NEW HORIZON
COLLEGE OF ENGINEERING
AUTONOMOUS COLLEGE Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade

GENERALIZED EPILEPTIC SEIZURE ALERT SYSTEM

A MINI PROJECT REPORT

Submitted by

KAMALA VENNELA VASIREDDY (1NH18EC047)

MELITA ROSE G (1NH18EC071)

RAKESH M (1NH18EC091)

SURAJ SURESH (1NH18EC110)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled "**Generalized Epileptic-Seizure Alert System**" carried out by **Kamala Vennela Vasireddy (1NH18EC047), Melita Rose G (1NH18EC071), Rakesh M (1NH18EC091) and Suraj Suresh (1NH18EC110)** bonafide students of Electronics and Communication Department, New Horizon College of Engineering, Bangalore. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Project guide	HOD
Dr. Sanjeev Sharma	Dr. Sanjeev Sharma
HOD	B. Tech, M.Tech,PhD
Dept of ECE	NHCE
NHCE	
-------------------------	--------------------

External Viva

Name of Examiner   Signature with Date

1.

2

ABSTRACT

Epilepsy is a central nervous system (neurological) disorder due to abnormal brain activity, causing seizures or periods of unusual behavior, sensations, and loss of awareness. Epilepsy is a non-curable disorder and the seizures can occur anytime. Most occurrences of epileptic seizures prove to be fatal due to delays in detection or lack of prompt medical response. Most cases require full time caretakers monitoring the well-being of the patient, failing which the patient's life is in immense danger. This proposed device eliminates the above mentioned issues and thereby reduces the burden on the caretaker.

Leading a normal life becomes a difficult-task for people with epilepsy. Their daily activities require someone constantly monitoring them to avoid cases of negligence, making them dependent. The most crucial of all challenges being, the seizure event can be fatal if not addressed at the earliest.

The primary requirement of the problem is to design a device which detects seizure movements; the other specications that the developed device would include are indicating sleep patterns of the user, it should be self-corrective, highly accurate and reliable. The devices available are expensive and their functionality is restricted to just detecting the epileptic seizures. These issues have been resolved in the idea we have used to develop the device.

Our idea is simple and cost effective which helps in the detection of epileptic seizures that are not usually detected and reported at the earliest. The device functions as a ‘portable seizure detection-alerting systems’. The patient with the help of our device can become independent and there is no requirement for continuous-monitoring.

ACKNOWLEDGEMENT

The satisfaction that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **Dr. Sanjeev Sharma** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

KAMALA VENNELA VASIREDDY	**1NH18EC047**
MELITA ROSE G	**1NH18EC071**
RAKESH M	**1NH18EC091**
SURAJ SURESH	**1NH18EC110**

TABLE OF CONTENTS

TOPIC **PAGE NO**

LIST OF FIGURES

LIST OF TABLES

CHAPTER 01

INTRODUCTION

Epilepsy is a central nervous system (neurological) disorder in which brain activity becomes abnormal, causing seizures or periods of unusual behavior, sensations, and sometimes loss of awareness. Seizures are of many kinds with the most prevalent being generalized tonic-clonic seizures. A generalized tonic-clonic seizure, sometimes called a grand mal seizure, is a disturbance in the functioning of both sides of the brain. This disturbance is caused by electrical signals spreading through the brain inappropriately. Often this will result in signals being sent to the muscles, nerves, or glands. The spread of these signals in the brain can make one lose consciousness and have severe muscle contractions.

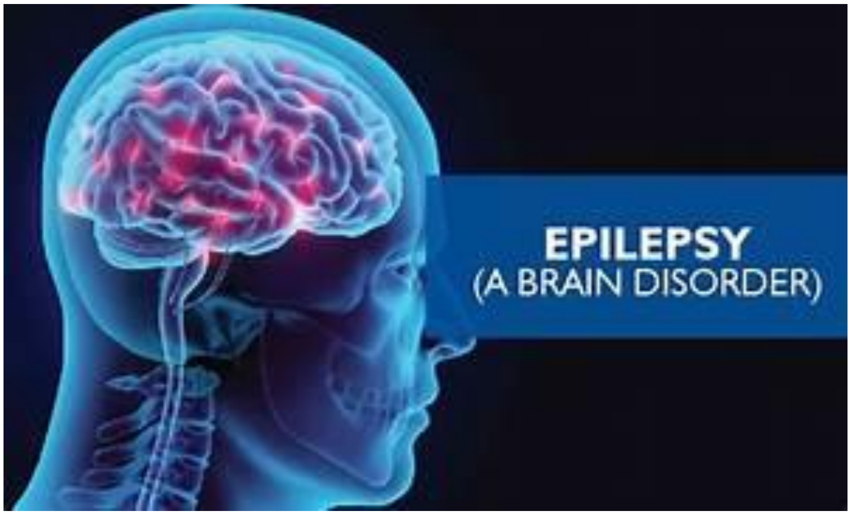


Fig 1(a): Introduction picture

Of the 70 million persons with epilepsy (PWE) worldwide, nearly 12 million PWE are expected to reside in India; which contributes to nearly one-sixth of the global burden. Epilepsy often manifests itself in childhood and is a cause of immense concern for the parents who are often kept on their toes anticipating the next seizure event. This causes psychological and health concerns amongst the primary caregivers who often stay awake through the night in anticipation.

The aim of this project is to develop a battery powered, low cost, portable generalized tonic-clonic seizure alert device which can notify nearby family members or caregivers when a seizure occurs through alarms, phone calls or text alerts, depending on the device. A caregiver can then help the person during and after the seizure. If breathing or other problems occur, they can call for medical help. They may also be able to give rescue medications or call for an ambulance if the seizure lasts too long or the person has repeated seizures.

The project aims at using Inertial Measurement Unit/tri axial accelerometer as the primary sensing unit coupled with machine learning techniques to detect abnormal motions akin to seizures in the patient's body. The proposed instrument shall also consist of heartbeat monitoring system for additional analytics related to sleep quality monitoring.

The project has significant hardware and software challenges which includes accurate measurement and classification of jerking motions, continuous low power heart rate monitoring, signal processing and machine learning algorithms at a very low power budget. The proposed device will reduce burden on the primary caregivers or family members who often spend sleepless nights anticipating the seizure even

CHAPTER 02

LITERATURE SURVEY

Title of the paper	**Author & Year of Publication**	**Techniques /Algorithm used**	**Limitation**
Epileptic Seizure Detection Using a Wrist-Worn Triaxial Accelerometer	Jussa Klapuri (2013)	Possibilities of detecting severe epileptic seizures by using just a triaxial accelerometer embedded into a watch. The approach is utilizing four kinds of accelerometry datasets recorded by a single wrist-worn accelerometer by three different test subjects and its analyses.	The DPS algorithm implemented in this thesis has potential for much more analysis involving more activities.
Artificial Intelligence and Computational Approaches for Epilepsy	Sora An, Chaewon Kang, Hyang Woon Lee (2020)	AI-based approach to generate a computational machine performing specific task – detecting a seizure	Bulky sensors used and complex algorithm
Detection of generalized tonic–clonic seizures by a wireless wrist accelerometer: A prospective, multicenter study	Sandor Beniczky, Tilman Polster Helle Hjalgrim (2013)	A wireless accelerometer sensor is used for detecting generalized tonic clonic seizure.	A single parameter might produce many false alarms
Detection of Light Sleep Periods Using an Accelerometer Based Alarm System	Egemen Turkyilmaz, Alper Akgul, Erkan Bostanci,Mehmet Serdar Güzel	Sleep classification using accelerometer and programmable alarm system	The values obtained on analyzing are not that accurate

Tab 2.1 : Literature survey

CHAPTER 03

PROPOSED METHODOLOGY

Most common approaches to detect seizure use EEG (electroencephalography) and EMG (electromyography) signals.

Our approach of detecting seizures is mainly dependent on three parameters :

1) ACCELEROMETER SIGNAL
2) HEART RATE AND OXIMETRY
3) SKIN CONDUCTIVITANCE

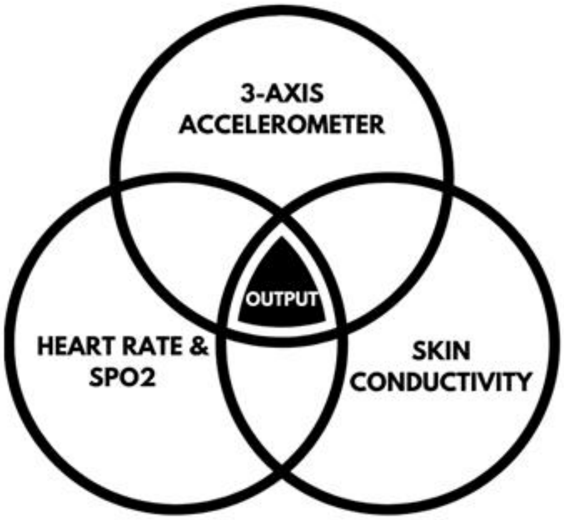


Fig 3(a): Parameters considered

How does seizure detection work?

Seizures can be detected by monitoring the muscle and brain activities, heart rate, oxygen level, artificial sounds, or visual signatures through EEG, EMG, ECG, motion, or audio/video recording on the human head and body.

The parameters that we are considering as said earlier are the motion, heart rate, blood oxygen and skin conductance. A combined output of all these parameters helps us detect the seizure.

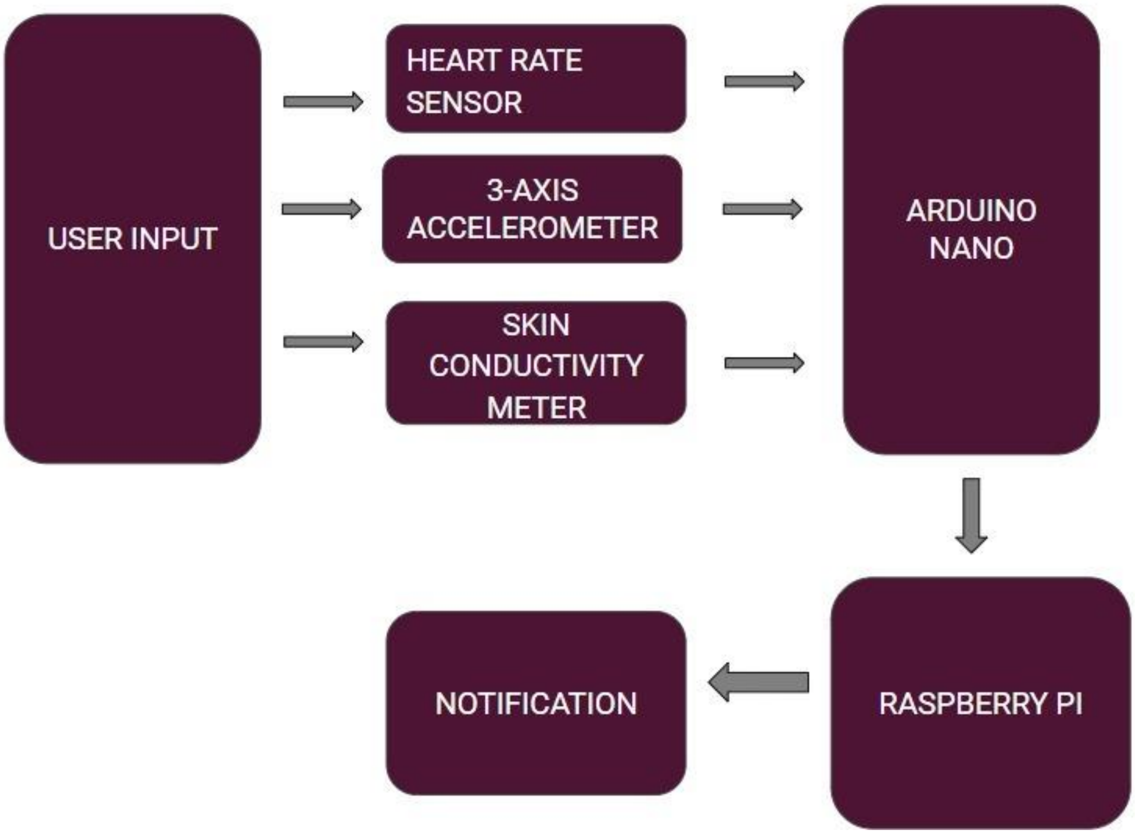


Fig 3(b): Basic Block diagram of working

Why this approach?

Our main objective is to detect a probable seizure, alert the caretaker and make that person independent. Even if accuracy takes a hit because of not considering EEM and EMG signals, the device serves its purpose of cost effective seizure detection. The accuracy of detection can be improved over time by taking user input for wrong detections and correcting it in the algorithm. The algorithm is trained from datasets containing sensor readings of seizures.

When a similar pattern as that of a seizure is detected by the model, it outputs a logic 1. The algorithms used are SVM and KNN. Both the algorithm’s outputs are combined to get better accuracy.

Why no EMG sensor?

A good quality EMG (Electromyography) sensor that most seizure detection methods use costs more than 2000 INR, but one of the objectives was to make the device cost less than that. Moreover data handling is difficult as it is noisy and complex (motion artifacts, static charges caused due to relative motion between the skin and the electrodes). EMG sensors are also quite bulky as they consist of three electrodes and the reference electrode needs to be connected to a bony area which doesn't have any muscle activity. The placement of the sensor, how it is attached, affects its output. All these make it difficult to fabricate as a band, drives up the cost and makes our device a lot more complex.

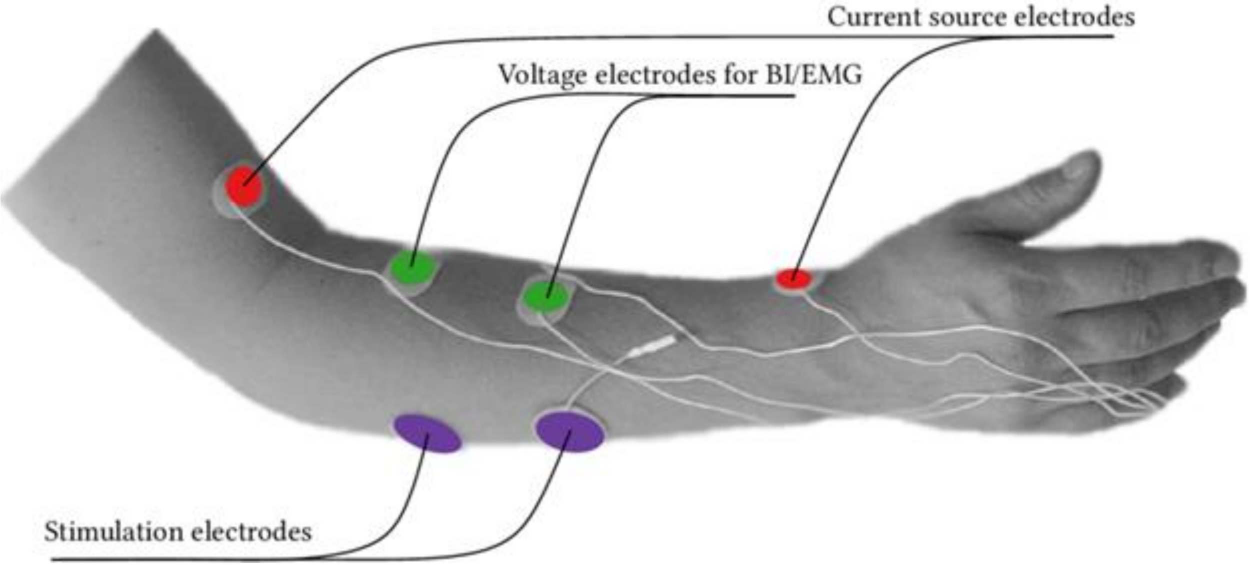


Fig 3(c): EMG sensor

GENERAL PICTURE OF THE PROJECT

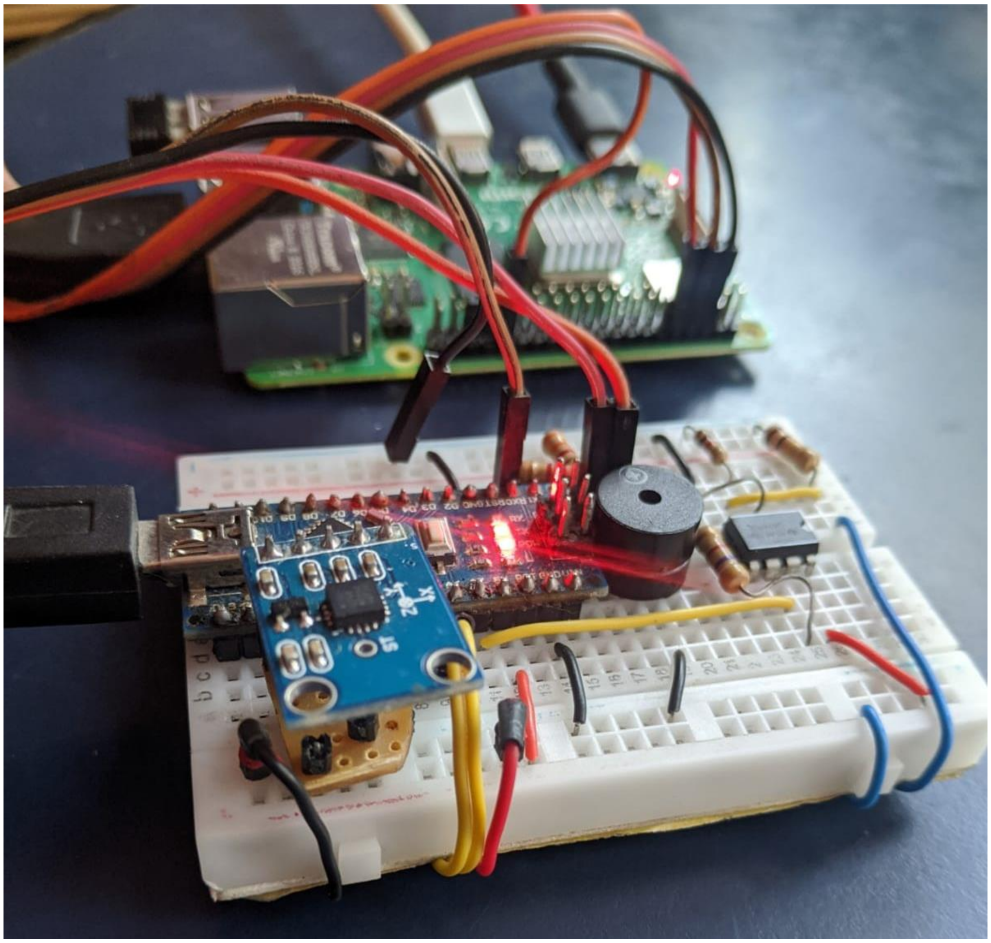


Fig 3(d): Picture of the project

CHAPTER 04

PROJECT DESCRIPTION

Our device consists of a data collection unit (Arduino microcontroller), a data processing unit (Raspberry Pi single-board computer), and the sensors (accelerometer, heart rate sensor and skin conductivity meter).

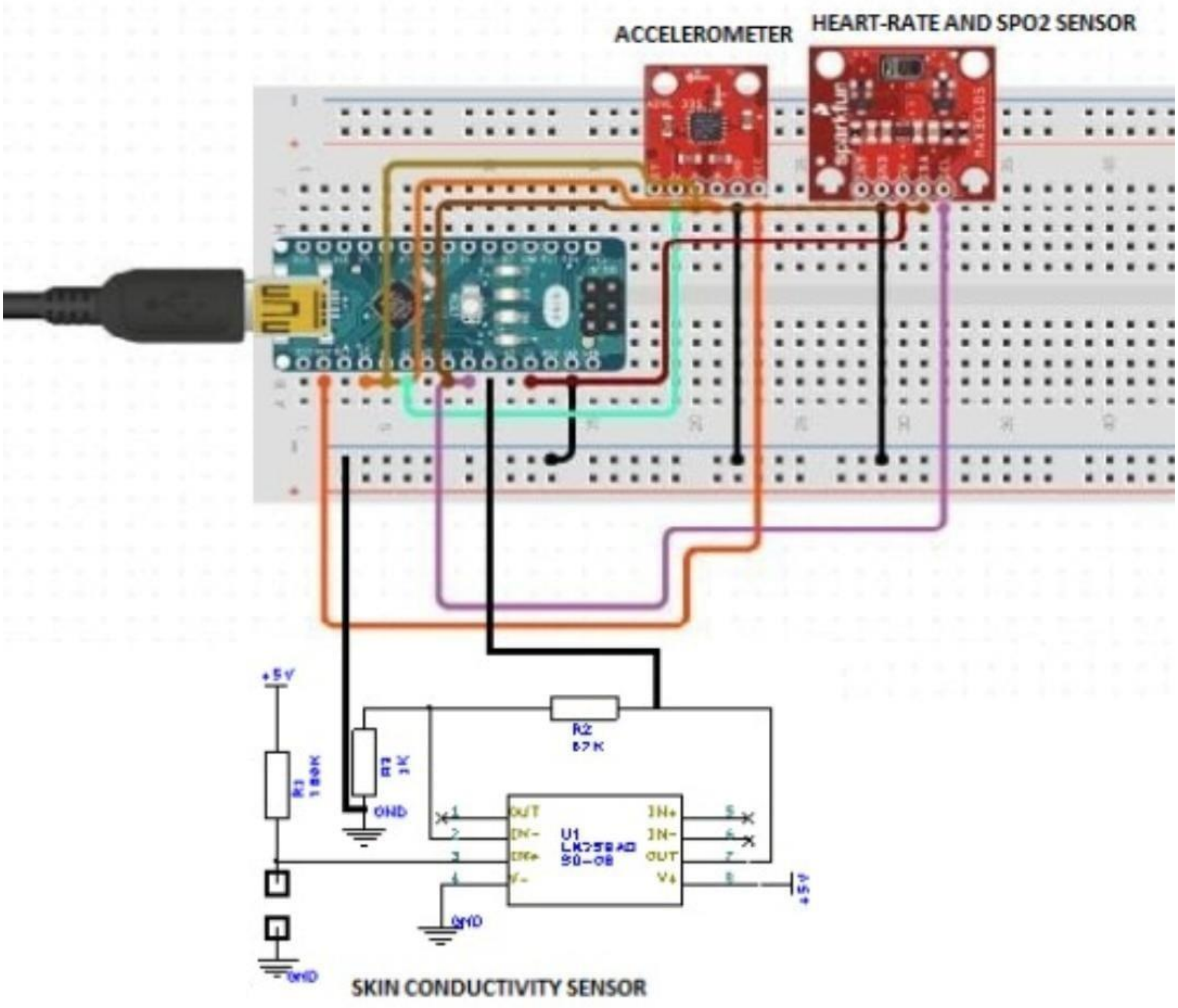
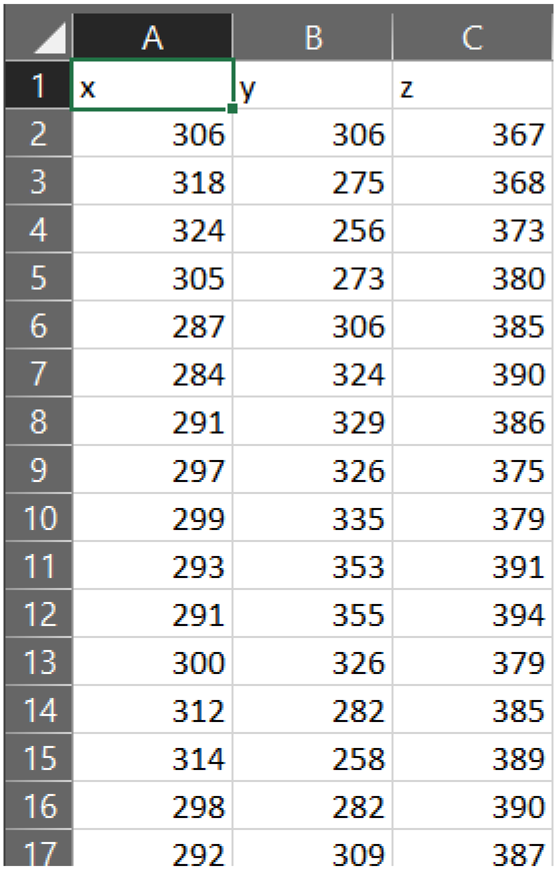


Fig 4(a): Circuit Diagram

Well, to begin with, we collected data from the sensors using Arduino to create the datasets (CSV files). The data was sampled at 1000 samples per second at a baud rate of 9600. The dataset consisted of both seizurely and non-seizurely data.



	A	B	C
1	x	y	z
2	306	306	367
3	318	275	368
4	324	256	373
5	305	273	380
6	287	306	385
7	284	324	390
8	291	329	386
9	297	326	375
10	299	335	379
11	293	353	391
12	291	355	394
13	300	326	379
14	312	282	385
15	314	258	389
16	298	282	390
17	292	309	387

Fig 4(b): Accelerometer dataset format

Using fourier transform this data is converted into frequency domain and models are trained using supervised classification method to detect similar patterns.

After the model is trained and the code is running, when a similar pattern of seizurely data is received, the models predict the output. Depending on the outputs from both the models, the accuracy of the prediction is estimated. This combined output is then used to turn on a buzzer (to alert nearby people) and send an alert notification to the caretaker. In case the detection was wrong and it actually wasn't a seizure, user feedback is taken to modify the dataset and retrain the algorithm.

The communication between the Arduino and Raspberry Pi happens serially through SPI (Serial Peripheral Interface). The code is written in Python programming language and the algorithms used are KNN(K-Nearest Neighbors) and SVM(Support Vector Machine).

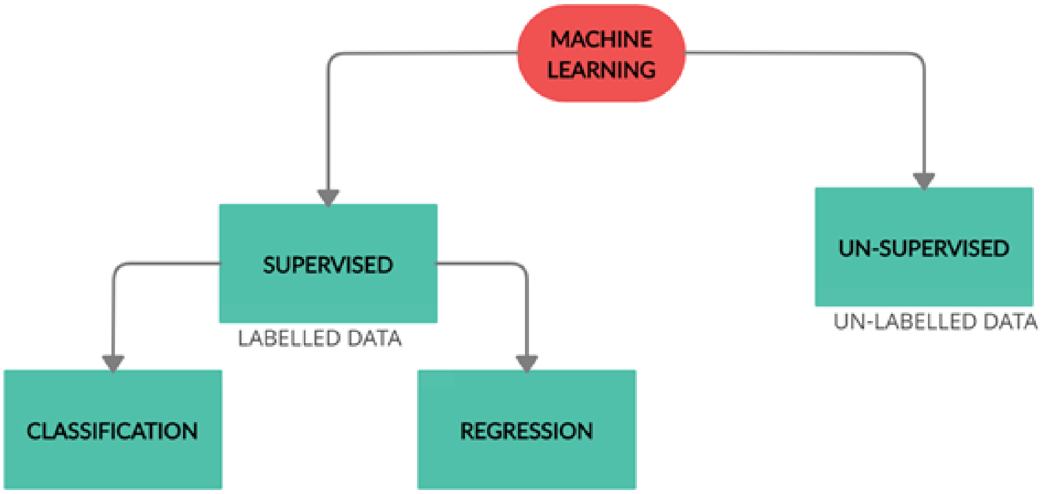
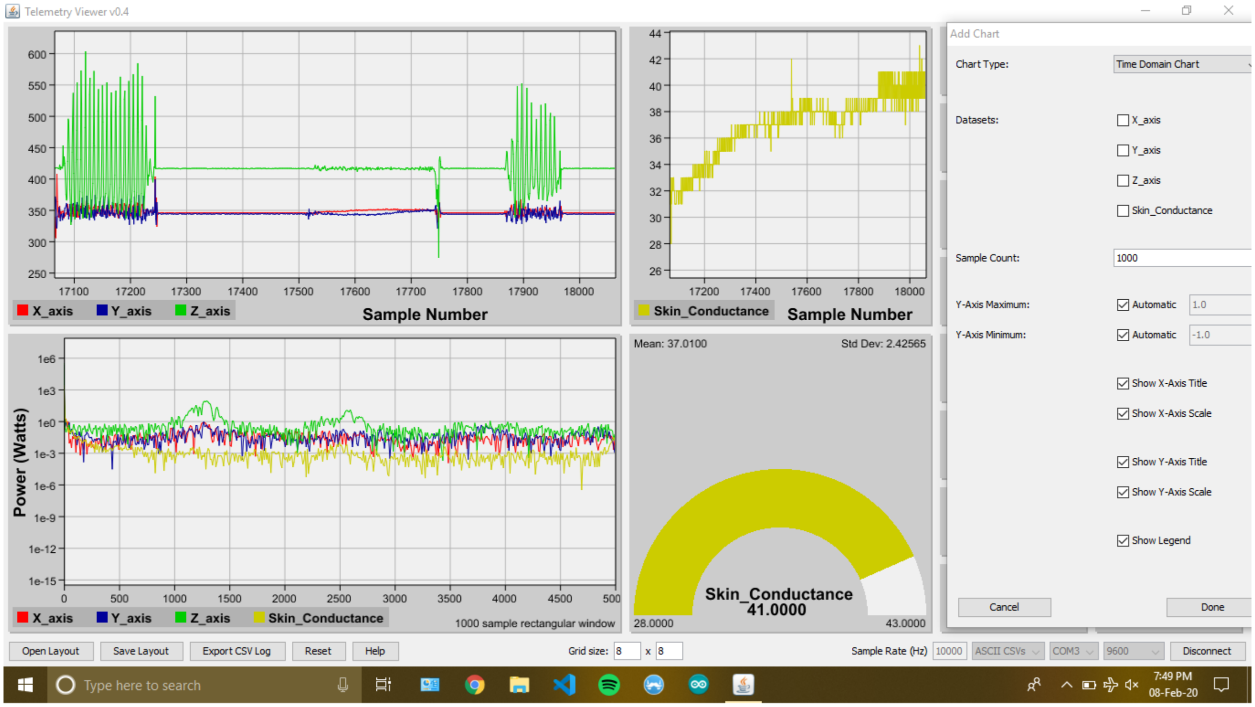


Fig 4(c): Machine learning Classification

The sensor data is visualized and logged onto CSV files using a telemetry viewer software.



Add Chart

Chart Type: Time Domain Chart

Datasets:
☐ X_axis
☐ Y_axis
☐ Z_axis
☐ Skin_Conductance

Sample Count: 1000

Y-Axis Maximum: ☑ Automatic 1.0

Y-Axis Minimum: ☑ Automatic -1.0

☑ Show X-Axis Title

☑ Show X-Axis Scale

☑ Show Y-Axis Title

☑ Show Y-Axis Scale

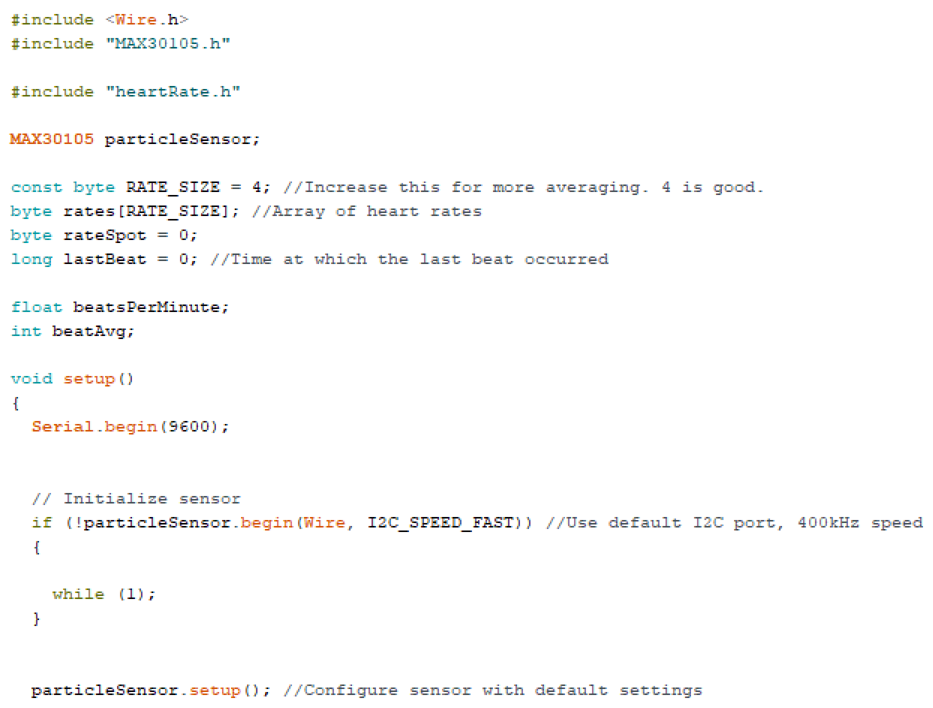
☑ Show Legend

Grid size: 8 x 8

Sample Rate (Hz) 10000 ASCII CSVs COM3 9600

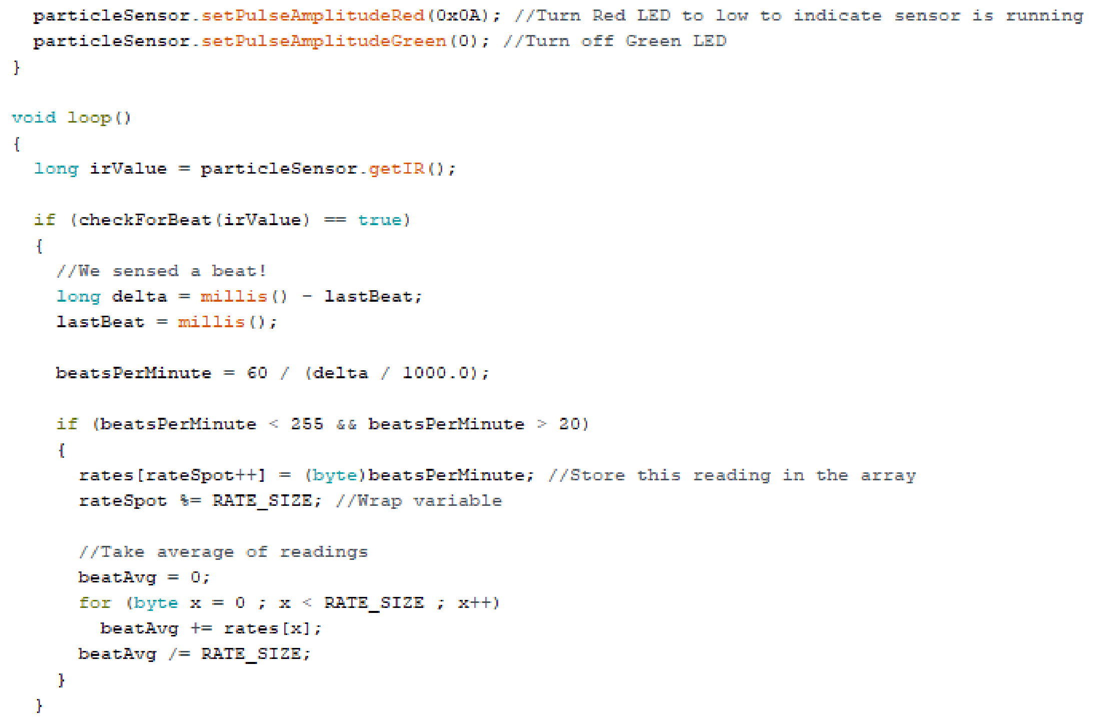
Fig 4(d): Data Visualization

C code for Arduino:



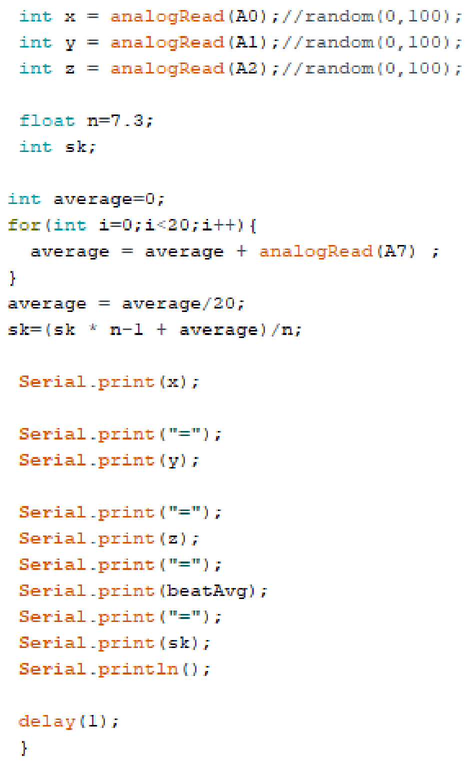
```
#include <Wire.h>
#include "MAX30105.h"

#include "heartRate.h"

MAX30105 particleSensor;

const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred

float beatsPerMinute;
int beatAvg;

void setup()
{
  Serial.begin(9600);


  // Initialize sensor
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
  {

    while (1);
  }


  particleSensor.setup(); //Configure sensor with default settings
```

Fig 4(e): Arduino code-1



```
  particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
}

void loop()
{
  long irValue = particleSensor.getIR();

  if (checkForBeat(irValue) == true)
  {
    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();

    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute < 255 && beatsPerMinute > 20)
    {
      rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
      rateSpot %= RATE_SIZE; //Wrap variable

      //Take average of readings
      beatAvg = 0;
      for (byte x = 0 ; x < RATE_SIZE ; x++)
        beatAvg += rates[x];
      beatAvg /= RATE_SIZE;
    }
  }
```

Fig 4(f): Arduino code-2



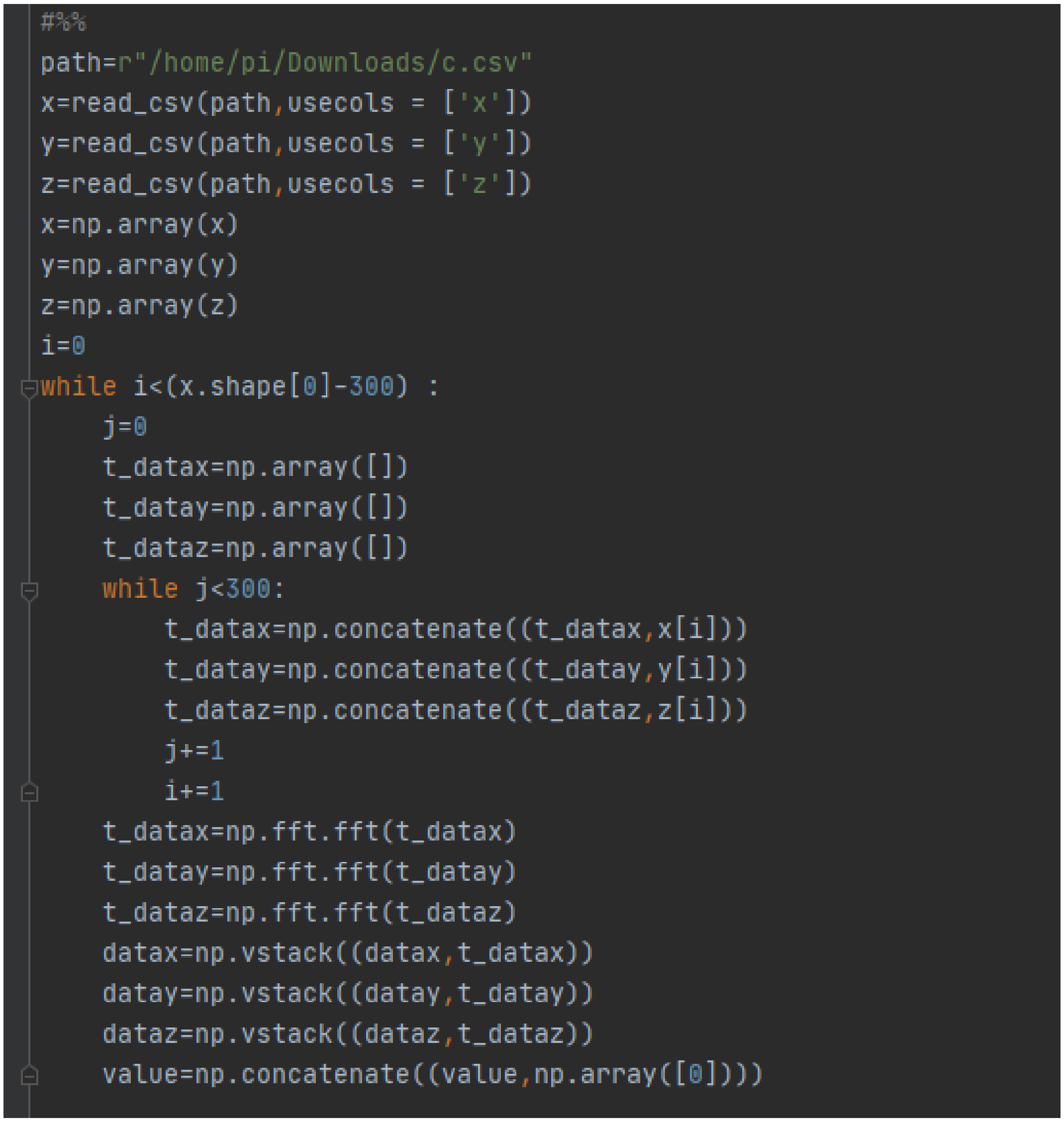
```
 int x = analogRead(A0);//random(0,100);
 int y = analogRead(A1);//random(0,100);
 int z = analogRead(A2);//random(0,100);

 float n=7.3;
 int sk;

int average=0;
for(int i=0;i<20;i++){
  average = average + analogRead(A7) ;
}
average = average/20;
sk=(sk * n-1 + average)/n;

 Serial.print(x);

 Serial.print("=");
 Serial.print(y);

 Serial.print("=");
 Serial.print(z);
 Serial.print("=");
 Serial.print(beatAvg);
 Serial.print("=");
 Serial.print(sk);
 Serial.println();

 delay(1);
 }
```

Fig 4(g):Arduino code-3

Code explanation: Libraries are included to interface the heart rate sensor. The Accelerometer and Skin conductivity meter are analog sensors which are interfaced by connecting them to analog pins and reading them using the analogRead() function whereas the heart rate sensor works over I2C protocol and the MAX30102 library reads its output. The values are then filtered and sent to the serial port(RX0 and TX1) using SPI

Python Codes:

For extracting data from CSV file, applying fourier transform and arranging it as a matrix

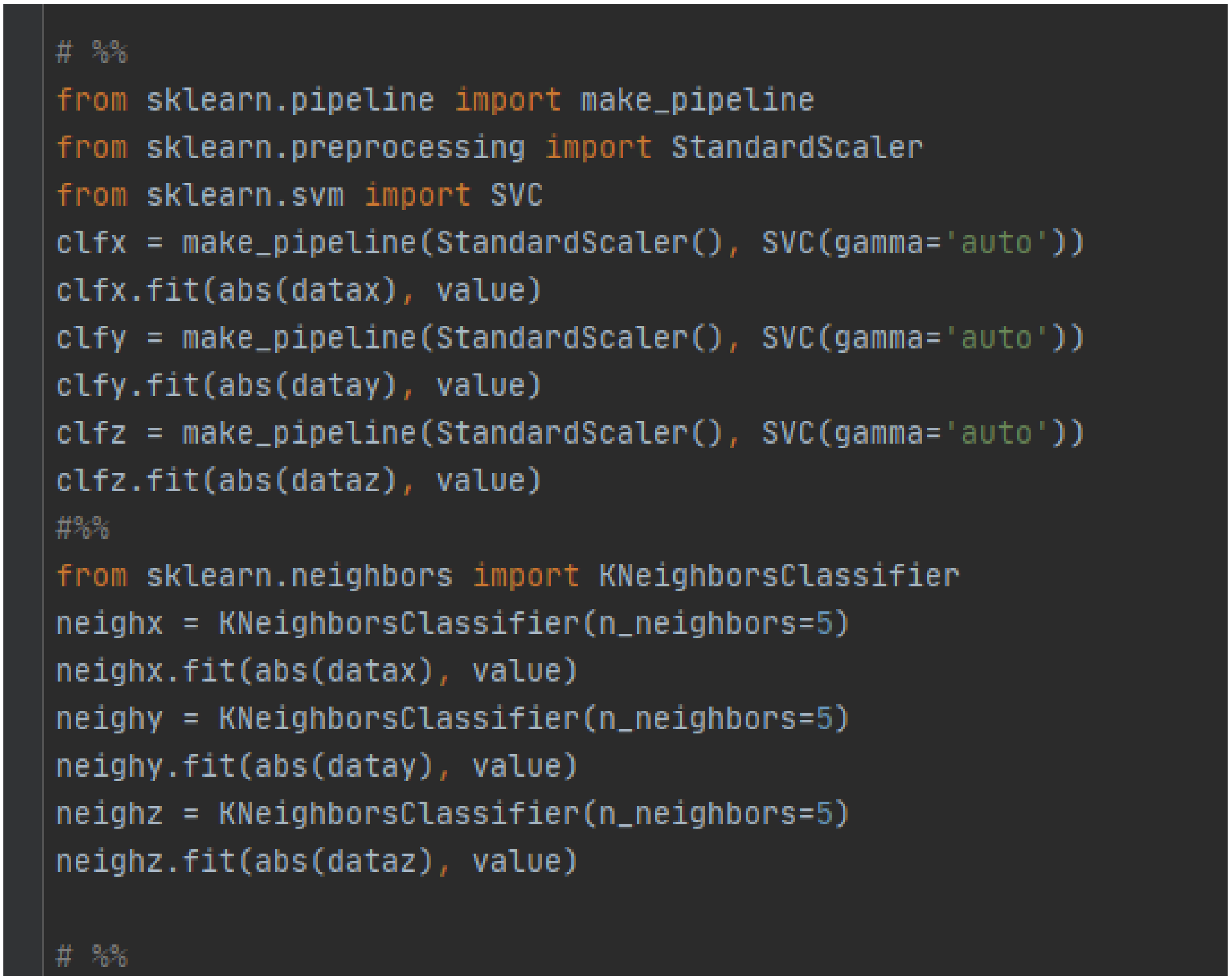


```
#%%
path=r"/home/pi/Downloads/c.csv"
x=read_csv(path,usecols = ['x'])
y=read_csv(path,usecols = ['y'])
z=read_csv(path,usecols = ['z'])
x=np.array(x)
y=np.array(y)
z=np.array(z)
i=0
while i<(x.shape[0]-300) :
    j=0
    t_datax=np.array([])
    t_datay=np.array([])
    t_dataz=np.array([])
    while j<300:
        t_datax=np.concatenate((t_datax,x[i]))
        t_datay=np.concatenate((t_datay,y[i]))
        t_dataz=np.concatenate((t_dataz,z[i]))
        j+=1
        i+=1
    t_datax=np.fft.fft(t_datax)
    t_datay=np.fft.fft(t_datay)
    t_dataz=np.fft.fft(t_dataz)
    datax=np.vstack((datax,t_datax))
    datay=np.vstack((datay,t_datay))
    dataz=np.vstack((dataz,t_dataz))
    value=np.concatenate((value,np.array([0])))
```

Fig 4(h): Python code-1

Explanation: The received data is divided into blocks of 300 values and arranged in a single dimension Numpy array. Fast Fourier Transform is then done to data and the data is arranged as a multi-dimensional Numpy array

For training the Machine learning models:

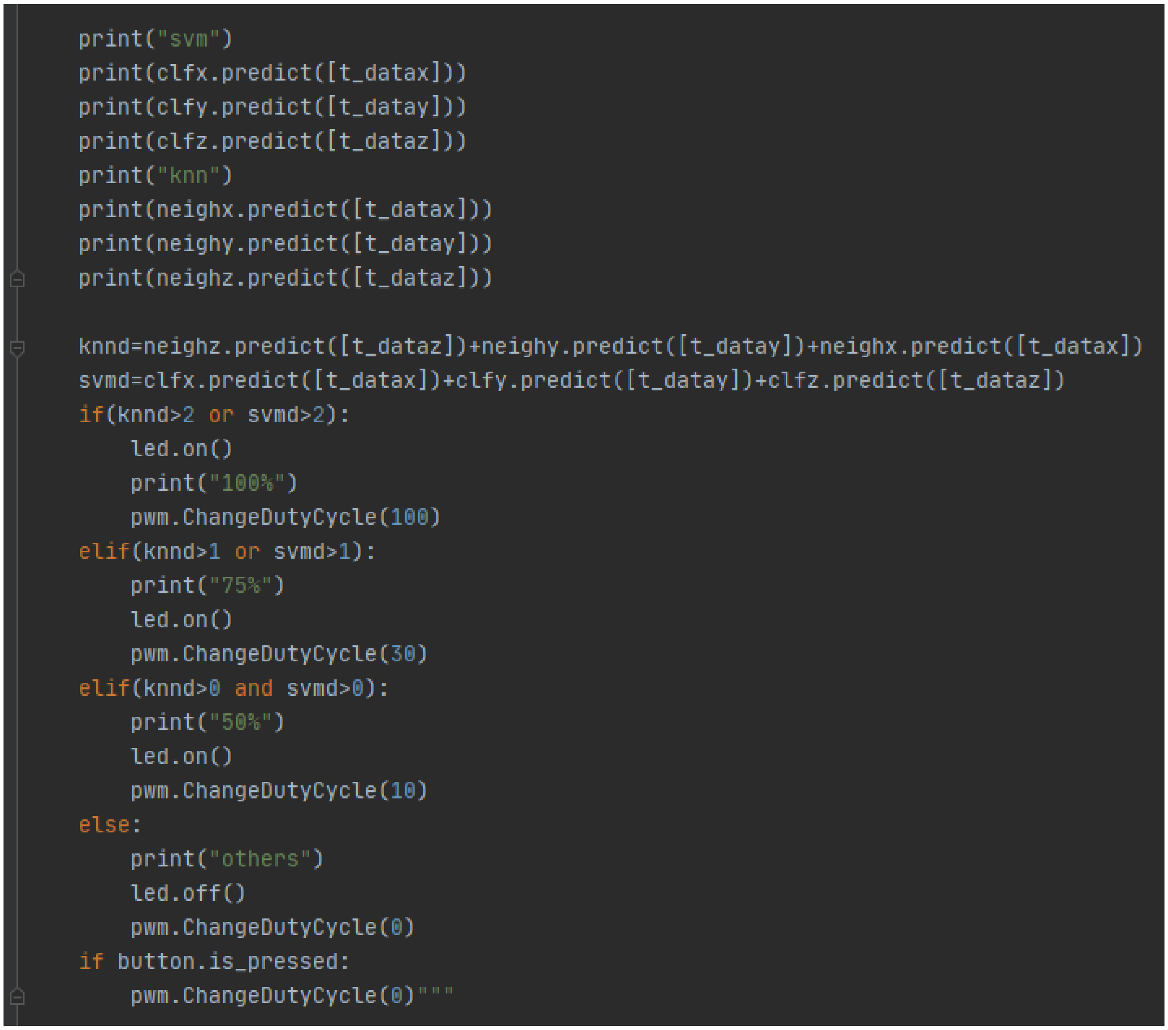


```
# %%
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
clfx = make_pipeline(StandardScaler(), SVC(gamma='auto'))
clfx.fit(abs(datax), value)
clfy = make_pipeline(StandardScaler(), SVC(gamma='auto'))
clfy.fit(abs(datay), value)
clfz = make_pipeline(StandardScaler(), SVC(gamma='auto'))
clfz.fit(abs(dataz), value)
#%%
from sklearn.neighbors import KNeighborsClassifier
neighx = KNeighborsClassifier(n_neighbors=5)
neighx.fit(abs(datax), value)
neighy = KNeighborsClassifier(n_neighbors=5)
neighy.fit(abs(datay), value)
neighz = KNeighborsClassifier(n_neighbors=5)
neighz.fit(abs(dataz), value)

# %%
```

Fig 4(i): Python code-2

Explanation: The machine learning functions are imported from the inbuilt libraries. These functions are used to train the model for each axis. The first block has functions for training the model using SVM algorithm and the second block has functions for training the model using KNN algorithm.

For predicting the output:



```
print("svm")
print(clfx.predict([t_datax]))
print(clfy.predict([t_datay]))
print(clfz.predict([t_dataz]))
print("knn")
print(neighx.predict([t_datax]))
print(neighy.predict([t_datay]))
print(neighz.predict([t_dataz]))

knnd=neighz.predict([t_dataz])+neighy.predict([t_datay])+neighx.predict([t_datax])
svmd=clfx.predict([t_datax])+clfy.predict([t_datay])+clfz.predict([t_dataz])
if(knnd>2 or svmd>2):
    led.on()
    print("100%")
    pwm.ChangeDutyCycle(100)
elif(knnd>1 or svmd>1):
    print("75%")
    led.on()
    pwm.ChangeDutyCycle(30)
elif(knnd>0 and svmd>0):
    print("50%")
    led.on()
    pwm.ChangeDutyCycle(10)
else:
    print("others")
    led.off()
    pwm.ChangeDutyCycle(0)
if button.is_pressed:
    pwm.ChangeDutyCycle(0)"""
```

Fig 4(j): Python code-3

Explanation: The predicted outputs of both the algorithms for all the sensors is combined together to generate a final output which tells if it was a seizure or not. The buzzer is turned on and the signal to send the alert notification to the caretaker is sent.

Node-Red flow for GUI:

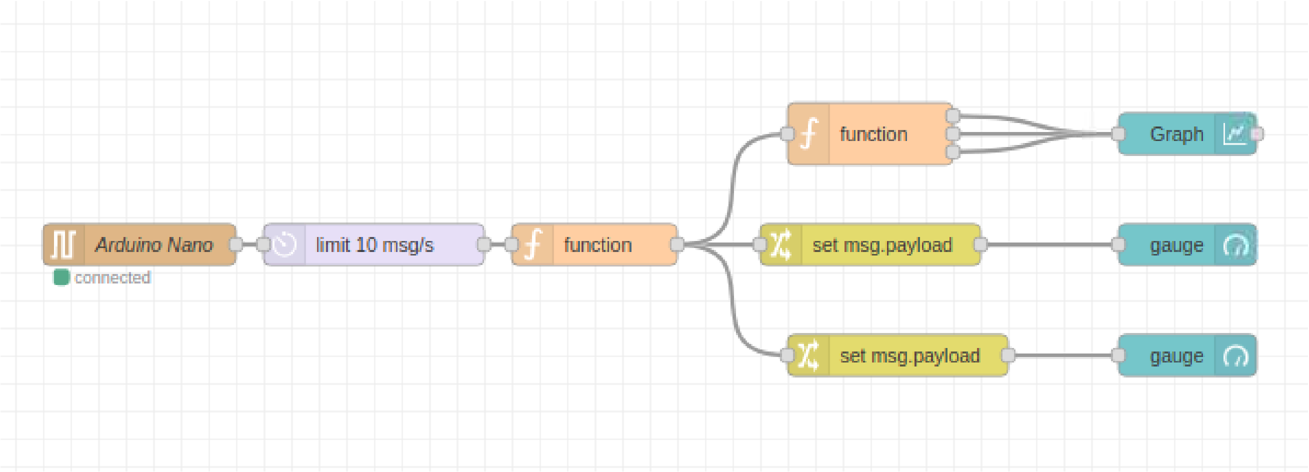


Fig 4(k): Node-Red flow-1

Explanation: The data received from the Arduino is filtered and the function blocks splits the combined data into individual sensor values. UI blocks such as graphs and gauges are used to display the data on the Node-Red dashboard.

Node-Red flow for email notification:

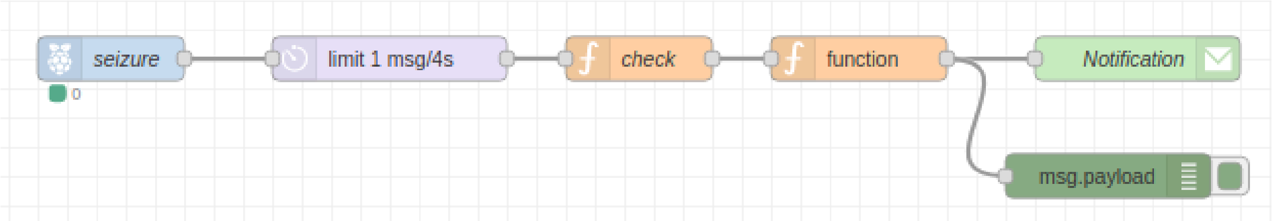


Fig 4(l): Node-Red flow-2

Explanation: On receiving the seizure detected signal, the notification block sends an email notification to the caretaker. The signal is limited to one message per 4 seconds so that the alert notification is not continuously sent.

GUI created using NoeRed:

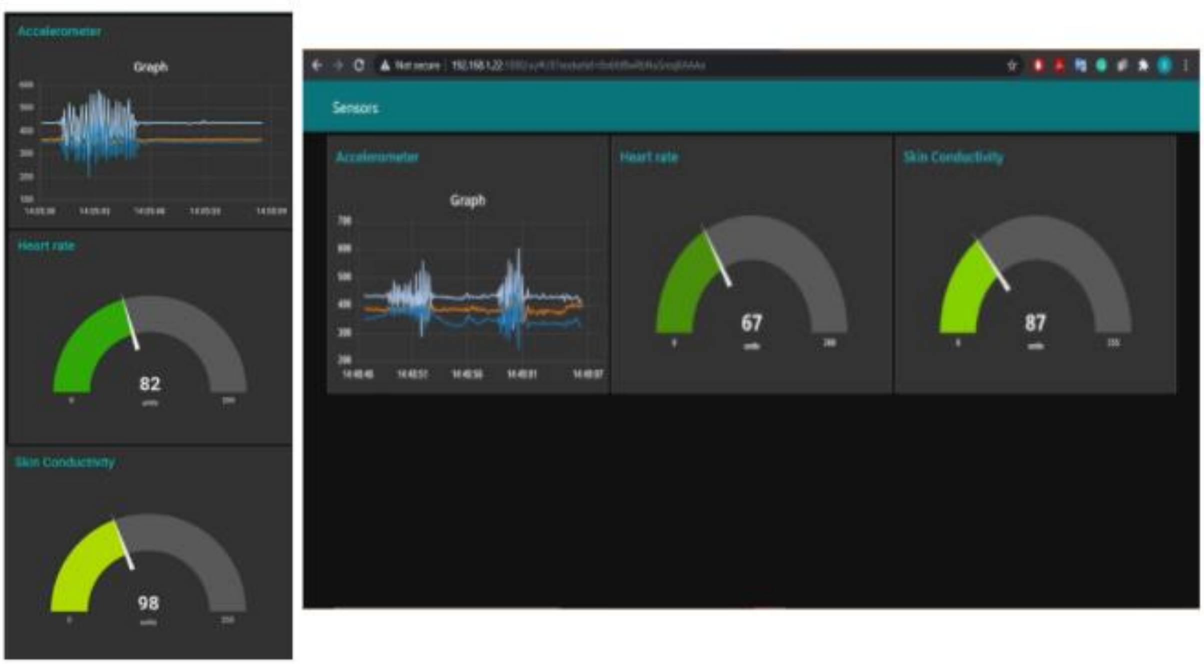


Fig 4(m): Mobile and Desktop versions of GUI

CHAPTER 05

HARDWARE AND SOFTWARE DESCRIPTION

HARDWARE DESCRIPTION

The following components have been used to build this circuit:

1. 3-Axis Accelerometer (ADXL 335)
2. Heart Rate and pulse-oximeter (MAX30102)
3. Raspberry Pi (MODEL 4B)
4. Arduino Nano
5. Buzzer
6. Skin Conductivity Meter
7. Power Supply

A detailed explanation about these elements is given for better understanding of the circuit.

1) 3-AXIS ACCELEROMETER

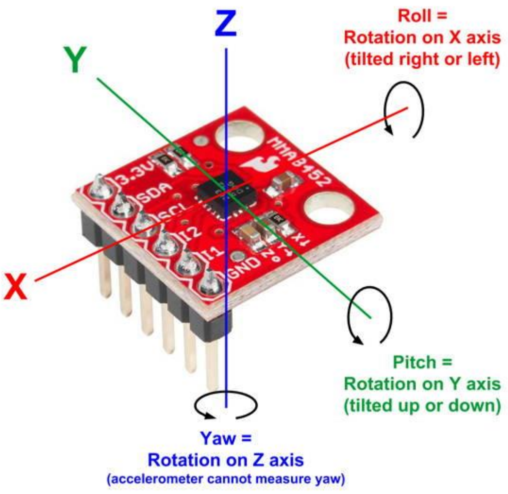


Fig 5(a): Accelerometer

An Accelerometer is a sensor that detects/measures the movement and identifies the orientation of the device it is mounted on. The triaxial accelerometer module used here is ADXL 335. This sensor can monitor both static and dynamic accelerations, although only dynamic acceleration (jerking motion) is observed in this case. The device is compact, power efficient and economical. The minimum fall range this sensor can record is 3g (positive or negative). The output produced is analog voltages which are proportional to the acceleration observed in the specific direction.

Working:

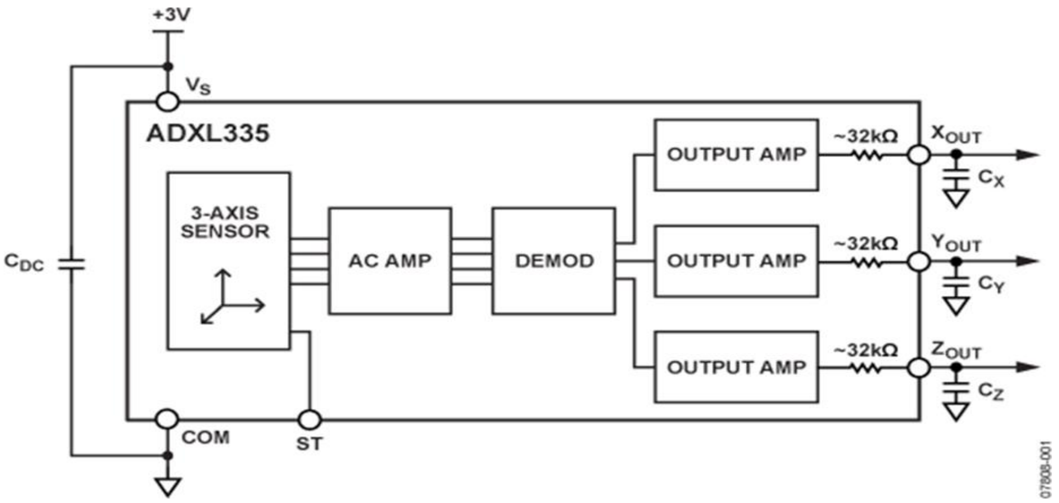


Fig 5(b):Internal block diagram of ADXL 335

The sensor houses 3 polysilicon springs which are suspended, these springs provide resistance against the acceleration forces in the respective directions. The deviation observed in the position of strings is measured using differential capacitors. Differential capacitors have fixed plates, the plates are in turn connected to moving masses. The plates of the capacitor are driven by out of phase square waves. The acceleration moves the mass attached to the capacitor which disturbs the stability, thus producing a sensor output which is proportional to the acceleration which caused the movement of the mass. The sensor output is further passed through a phase sensitive demodulator to compute the magnitude and direction(x, y or z). The demodulator is further amplified and the final output received from the module is passed through a 32kΩ resistor. The signal bandwidth can be further modified by adding additional capacitors. This filtering helps in improvement of output observed in terms of resolution.

2) HEART RATE AND PULSE-OXIMETER

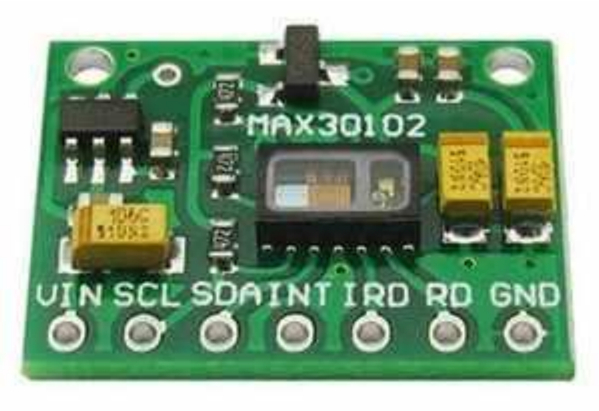


Fig 5(c): Heart rate and SPO2 sensor

A heart rate sensor is a device that tracks, monitors and records the heart rate of the person in contact with it. A pulse-oximeter indicates the oxygen levels of the user. Both the functions can be done using the module MAX30102. A MAX30102 is an integrated pulse oximeter and heart rate monitor biosensor module. The module consists of internal LEDs, photodetectors and optical elements.

In this approach, the heart rate sensor is used to detect the heart rate of the user at every instant during the seizure wherein the heart rate fluctuates and this is monitored by the device.

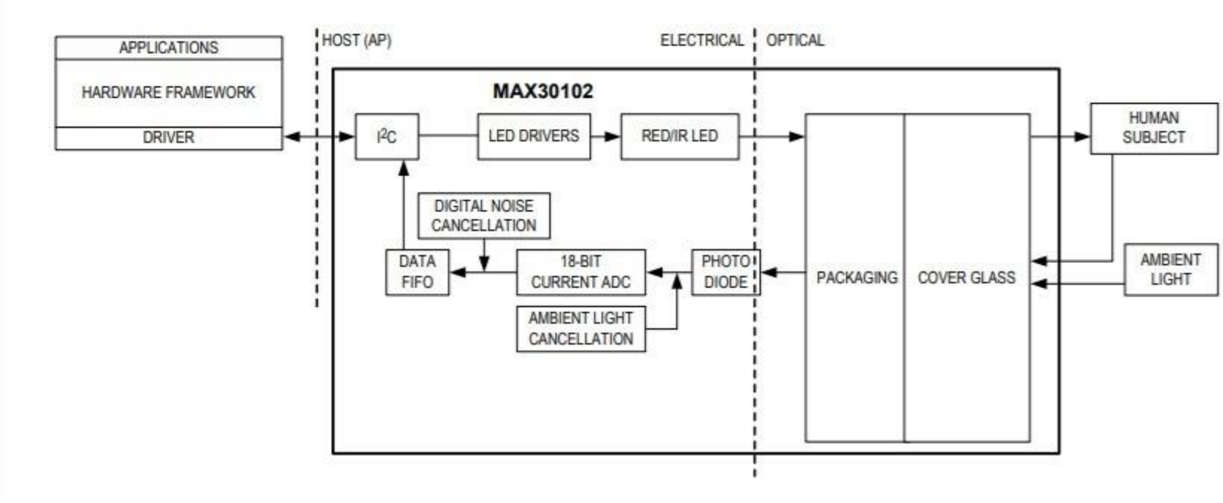


Fig 5(d): Internal block diagram of MAX30102

Working:

This sensor module outputs data in digital form which is stored in a 32-deep FIFO(first in first out) form within the IC. The FIFO present in the IC allows the interfacing of the module (MAX30102) to be interfaced with a microcontroller / microprocessor or a shared bus wherein the output is not being accessed instantly.

The module consists of a subsystem SPO_2 which helps in determining the oxygen levels of the user. This subsystem consists of 3 main parts:

1) Ambient light cancellation(ALC)

2) Continuous-time sigma-delta time filter

3) Proprietary discrete time filter

This module integrates RED and Infrared LED drivers to modulate the input(LED pulses) from oxymeter (SPO2) and heart rate measurements. The current at the LED is programmable with a range of 0-50 mA with suitable power supply.

3) SKIN CONDUCTIVITY METER

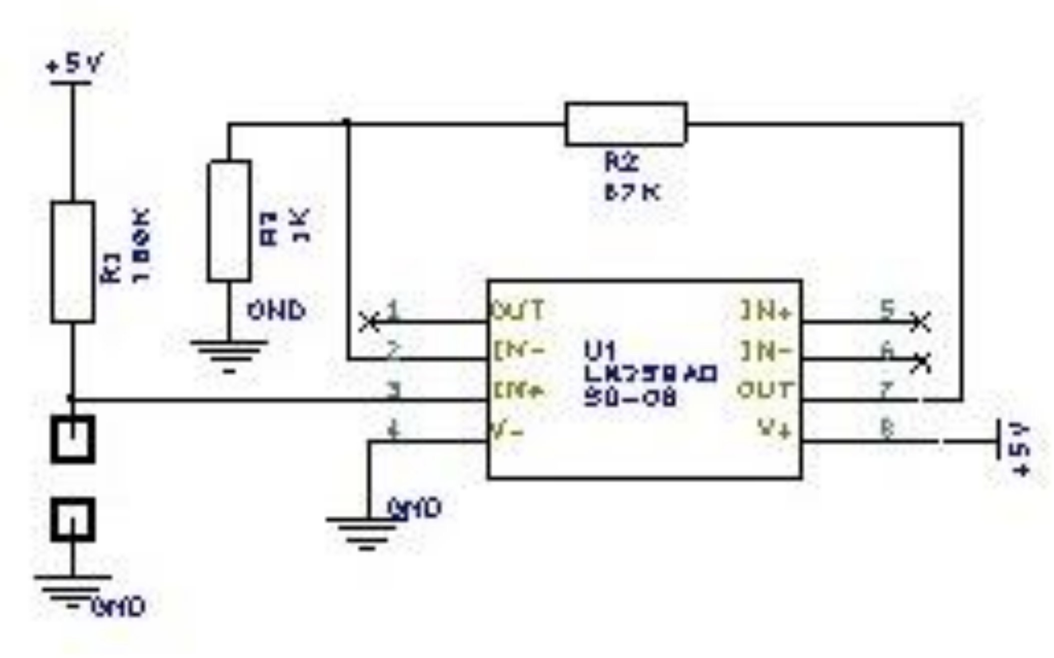


Fig 5(e):Skin Conductivity meter

A skin conductivity meter measures the variations in the resistance of the skin. The output of this meter is an analog voltage value which is further amplified. The input is given to the touch pads. The touch pads are nothing but a pair of metal contacts.

The voltage variations detected in this stage are amplified by the IC741, this IC is a non-inverting amplifier. The gain is maintained to be 2. This is achieved by varying the value of the feedback resistance Rf. Thus, the voltage leaving the IC will be two times the input.

In our project, skin conductance is also one of the important parameters. The skin conductivity meter built detects the rapid sweating that takes place during seizures which increases the skin conductivity.

4) ARDUINO NANO

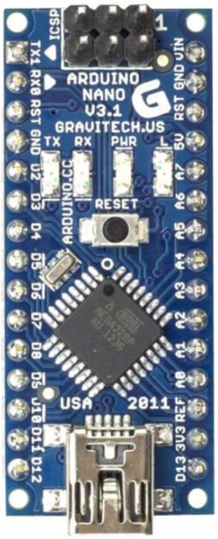


Fig 5(f): Arduino Nano

An arduino nano is a microcontroller developed by Arduino.cc and is based on ATMEGA328p. Basically a breadbroad embedded version surface mount that comes with an integrated USB. Nano has an operating voltage of 5V, contains digital pins,analog pins,power and reset pins. These inputs are interfaced with the sensors and the data is collected. This data collected from the sensors is transferred to the raspberry pi.

Arduino Nano Pin Description:

PIN NO	PIN NAME	PIN DESCRIPTION
1	**TX1**	**DIGITAL PIN 1 , ACTS AS TX PIN**
2	**RX0**	**DIGITAL PIN 0, ACTS AS RX PIN**
3	**RST**	**RESET PIN**
4	**GND**	**GROUND PIN**
5	**D02**	**DIGITAL PIN 2,INTERRUPT**
6	**D03**	**DIGITAL PIN 3,INTERRUPT,PWM**
7	**D04**	**DIGITAL PIN 4,12C:SDA**
8	**D05**	**DIGITAL PIN 5,PWM,12C:SCL**
9	**D06**	**DIGITAL PIN 6,PWM**

10	**D07**	**DIGITAL PIN 7**
11	**D08**	**DIGITAL PIN 8,**
12	**D09**	**DIGITAL PIN 9,PWM**
13	**D10**	**DIGITAL PIN 10,SPI SS,PWM**
14	**D11**	**DIGITAL PIN 11,SPI:MOSI,PWM**
15	**D12**	**DIGITAL PIN 12,SPI:MISO**
16	**D13**	**DIGITAL PIN 13,LED , SPI :SOK**
17	**3V3**	**+3.3 VOLTS**
18	**REF**	**ANALOG REFERENCE VOLTAGE**
19	**A00**	**ANALOG 00**
20	**A01**	**ANALOG 01**
21	**A02**	**ANALOG 02**
22	**A03**	**ANALOG 03**
23	**A04**	**ANALOG 04**
24	**A05**	**ANALOG 05**
25	**A06**	**ANALOG 06**
25	**A07**	**ANALOG 07**
27	**5V**	**+5.0 VOLTS**
28	**RST**	**RESET PIN**
29	**GND**	**GROUND**
30	**VIN**	**SUPPLY VOLTAGE VIN(7-12 VOLTS DC)**

TAB 5.1: NANO Pin Description

5) RASPBERRY PI

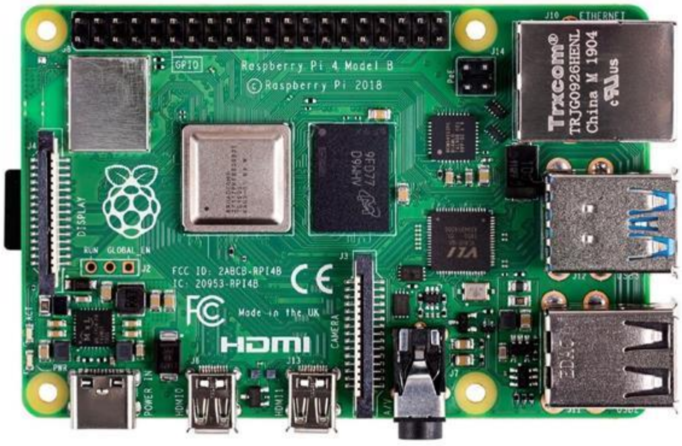


Fig 5(g): Raspberry PI top view

A single chip computer is known as Raspberry Pi. It was first developed by the raspberry pi foundation in association with Broadcom. The processor, memory and graphics card are soldered onto a single chip.

Pi runs on linux and provides GPIO pins which are general purpose input/output pins that enable us to connect it with electronic components.This board can be later interfaced with input-output peripherals such as monitor, keyboard and a mouse to make it work like a normal computer. This device is an economical option when compared with other similar devices. Raspberry pi comes in various variants. The model used here is the Raspberry Pi Model 4B.

Several features of raspberry pi 4b include:

1. Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
2. 2GB LPDDR4-3200 SDRAM
3. 2 X micro-HDMI ports
4. 2 USB 3.0 ports; 2 USB 2.0 ports

5. 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE, Gigabit Ethernet.

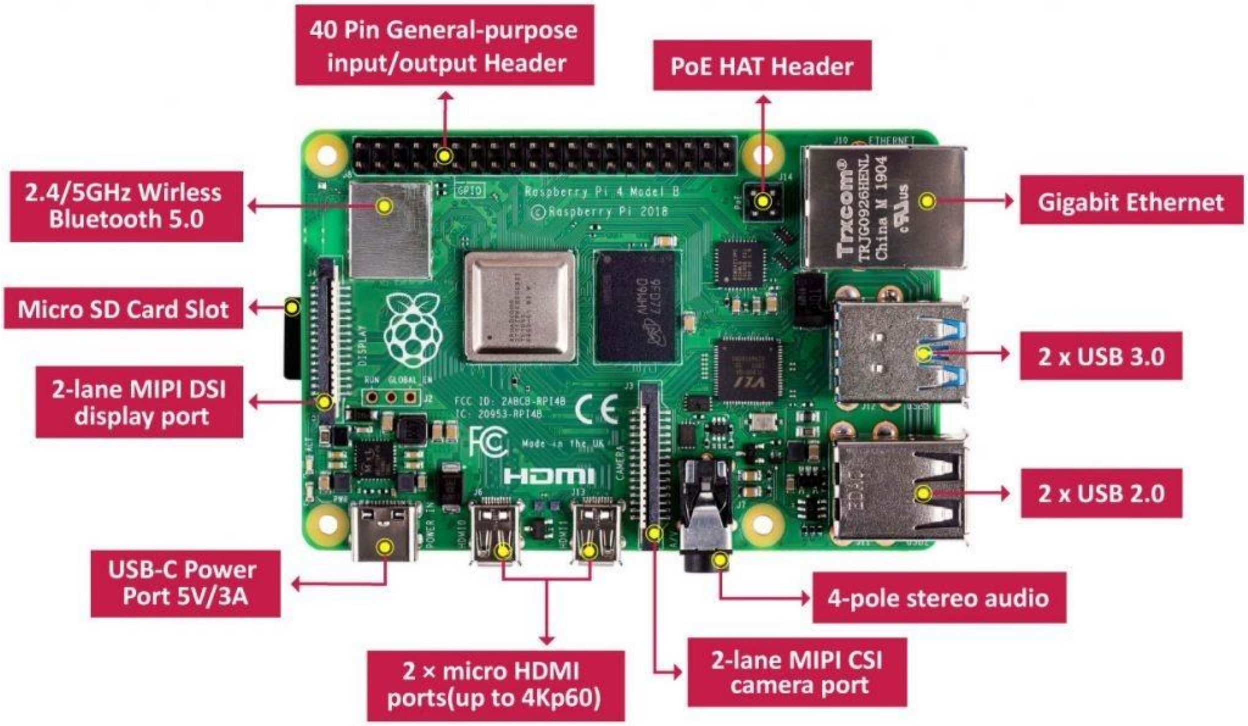
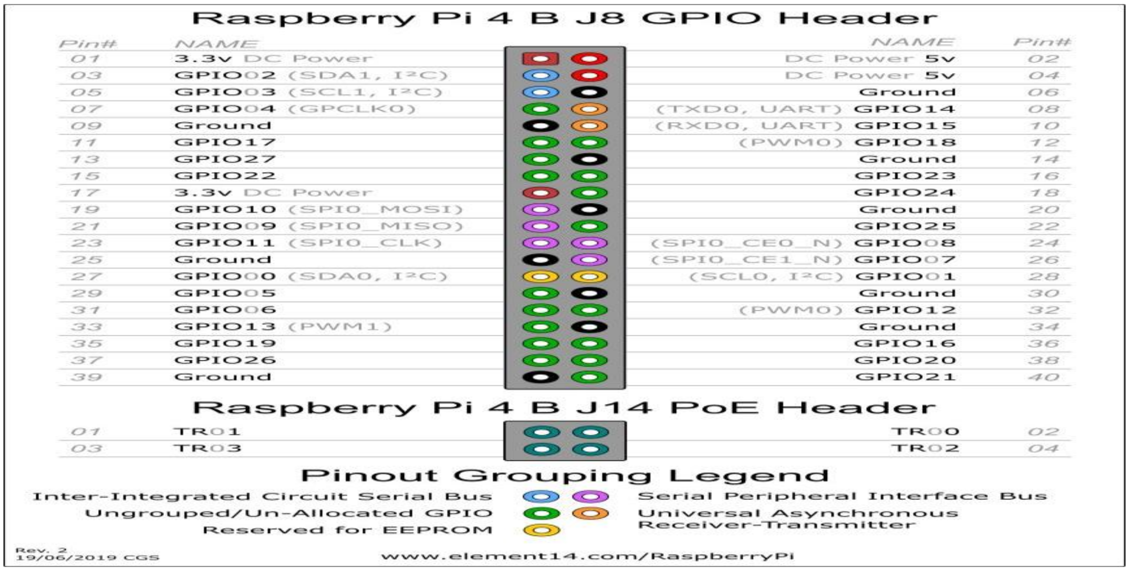

40 Pin General-purpose input/output Header
PoE HAT Header
2.4/5GHz Wirless Bluetooth 5.0
Gigabit Ethernet
Micro SD Card Slot
2 x USB 3.0
2-lane MIPI DSI display port
2 x USB 2.0
USB-C Power Port 5V/3A
4-pole stereo audio
2 × micro HDMI ports(up to 4Kp60)
2-lane MIPI CSI camera port

Fig 5(h): Raspberry PI Description

.THE GPIO(GENERAL PURPOSE INPUT/OUTPUT PINS):



Raspberry Pi 4 B J8 GPIO Header

Pin#	NAME			NAME	Pin#
01	3.3v DC Power	🔴	🔴	DC Power 5v	02
03	GPIO02 (SDA1, I²C)	🔵	🔴	DC Power 5v	04
05	GPIO03 (SCL1, I²C)	🔵	⚫	Ground	06
07	GPIO04 (GPCLK0)	🟢	🟠	(TXD0, UART) GPIO14	08
09	Ground	⚫	🟠	(RXD0, UART) GPIO15	10
11	GPIO17	🟢	🟢	(PWM0) GPIO18	12
13	GPIO27	🟢	⚫	Ground	14
15	GPIO22	🟢	🟢	GPIO23	16
17	3.3v DC Power	🔴	🟢	GPIO24	18
19	GPIO10 (SPI0_MOSI)	🟣	⚫	Ground	20
21	GPIO09 (SPI0_MISO)	🟣	🟢	GPIO25	22
23	GPIO11 (SPI0_CLK)	🟣	🟣	(SPI0_CE0_N) GPIO08	24
25	Ground	⚫	🟣	(SPI0_CE1_N) GPIO07	26
27	GPIO00 (SDA0, I²C)	🟡	🟡	(SCL0, I²C) GPIO01	28
29	GPIO05	🟢	⚫	Ground	30
31	GPIO06	🟢	🟢	(PWM0) GPIO12	32
33	GPIO13 (PWM1)	🟢	⚫	Ground	34
35	GPIO19	🟢	🟢	GPIO16	36
37	GPIO26	🟢	🟢	GPIO20	38
39	Ground	⚫	🟢	GPIO21	40

Raspberry Pi 4 B J14 PoE Header

01	TR01	●	●	TR00	02
03	TR03	●	●	TR02	04

Pinout Grouping Legend

Inter-Integrated Circuit Serial Bus	🔵	🟣	Serial Peripheral Interface Bus
Ungrouped/Un-Allocated GPIO	🟢	🟠	Universal Asynchronous Receiver-Transmitter
Reserved for EEPROM	🟡		

Rev. 2
19/06/2019 CGS

www.element14.com/RaspberryPi

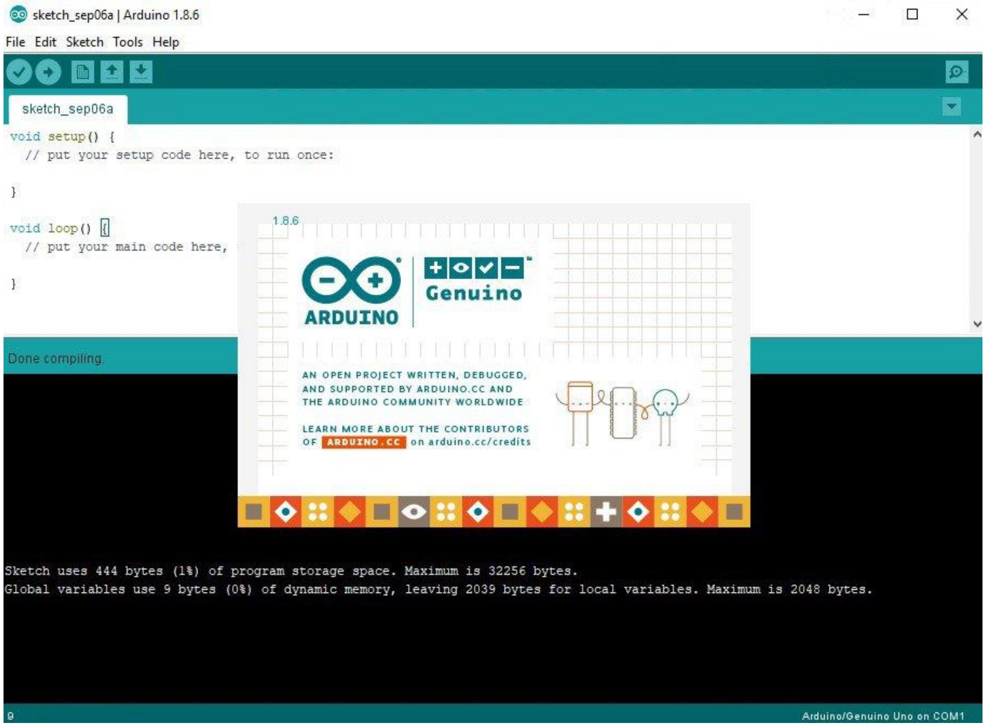
Fig 5(i): GPIO DESCRIPTION

SOFTWARE DESCRIPTION

The basic software requirements for this project are:

- Arduino IDE
- Python 3
- Node-RED

1) ARDUINO IDE



sketch_sep06a | Arduino 1.8.6
File Edit Sketch Tools Help
sketch_sep06a
1.8.6
ARDUINO
Genuino
Done compiling.
9
Arduino/Genuino Uno on COM1

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here,

}
```

AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

```
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

Fig 5(j): Arduino IDE

This Arduino (integrated development environment) , also referred to as the Arduino code package is an application program that provides a prototyping platform , thereby allowing users to develop interactive electronics objects. The Arduino IDE software that runs on Windows, Linux, Mac operating system makes it easier to write codes and execute and upload them to the boards. These Arduino boards are basically those development boards that enable makersto build devices, design and interact with oy devices.

This software contains the applications program for composing codes, a book support, a toolbar having the provision for basic capacities,a message region and also a progression of menus.

2)PYTHON 3

Fig 5(k): Python IDE

Python as we know is a general purpose high level programming language. Python is an interactive and object-oriented programming language which supports the technique involving programming which encapsulates codes within objects.

Python programming consists of high level data types that are dynamic and also support dynamic type checking. Most times it is used a s a scripting language , can be compiled for building large scale applications

Integrated development environment(IDEs) provide the essentials needed that are needed for the particular software's development. Incase of Python there are various IDEs available such as +PyDev, Spyder, IDLE(at beginner-level), Wing , VS code, Eclipse,Thonny, PyCharm. The most commonly used IDE is PyCharm.

This is a very suitable and recommended coding language for beginner-level as it highly readable and supports the creation and development of variety of applications. Python can be very easily integrated with other languages like C,COM,C++,Java and Cobra.

3)Node-RED

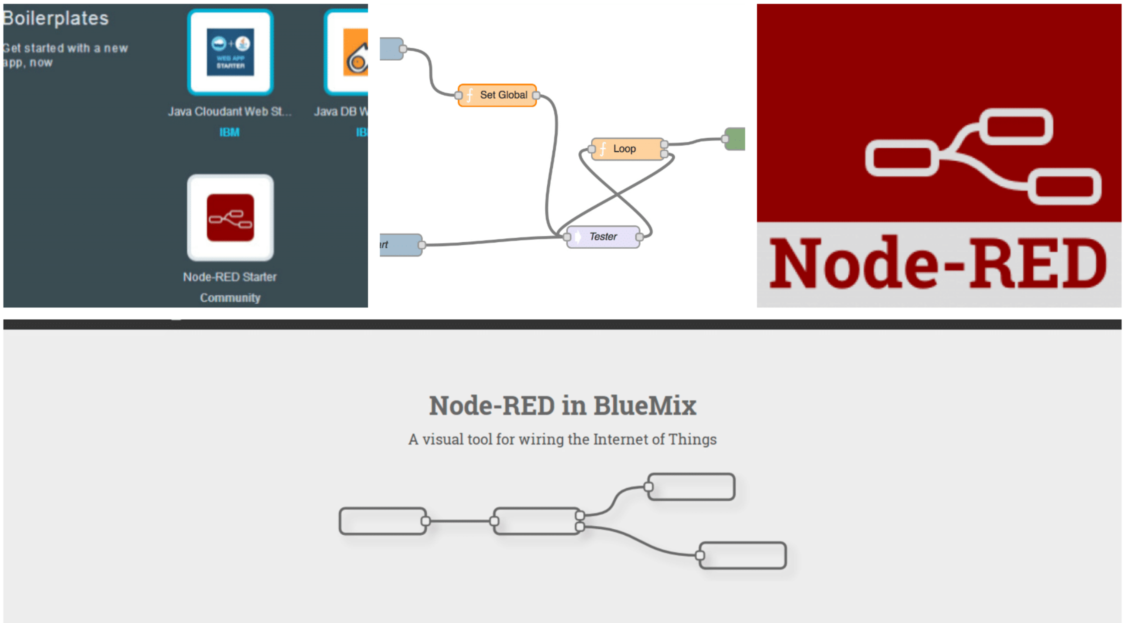


Fig 5(l): Node-RED

Node-RED is an open source flow based programming software that wires together hardware devices, online-services, APIs,etc. It is basically a flow editor that connects together the flows what is known as the nodes which are available in the palette provided by node red.

NodeRED gives us the option of flow editing which is browser based, wherein the flows created with the help of the required nodes can be deployed with the run time in just a click. NodeREd generally uses JavaScript functions that can be created using the editor available, but there is a built-in library using which we can save the created functions,flows,templates, etc and can also write the node functions in other programming languages.

There are over 225000 modules in Node' s repository which makes it easy to extend the palette node range and also include new functionalities. It is also an IOT and Dashboard tool.

CHAPTER 06

RESULT AND DISCUSSION

- The primary outcome expected is that the device will serve the purpose of saving a patient during the time of seizure-events.

- Thereby avoiding cases of negligence or delay in medical aid reaching the patient at the right time.

- With timely medical help arriving, the risk involved for the patient is reduced, as the seizures might not prove to be fatal

- It will also be useful for people who just want this device as a band that monitors their health.

- Another expected outcome is that the patient's daily routine should not be disturbed.

CHAPTER 07

CONCLUSION AND FUTURE SCOPE

- Although there are similar devices present in the market, their functionality is restricted to detecting just a few cases of seizures. Our model will solve these drawbacks and have additional features.

- Making sure a medical response arrives at the appropriate time is also taken care of.

- Traditionally, the seizure detection methods are complex involving brain activity monitoring through ECGs.

- Reducing this complexity we are detecting the seizure by monitoring their body parameters. Our model will be cost-effective and easily affordable unlike the existing products which are expensive. The estimated price is 1500 rupees.

FUTURE SCOPE:

- In-order to avoid cases of negligence, nearby hospitals can be alerted by sending the patient’s current GPS location.
- Proper sleep monitoring.
- When it is a product, it will be made battery powered and compact.

REFERENCES

[1] Jussa Klapuri,"**Epileptic Seizure Detection using a Wrist-worn Triaxial Accelerometer**", 2013

[2] Sora An,Chaewon Kang,"**AI-based approach to generate a computational machine performing specific task-detecting a seizure**",2020

[3] Sandor Beniczky, Tilman Polster,Helle Hjalgrim," **Detection of generalized tonic–clonic seizures by a wireless wrist accelerometer: A prospective, multicenter study**",2013

[4] ADXL 335 datasheet by Analog devices

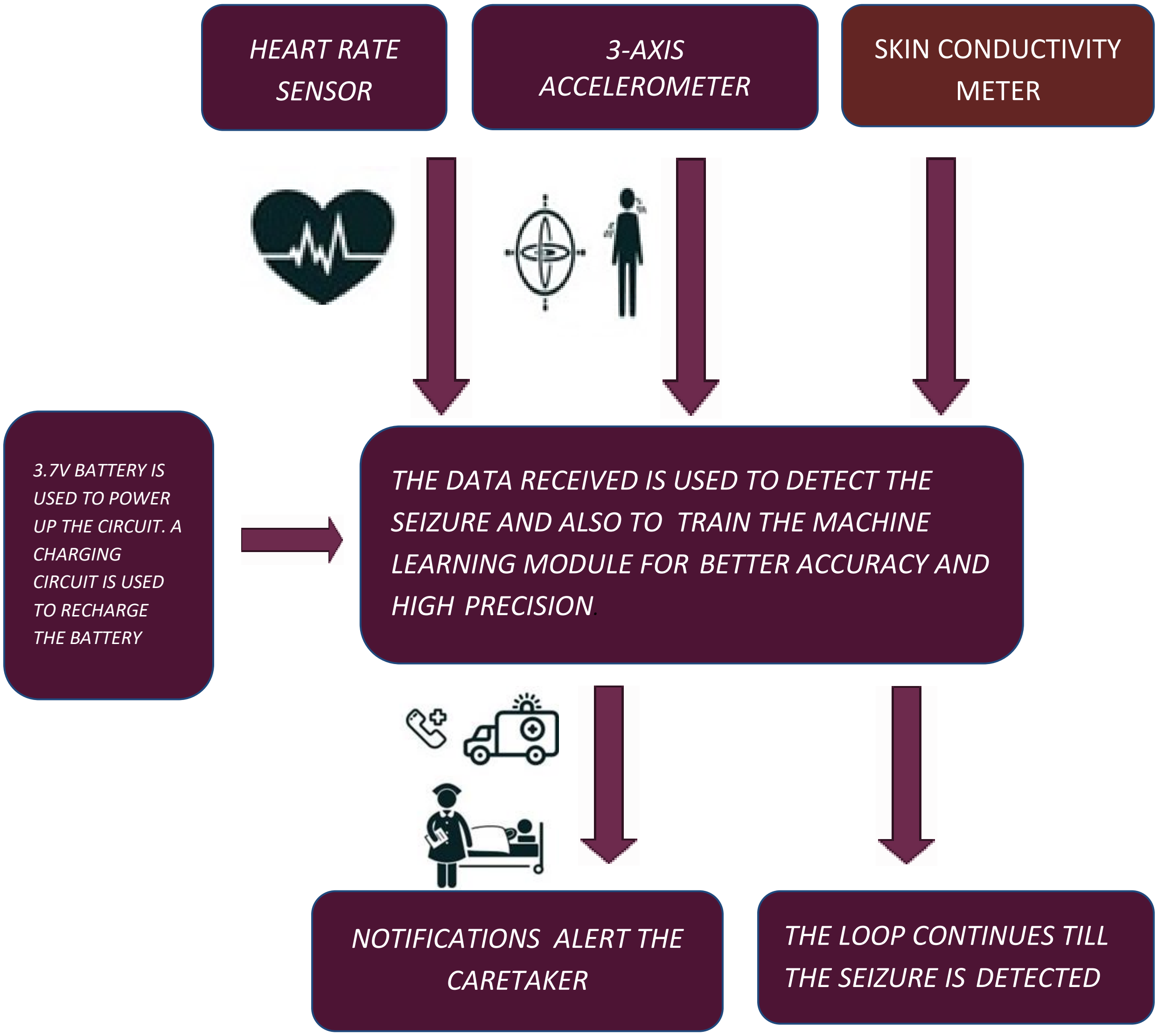
[5] MAX30102 datasheet by Maxim Integrated

[6] https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf

[7] https://www.raspberrypi.org/documentation/usage/gpio/

APPENDIX

AN OVERVIEW



HEART RATE SENSOR

3-AXIS ACCELEROMETER

SKIN CONDUCTIVITY METER

3.7V BATTERY IS USED TO POWER UP THE CIRCUIT. A CHARGING CIRCUIT IS USED TO RECHARGE THE BATTERY

THE DATA RECEIVED IS USED TO DETECT THE SEIZURE AND ALSO TO TRAIN THE MACHINE LEARNING MODULE FOR BETTER ACCURACY AND HIGH PRECISION.

NOTIFICATIONS ALERT THE CARETAKER

THE LOOP CONTINUES TILL THE SEIZURE IS DETECTED

Fig A: An overview of the working