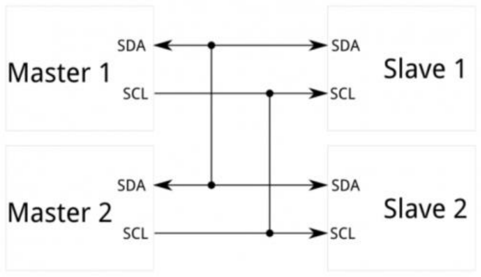


Figure 2: I2C Communication between Master and Slave Devices (Source: Sparkfun, 2020)

Most Camera used for surveillance are static and passive and do not react to changes in environment.

Objective

The objective of this project was to design and develop a smart surveillance system that can be used by to monitor and detect seizures in patients with paroxysmal dystonia attack, epilepsy or other related activities and alert the necessary persons by giving both audio and visual feedback.

Materials and Method

Bill of Materials

The materials used for this project as well as the cost for each of them is shown in Table 1 below;

Table 1: Bill of Materials

S/N	ITEM	QUANTITY	TOTAL AMOUNT($)
1	Arduino Uno	1	50
2	Raspberry Pi 3 B+	1	39.38
3	Raspberry Pi Camera Module V2.1	1	29.90
4	5v Active Buzzer	1	1.82
5	Pushbutton	1	1.69
6	BSS138 Bidirectional Logic Level Converter	1	3.95
7	Cables	Lots	2.25
8	SD Card	1	11.62
9	Battery	1	1.90
	Total		142.51

Methods

- The primary sensor for the surveillance system was the Raspberry Pi Camera. It is mounted on the Raspberry pi B+. It continuously takes images of the environment in its view, saves it temporarily in bitmap format and then compares it with the past image it took to see if a change occurred quickly. If the change exceeds the set threshold, it saves that image as a JPEG format in the directory specified in the code as well as start a 5 second video recording to show what was happening.
- The image taken is timestamped for record and investigatory purposes.
- Apart from taking pictures and videos, the program integrates Twilio REST APIs that allows a specified text message to be sent to specified recipients.

- The Raspberry pi communicates with the Arduino Uno using I2C Communication Protocol. The Raspberry Pi acts as the master while the Arduino Uno is the slave. A logic level converter is used to interface the Raspberry with the Arduino because Raspberry pi 3 B+ operates with maximum of 3.3v while the Arduino uno operates with maximum of 5v.
- Once the Pi camera detects changes in observed environment, the raspberry pi sends a high signal via the SDA line and start and stop bit to the logic level converter, the Arduino then receives the corresponding signal from the high voltage side of the logic level converter and turns on the buzzer. The buzzer turns off when the button is pressed.

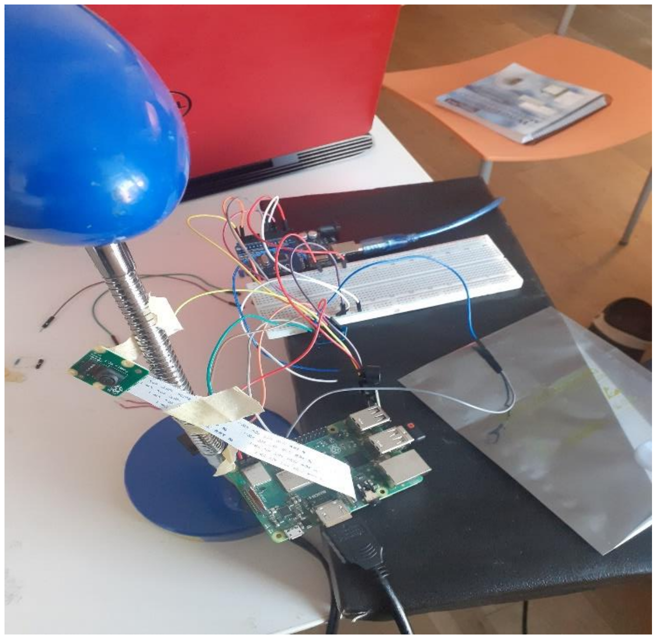


Figure 3: Actual view of Circuit Connection

Results and Discussion

Circuit Diagram

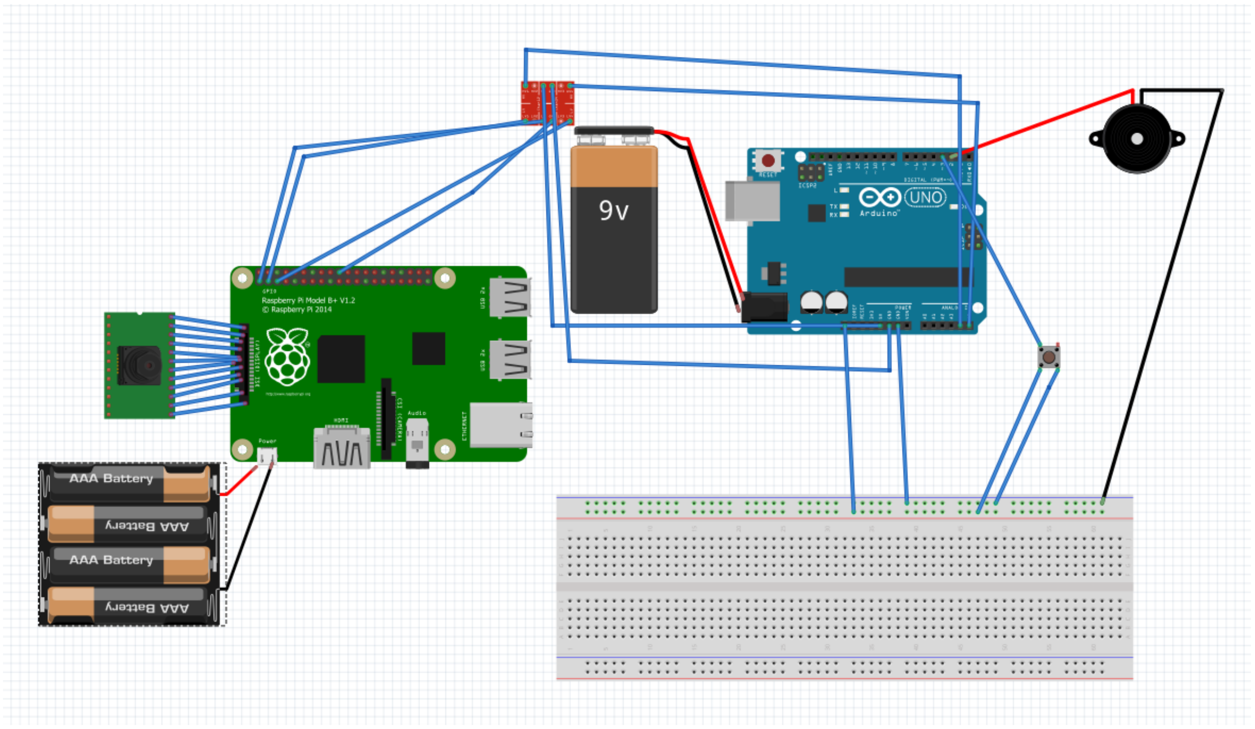
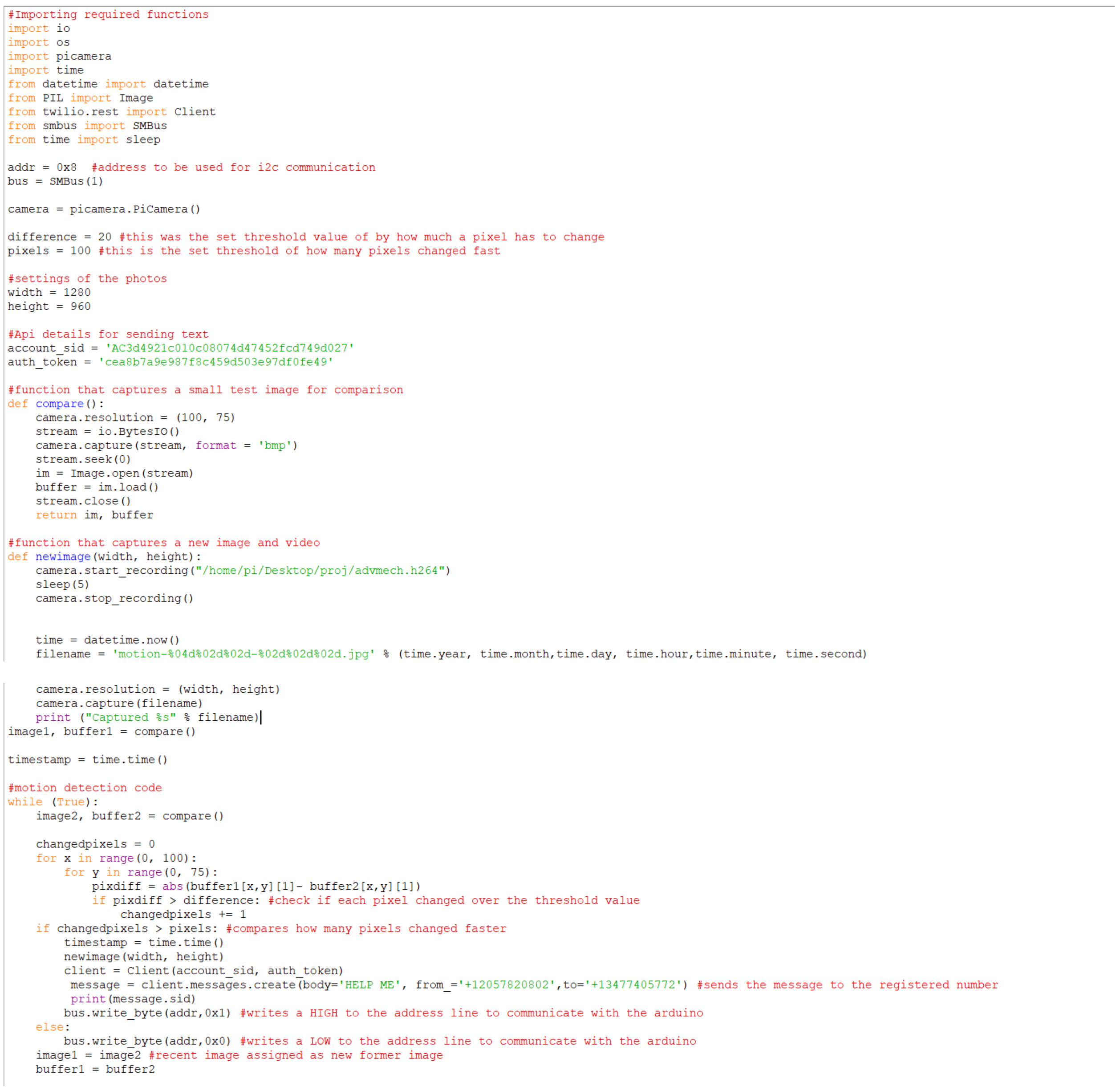


Figure 4: Circuit Diagram for the Device

Codes

The code that was uploaded to the Raspberry Pi is shown in Figure 5 below;

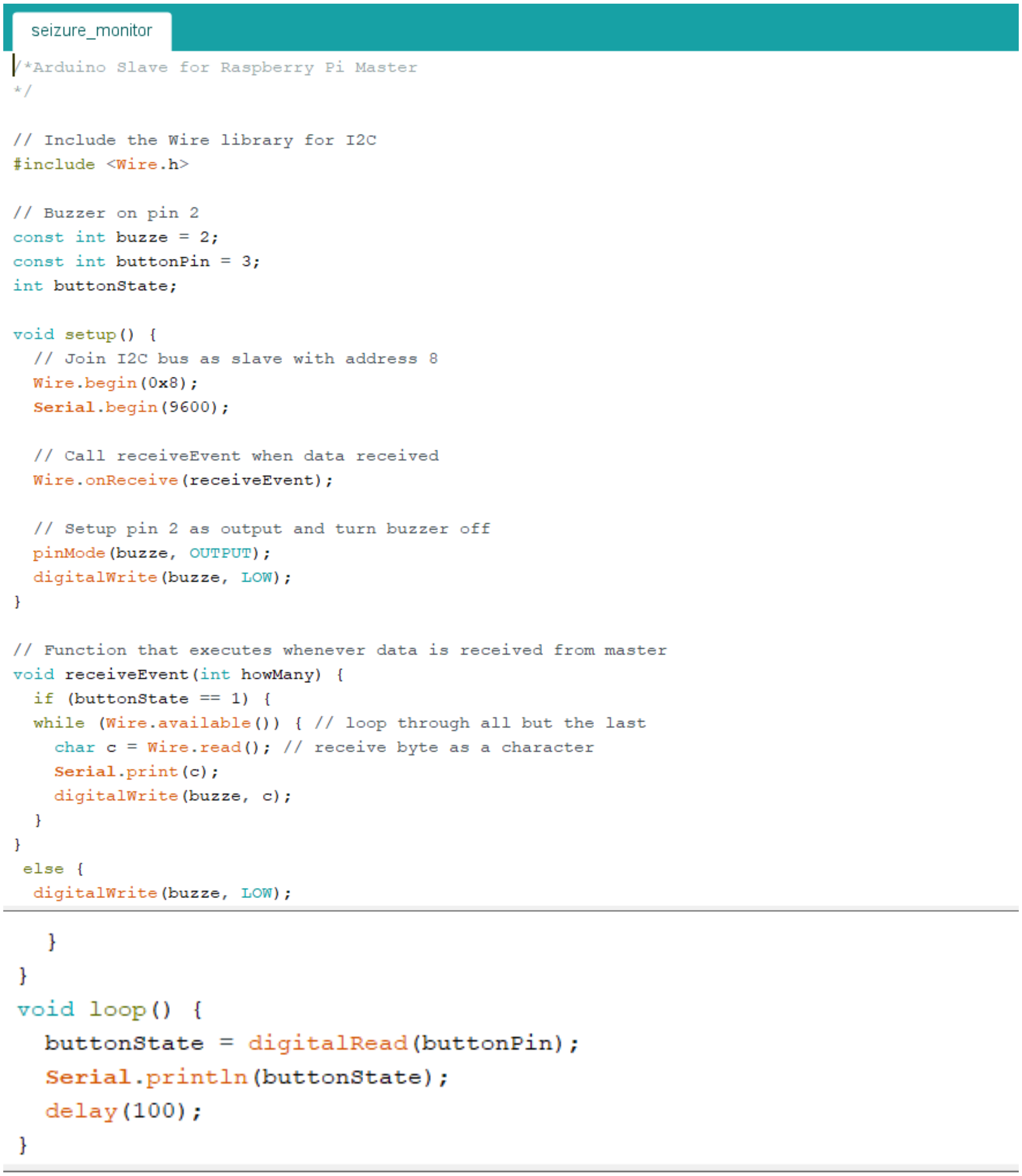


```
#Importing required functions
import io
import os
import picamera
import time
from datetime import datetime
from PIL import Image
from twilio.rest import Client
from smbus import SMBus
from time import sleep

addr = 0x8  #address to be used for i2c communication
bus = SMBus(1)

camera = picamera.PiCamera()

difference = 20 #this was the set threshold value of by how much a pixel has to change
pixels = 100 #this is the set threshold of how many pixels changed fast

#settings of the photos
width = 1280
height = 960

#Api details for sending text
account_sid = 'AC3d4921c010c08074d47452fcd749d027'
auth_token = 'cea8b7a9e987f8c459d503e97df0fe49'

#function that captures a small test image for comparison
def compare():
    camera.resolution = (100, 75)
    stream = io.BytesIO()
    camera.capture(stream, format = 'bmp')
    stream.seek(0)
    im = Image.open(stream)
    buffer = im.load()
    stream.close()
    return im, buffer

#function that captures a new image and video
def newimage(width, height):
    camera.start_recording("/home/pi/Desktop/proj/advmech.h264")
    sleep(5)
    camera.stop_recording()


    time = datetime.now()
    filename = 'motion-%04d%02d%02d-%02d%02d%02d.jpg' % (time.year, time.month,time.day, time.hour,time.minute, time.second)

    camera.resolution = (width, height)
    camera.capture(filename)
    print ("Captured %s" % filename)
image1, buffer1 = compare()

timestamp = time.time()

#motion detection code
while (True):
    image2, buffer2 = compare()

    changedpixels = 0
    for x in range(0, 100):
        for y in range(0, 75):
            pixdiff = abs(buffer1[x,y][1]- buffer2[x,y][1])
            if pixdiff > difference: #check if each pixel changed over the threshold value
                changedpixels += 1
    if changedpixels > pixels: #compares how many pixels changed faster
        timestamp = time.time()
        newimage(width, height)
        client = Client(account_sid, auth_token)
         message = client.messages.create(body='HELP ME', from_='+12057820802',to='+13477405772') #sends the message to the registered number
         print(message.sid)
        bus.write_byte(addr,0x1) #writes a HIGH to the address line to communicate with the arduino
    else:
        bus.write_byte(addr,0x0) #writes a LOW to the address line to communicate with the arduino
    image1 = image2 #recent image assigned as new former image
    buffer1 = buffer2
```

Figure 5: Raspberry Pi Python Code

The code was written in Thonny IDE.

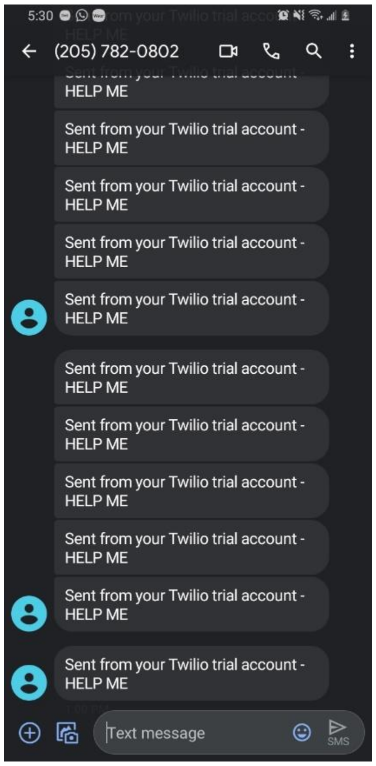
The code that was uploaded to the Arduino Uno is shown in Figure 6 below;



```
/*Arduino Slave for Raspberry Pi Master
*/

// Include the Wire library for I2C
#include <Wire.h>

// Buzzer on pin 2
const int buzze = 2;
const int buttonPin = 3;
int buttonState;

void setup() {
  // Join I2C bus as slave with address 8
  Wire.begin(0x8);
  Serial.begin(9600);

  // Call receiveEvent when data received
  Wire.onReceive(receiveEvent);

  // Setup pin 2 as output and turn buzzer off
  pinMode(buzze, OUTPUT);
  digitalWrite(buzze, LOW);
}

// Function that executes whenever data is received from master
void receiveEvent(int howMany) {
  if (buttonState == 1) {
  while (Wire.available()) { // loop through all but the last
    char c = Wire.read(); // receive byte as a character
    Serial.print(c);
    digitalWrite(buzze, c);
  }
}
 else {
  digitalWrite(buzze, LOW);
  }
}
void loop() {
  buttonState = digitalRead(buttonPin);
  Serial.println(buttonState);
  delay(100);
}
```

Figure 6: Arduino Code

After doing all connections and uploading the code the code to the microcontroller, we tested the device and we got the designed outcome for this stage of the project. Once person in the view of the camera changed position at a fast pace, the buzzer came up and a text message was sent to the specified recipient. A snapshot of the text messaged received by the recipient is shown in Figure 7 below;


5:30

(205) 782-0802

HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Sent from your Twilio trial account - HELP ME

Figure 7: Picture of the text message received

Also, a picture was taken and a 5 second video recorded. The picture was timestamped as shown in Figure 8 below;

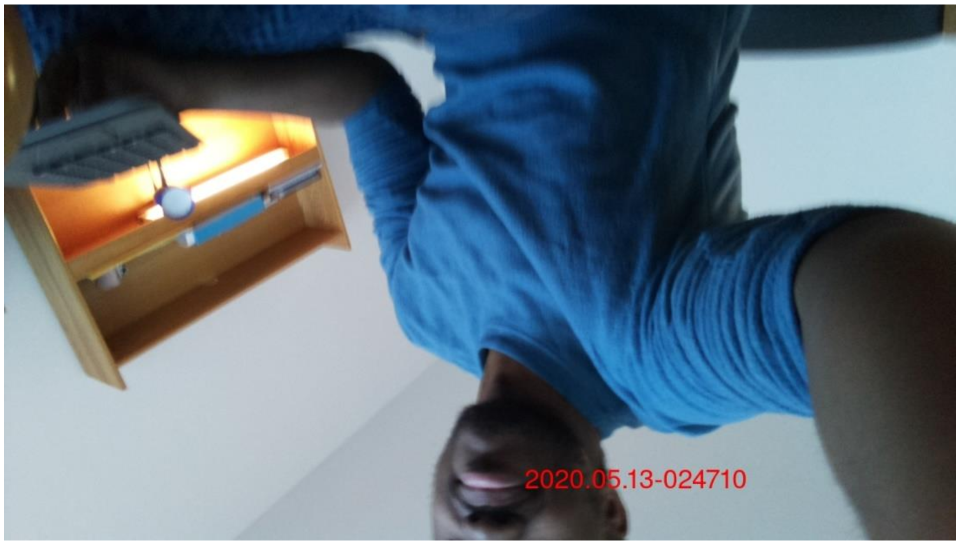


Figure 8: Time-stamped image taken by the camera

Future Developments

This is the beginning stage of this project and some false alarms can be given and as such, an upgrade to this project is to use OpenCV to increase the effectiveness of the seizures and dystonic movements and reduce detecting of other non-dystonic or seizure attacks.

Another upgrade will be to send short videos of the monitored personnel via social media like WhatsApp to the specified recipients so they can have a better idea of the situation and respond appropriately.

References

1. Devinsky O, Spruill T, et al. Recognizing and preventing epilepsy-related mortality: A call for action. Neurology. 2016 Feb 23; 86(8): 779–786.

2. Shneyder N, Harris M.K, Minagar A. Movement disorders in patients with multiple sclerosis. Hyperkinetic Movement Disorders. 2011. Vol. 100

3. TheEpilepsyNetwork. "Facts & Statistics." The Epilepsy Network, www.theepilepsynetwork.com/facts-and-statistics/

4. https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/