Integration of Galvanic skin response, skin temperature, Pulse volume changes and Heart Rate variability in Pain monitoring

A Thesis

Submitted By

Himanshu Kumar

For the award of degree of

MASTER OF TECHNOLOGY, CLINICAL ENGINEERING

Jointly offered by

Indian Institute of Technology, Madras

Christian Medical College, Vellore

Sree Chitra Thirunal Institute of Medical Sciences and Technology, Trivandrum

June 2021

CERTIFICATE

This is to certify that the thesis titled ‘

Integration of Galvanic skin response, skin temperature, Pulse volume changes and Heart Rate variability in Pain monitoring

’ being submitted by **Himanshu Kumar** to SCTIMST Trivandrum, for the award of degree of **Master of Technology in Clinical Engineering** jointly offered by IIT Madras, CMC Vellore and SCTIMST Trivandrum, is a bonafide record of research work done by him under our supervision. The contents of this thesis in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

The research had been carried out at Sree Chitra Institute of Medical Sciences and Technology, Trivandrum.

Dr. Manikandan Sethuraman

Guide

Professor and Incharge

Division of Neuroanaesthesia and Neurocritical Care,

Department of Anesthesiology,

SCTIMST, Trivandrum

Mr. Manoj G. S.

Guide

Engineer ‘C’

Department of Clinical Engineering,

SCTIMST, Trivandrum

ABSTRACT

This project concerns with the development of a python-based application on Raspberry Pi that can help in quantifying pain to an extent. Pain in general can be described as a feeling of discomfort felt by an individual. Several words can be used to describe pain but they do not convey as to what measures it can be put up to easily quantify one's pain from others. Words like terrible and some are probably the most useless terms to quantify pain. So, it is not easy to quantify pain using descriptive terms. Here various human body parameters have been measured and displayed on a graph of appropriate scale so as to differentiate the pain of an individual. The parameters used include GSR (Galvanic Skin Response) values, Body Temperature, Pulse Volume variations and changes in ECG (Electrocardiography) variations.

CONTENTS

1 INTRODUCTION

If someone is looking for a small-scale and cheaper way to do long-term testing, then this can be tackled with this very project. Here the need of laptops is eliminated by using single board computer running Linux.

One of famous foundation responsible for such single board computers is the Raspberry Pi Series that is a culmination of both, high performance and low price. They easily replace larger and powerful computers in certain tasks. This project makes an avid use of the Raspberry Pi hardware. A Sensor integrated device was developed controlled by Arduino and Raspberry Pi controller.

Data storage has been done in CSV files. The visualization of the sensor measurements has been through a GUI (Graphical User Interface) that is capable of displaying graphs over certain intervals of time. The pySide2 framework has been used to create a splash screen that leads to the main window where the graphs can be viewed.

There is provision to save images of graphs formed and view CSV file that stores the sensor data as well.

2 BACKGROUND

Hardware for this was already developed by the Hospital wing at SCTIMST which is the development of sensory systems to measure human body parameters. Various Sensors that were developed are Development of Galvanic Skin Response (GSR), Development of Body Temperature, Development of Pulse Monitor, Development of ECG Monitor.

Programming and Software Development was also done but language used was java compared to the one that is being used in this project which is python. Arduino microcontroller was programmed by C-programming language. Signal analysis and user interface developed through java language.

Graphical User Interface was developed. A basic Java software was already developed for the project which is currently capable of displaying real time GSR, Body temperature, pulse oximeter output and ECG, where the measured values have been plotted against time.

Before going Further, an explanation of various body parameters is necessary.

2.1 GSR (Galvanic Skin Response)

The Galvanic Skin Response is the change in the activity of sweat gland on the body that indicates the emotional state intensity of an individual. It is term that comes under the term EDA (Electrodermal Activity).

Both negative stimuli (sadness or threat) and positive stimuli (happiness or joy) can bring about a change (increase) in one's skin conductance. So, we can safely say that GSR represents the intensity of the emotion felt rather than the type of it.

Skin conductance is usually captured from hand or foot areas, generally fingers using electrodes that are easy to apply on the skin. Most of the electrodes have an Ag/AgCl contact point. These are cheap, robust, safe to be in contact with and can transmit the signal quite accurately.

A GSR sensory system was already developed that senses values and continuously send them to the micro-controller.

2.2 Body Temperature Sensor

An adult typically has a body temperature range from 97 F to 99 F whereas babies and children have a little higher range: 97.9 F to 100.4 F. Temperature of a human body doesn't remain same all day and vary all through the lifetime. Women tend to have body temperature that is higher than that of men.

For this project a System was successfully developed for the real time monitoring of body temperature.

2.3 Pulse Volume changes in SpO2

One of the non-invasive methods for monitoring an individual's oxygen saturation is pulse oximetry.

The most well-known methodology is transmissive heartbeat oximetry. In this methodology, a sensor gadget is put on a slender piece of the patient's body, generally a fingertip or ear cartilage, or a newborn child's foot. Fingertips and ear cartilage have higher blood stream rates than different tissues, which works with heat move.

The gadget passes two frequencies of light through the body part to a photodetector. It estimates the changing absorbance at every one of the frequencies, permitting it to decide the absorbances because of the beating blood vessel blood alone, barring venous blood, skin, bone, muscle, fat, and (as a rule) nail clean.

So, for this project, the pulse was continuously monitored using Light based Pulse sensor and the data was sent to controller.

2.4 ECG (Electrocardiograph) Monitor

An electrocardiogram records the electrical signs in your heart. It's a typical and effortless test used to rapidly identify heart issues and screen your heart's wellbeing.

An electrocardiogram is an effortless, noninvasive approach to help analyze numerous basic heart issues in individuals, all things considered. Your primary care physician may utilize an electrocardiogram to decide or distinguish:

- Unusual heart musicality (arrhythmias).
- Whenever obstructed or limited courses in your heart (coronary supply route sickness) are causing chest torment or a cardiovascular failure.
- Regardless of whether you have had a past respiratory failure.
- How well certain coronary illness medicines, like a pacemaker, are working.

Here, three lead ECG picking device was developed using AD8232 instrumentation amplifier where the electrodes send the electrical activity to the controller.

3 PURPOSE, AIM and MOTIVATION

To better allow the medical workers to visualize the pain intensity of the patient, this project main purpose is to make data visualization easy (touch and see). One click is all it should take rather than navigating an array of buttons to figure out what is what.

There are device makers who likes to give their own product according to what they think is useful regardless of whether eliminating every last bit is conceivable. This sometimes can also meddle with framework activity of the system. The idea is to make any future upgradation as smooth sailing as possible without hindering the current operation. Hence the use of python that allows for such an activity to take place easily.

The general query that this project intends to answer is whether the Raspberry Pi single board PC can be utilized to store and view various sorts of information. The task is significant on the grounds that it would bring about a broadly useful answer for the issue of viewing and storing information.

A wide range of arrangements exist for the said purpose that utilizes traditional PCs, however there is an absence of a universally useful arrangement utilizing a modest, small, and convenient framework like the Raspberry Pi.

3.1 Aim

Aim of the project is to develop a software solution for the Raspberry Pi that will make the data interpretation of human body parameters easier and can be done so in cheaper and more accessible way compared to using traditional computers.

An important part of this project is to create a code base that is properly standardized, comments are placed wherever necessary (usually the ratio of lines od code to line of comment that is maintained is 5:1), and correct documentation is done for future works. The result expected out of this is that any individual would be able to go through the application without any hiccup and any individual with enough knowledge of the software platforms should be easily able to understand the code and be in full capacity to modify it when needed.

3.2 Delimitations

The most important constraint in development of this project is that, it has only been developed for the Raspberry Pi 3 model, no testing has been done for models higher than this. Older model might show performance issue as they tend to have lower specifications that the currently being used.

One of the constraints is how fast the data reading is required by the sensor. As the sensor reads the data quite frequently (every 1/10th of a second). So, one need to decide what time intervals are needed to view the data.

Also, only last run data of the sensor is stored in the CSV file that captures the sensor output according the header values (GSR, Temperature, Pulse and ECG data). So, it becomes increasingly difficult to store the previous data, but the graph images can be stored before exiting the run as many times as possible.

4 EXISTING WORK

There is already a code base existing in java that captures the human body parameters namely GSR, Body temperature, Pulse Oximeter Output and ECG, where the measured values are plotted with respect to time. But the display is haphazard and any further changes to the code base is difficult.

Another similar project exists where temperature and pressure logger are built using the Raspberry Pi module. Here the temperature and pressure are measured by the Raspberry Pi and the collected data is sent to a third-party server that hosts the data and also does the visualization. The difference between the said project and this project is that it removes the need for any third-party server to store and visualize the collected sensor outputs.

5 METHOD

This section explains the various choices that led to the specifications being used in this project. In particular, the choice of programming language, and platform used to build the application is further discussed.

5.1 Application Language

Python is preferred programming language of choice for this project. First of all, no urgent Primary Memory (RAM) is required for this project as Pi already has 1 gigabyte of memory available, hence there is no need to choose a language that is memory efficient.

Productivity level is one of the important aspects for choosing this language. Syntax used for python are similar to C programming language and someone familiar to C, it is easier to learn it. Even for beginners it is simple compared to other languages like java or PERL. This helps anyone wanting to familiarize with the source code of the application. Python being a Cross-platform language, it allows the code to be run on other systems quite easily if needed.

One of the drawbacks of python is that it is an interpreted language rather than compiled languages like C, hence more hardware resources are needed to run the application programs.

5.2 Language Libraries

Python inherently provides various libraries that are helpful in creating in a working program, but in cases where there is need to build full-fledged GUI (Graphical User Interface), various other libraries are needed. Here the libraries used are PySide2, Matplotlib, Pandas and pyqtgraph to create and visualize a suitable user interface with minimal interventions.

5 SYSTEM ARCHITECTURE

The whole system architecture consists of a single Raspberry Pi that has sensors connected to it that provide the output on the monitor attached to it.

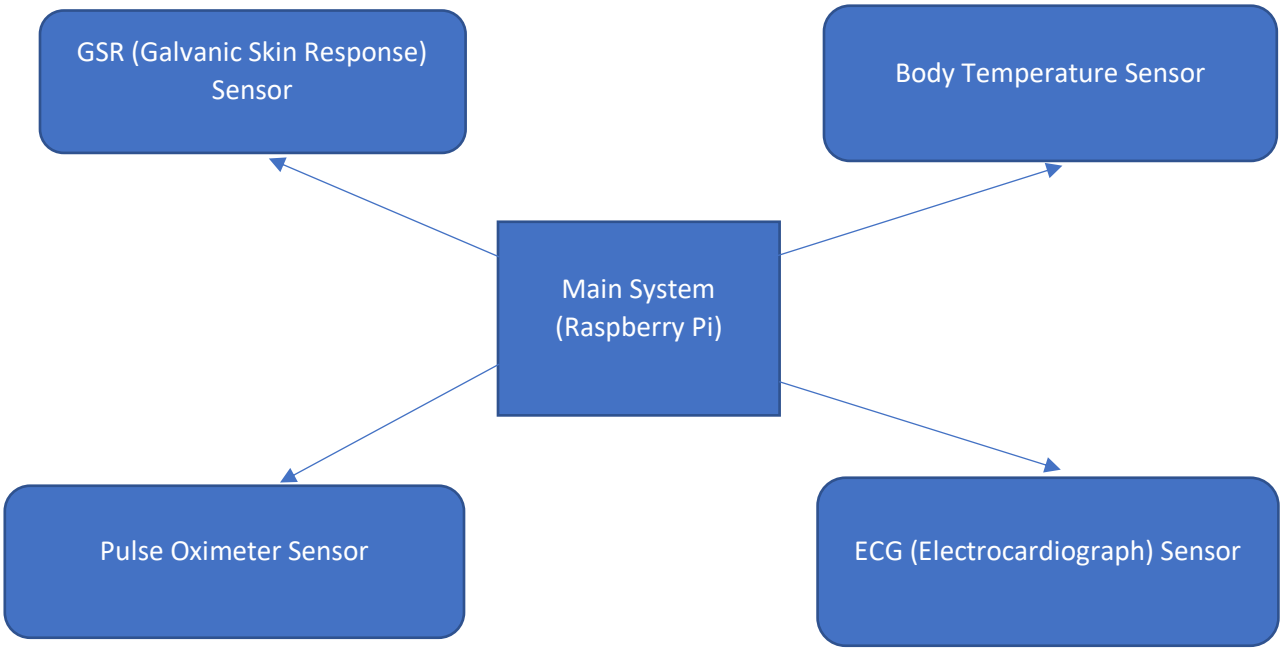

Figure1: Overall Structure of the System

5.1 Main System

This consists of a Raspberry Pi and an Arduino coupled together with a touchscreen that is being used to interact with the system GUI. The data is received through the Raspberry Pi USB-port which is then stored as CSV file and as more data is received, it is appended to that file for as long as the application doesn't exit.

Each system is linked with each other so when the screen is touched the input received by the hardware is what tells the system to behave in a certain manner. All these interactions combined gives the desired result.

The structure of the main system can be viewed in Figure2.

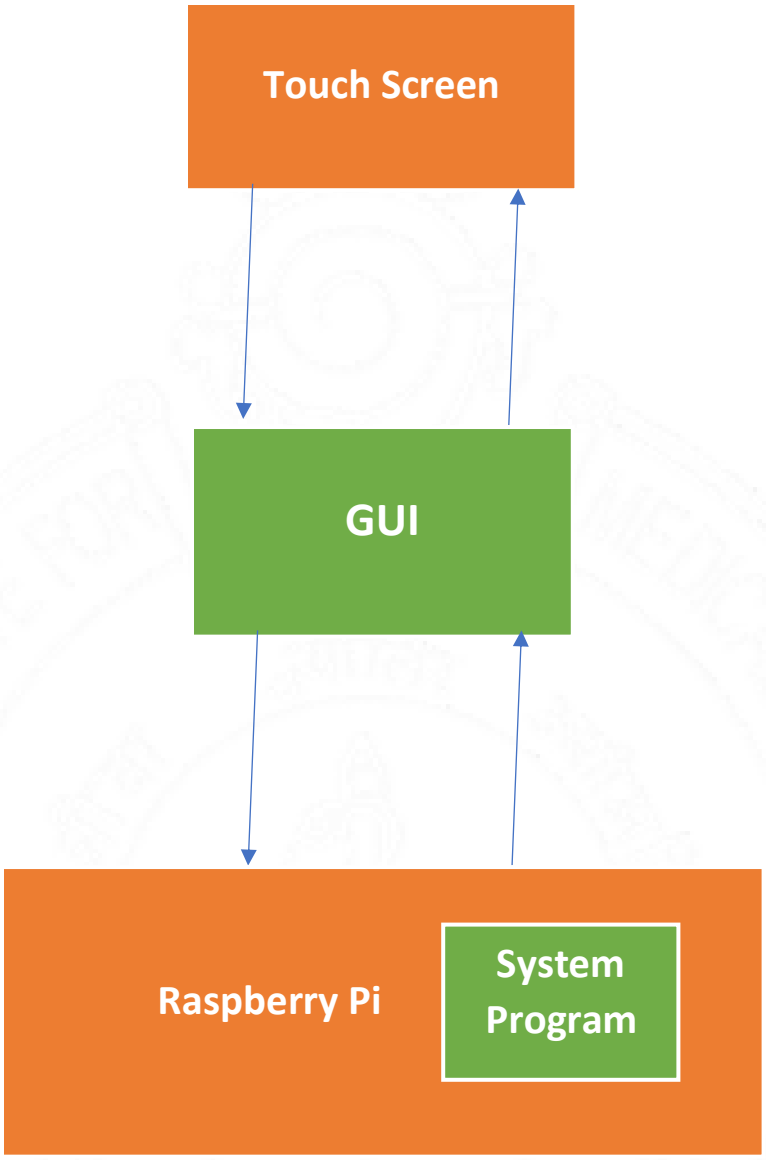

Figure2: Showing Structure of the System. The orange marked boxes are pre-existing hardware used in the project and green marked boxes are the ones being developed.

6 SYSTEM DESCRIPTION

The system is made of various parts that interact with each other to give a desired result. In this section, for each functionality of the front end (GUI) discussed here a view of how back end works for the said GUI will be discussed.

6.1 Splash Screen

Running the application at the beginning presents the user with a screen that allows the user to understand what the application is about while all the required data is being loaded.

When the application is run the first screen the user encounters is as shown in figure3 to figure5 below.

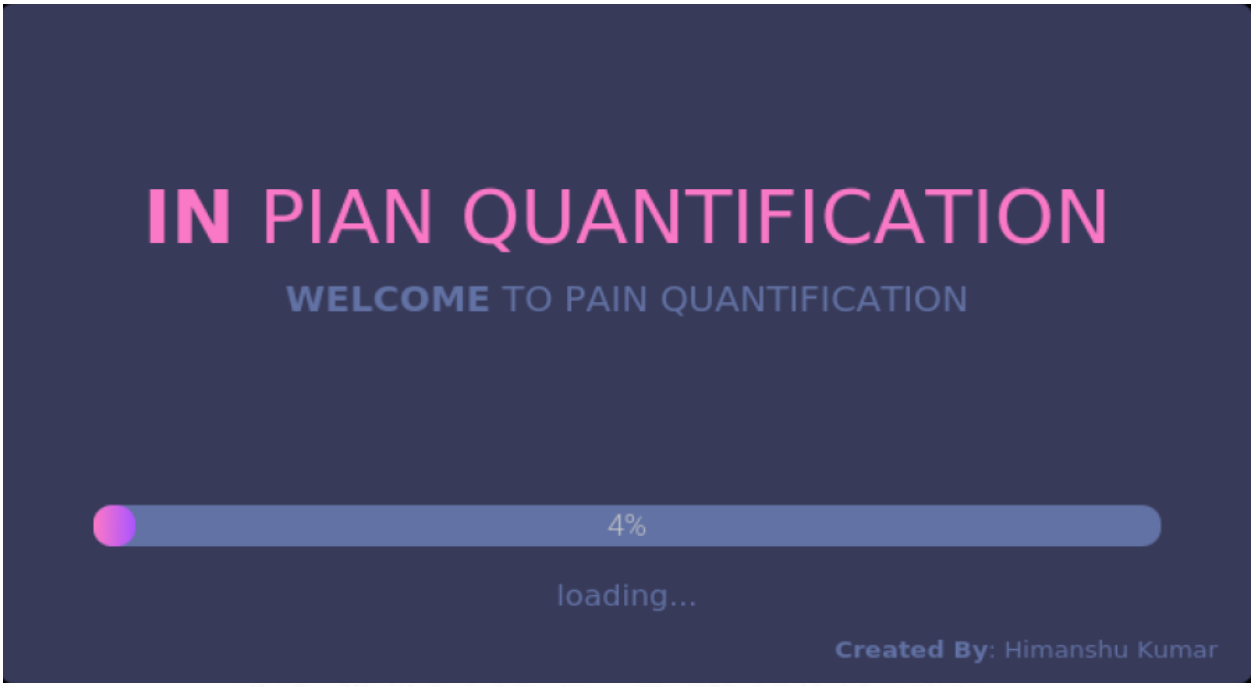

Figure3: Welcome Screen when the application is run

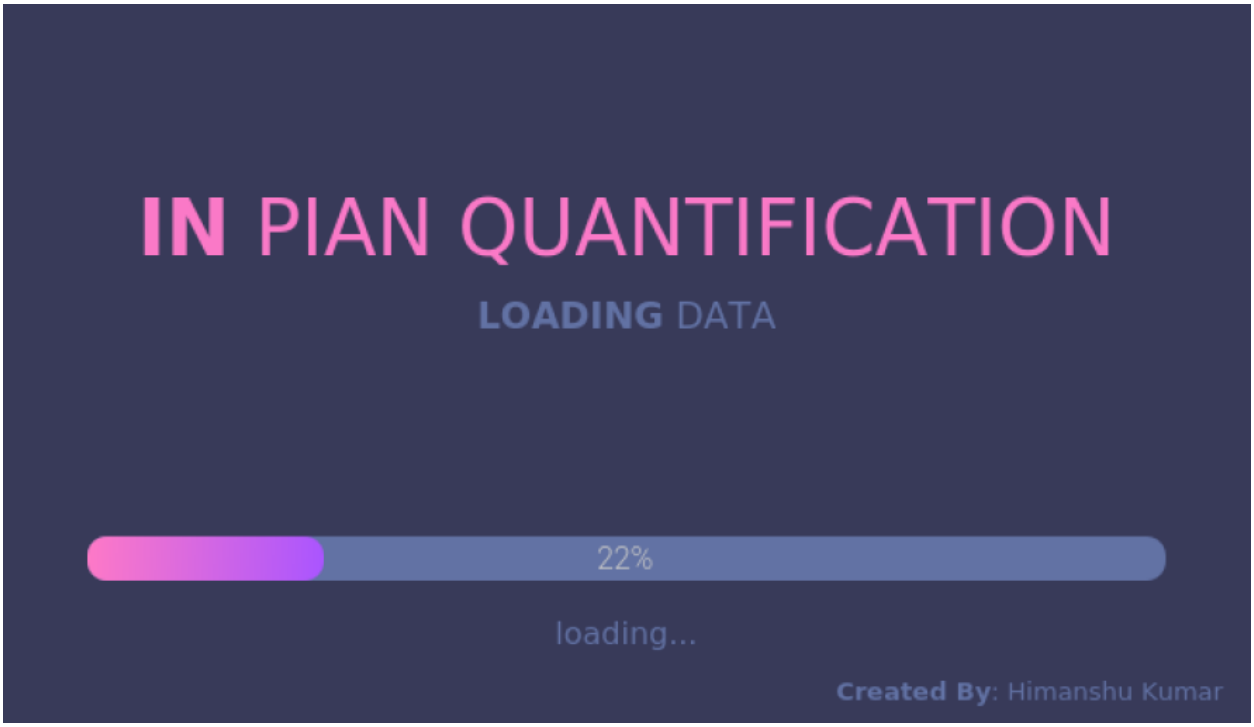

Figure4: Loading Screen (Part of the welcome Screen)

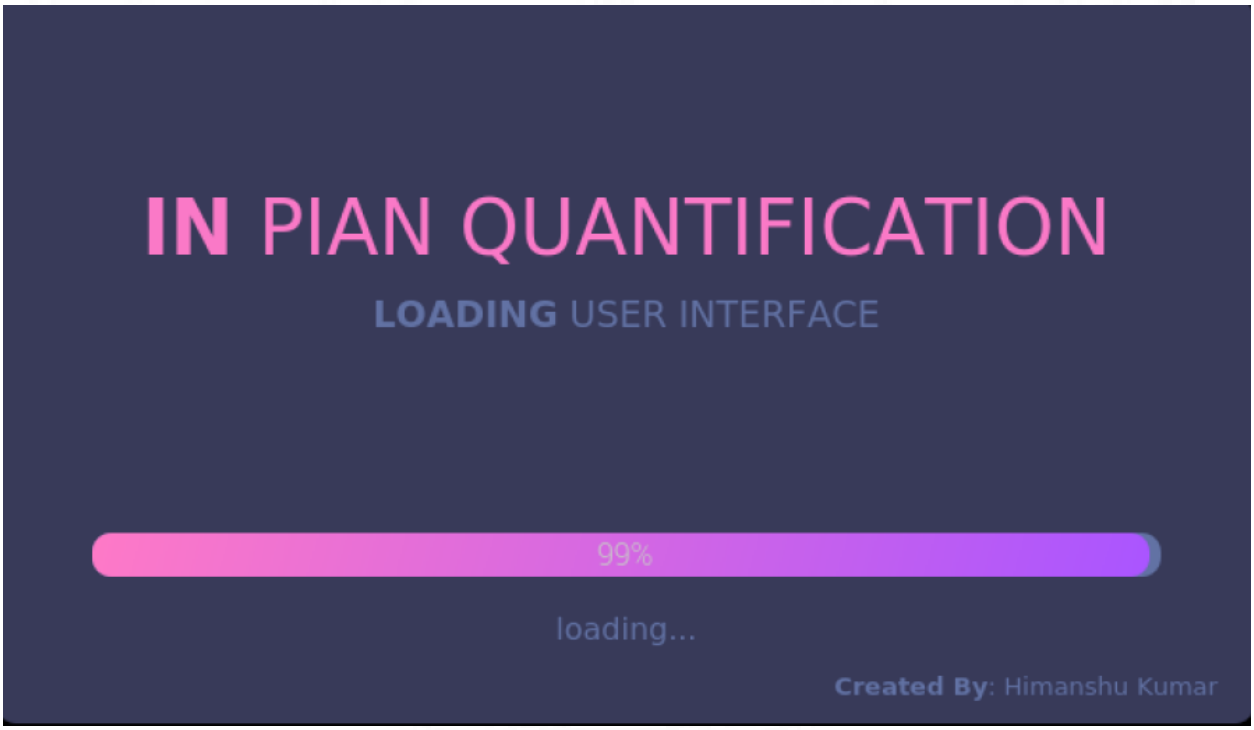

Figure5: User Interface being initialized (Part of the welcome Screen)

Part of code responsible for the above screen is as follows:

```
<widget class="QLabel" name="label_title">
      <property name="geometry">
       <rect>
        <x>0</x>
        <y>80</y>
        <width>400</width>
        <height>61</height>
       </rect>
      </property>
      <property name="font">
       <font>
        <family>Segoe UI</family>
        <pointsize>40</pointsize>
       </font>
      </property>
      <property name="styleSheet">
       <string notr="true">color: rgb(254, 121, 199);</string>
      </property>
      <property name="text">
       <string>&lt;strong&gt;IN&lt;/strong&gt; PAIN QUANTIFICATION</string>
      </property>
      <property name="alignment">
       <set>Qt::AlignCenter</set>
      </property>
</widget>
```

This part of the code Adds the label to the splash screen (i.e. IN PAIN QUANTIFICATION) with the given font family, font size and color gradient.

6.2 Main Window

As the splash screen loads all the necessary data into the main window, the data is being collected in the background. The main Window rendered is as shown in figure6.

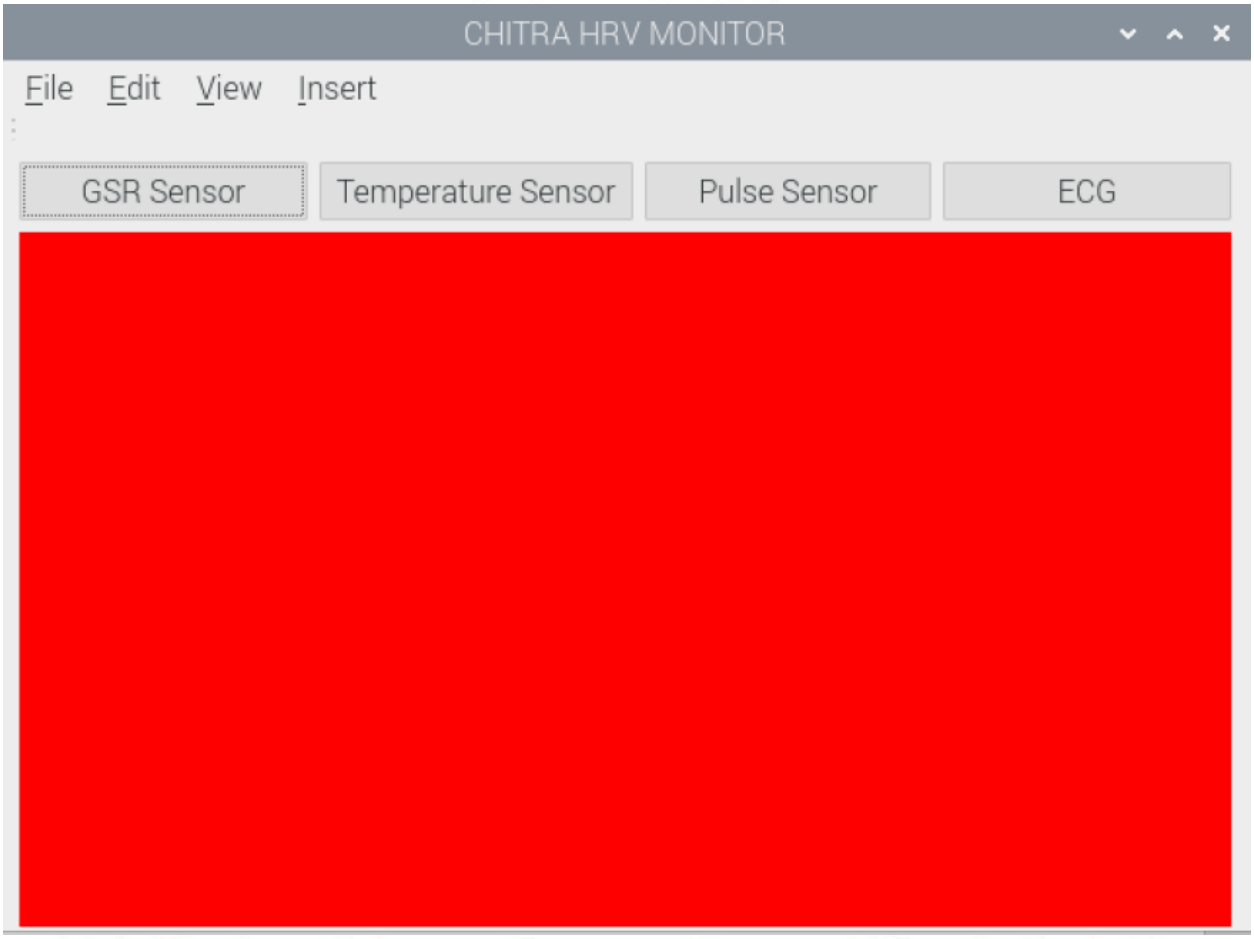

Figure6: The Main Window of the application

The following three line of code renders the main window:

```
if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = SplashScreen()
```

```
sys.exit(app.exec_())
```

6.2.1 The Menu Buttons

The menu buttons (File, Edit, View, Insert) each has two buttons attached to them as shown in figure7. The buttons when clicked lead to an empty area that can be used for future works of opening the CSV file from here and more.

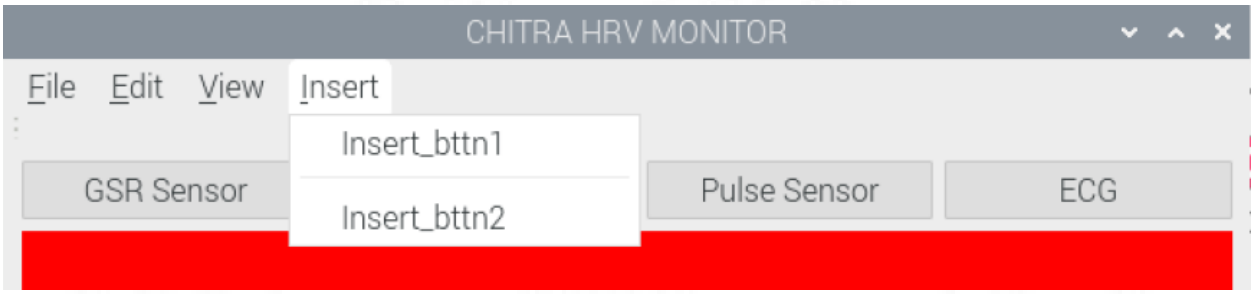

figure7: The menu button has two button attached to them.

When any one of these buttons are pressed, a pop-up dialog box is opened as shown in figure8.

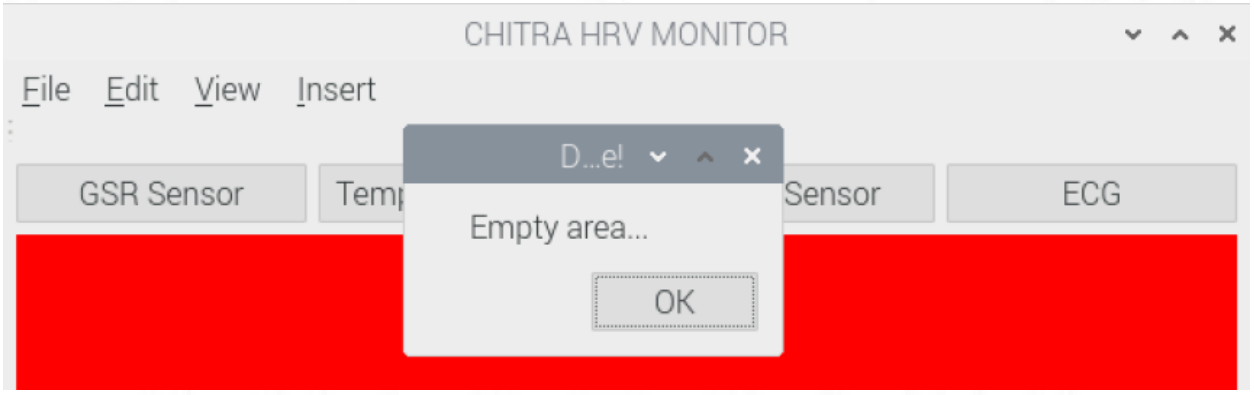

figure8: The popup Dialog box

6.2.2 Serial Data Collection

The serial data collection is done through a serial port which then writes the data collected into a CSV file (‘serial_data.csv’) after parsing the data according to whichever field names they belong to, namely GSR, Temperature, Pulse, ECG.

The below line of code is used to set the port value and the rate at which the said port can receive the data (bits per second)

```
serial.Serial('/dev/ttyACM0', timeout = 1, baudrate=115200)
```

The below code opens a CSV file called ‘serial_data.csv which is then ready to be written into.

```
with open('serial_data.csv', 'w') as csv_file:
    csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
    csv_writer.writeheader()
```

The below line allows data to be read at an interval of 0.1 second.

```
time.sleep(0.1)
```

A sample data read is as follows:

X_value	gsr	temp	pulse	ecg
0	54	60	0	0
1	57	65	946	1006
2	52	65	946	1006
3	53	72	949	1006
4	47	72	949	1006
5	61	71	949	1004
6	53	71	949	1004
7	51	121	946	21
8	49	121	946	21

9	55	63	952	996

6.2.3 GSR Sensor Graph

GSR Sensor Graph button plots the sensor output with respect to time.

The result in one of those cases is as shown in figure9.

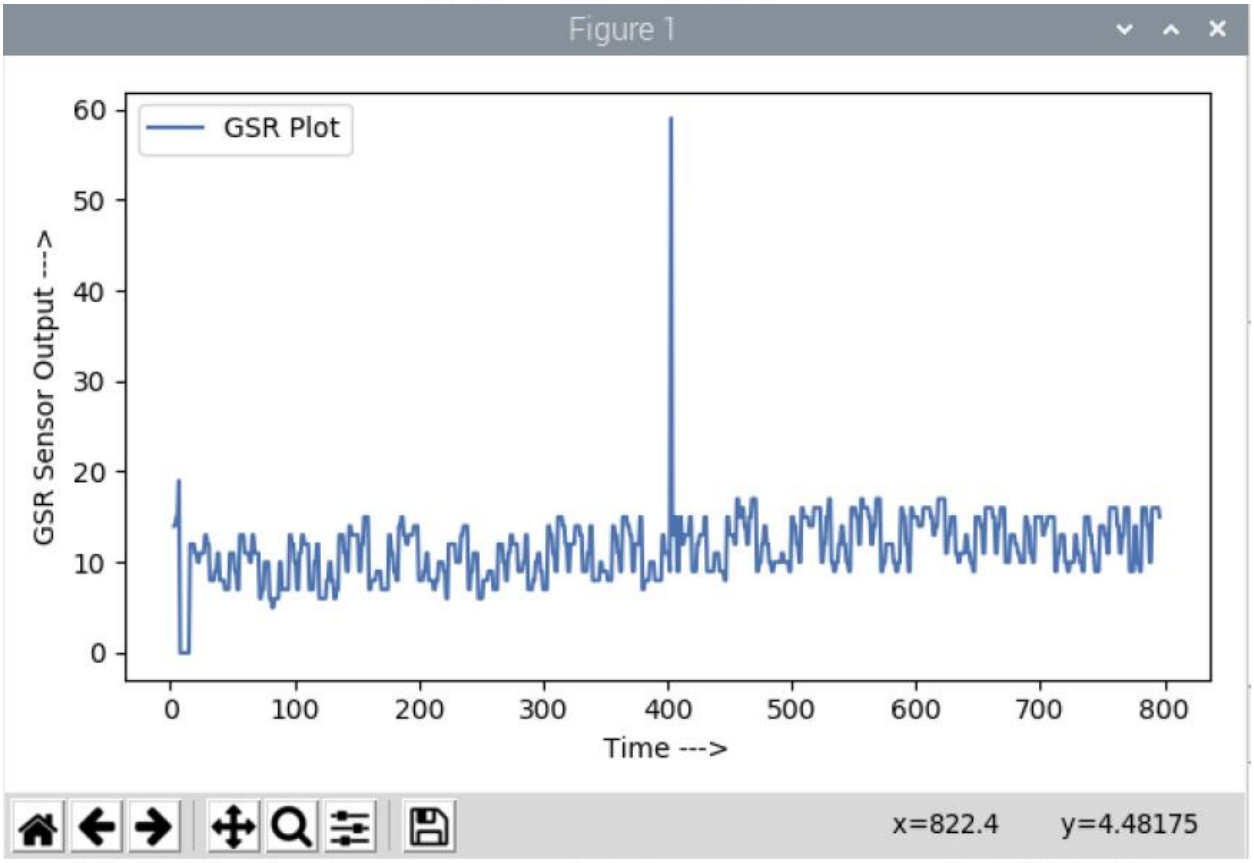

figure9: GSR Sensor plot

Different peaks in GSR action that are not identified with the introduction of a boost are termed to as non-Stimulus-bolted Skin Conductance Responses (NS-SCR). By utilizing the skin conductance esteems, or the quantity of GSR peaks, it's feasible to add quantitative information to investigations of emotional excitement.

Code part:

```
def animate_g(i):
```

```
# reads csv file
data = pd.read_csv('serial_data.csv')
data = data[3:]
# sets the x and y values respectively
x_g = data['x_value']
y_g = data['gsr']
plt.cla()
# Creates the plot
plt.plot(x_g, y_g, label='GSR Plot')
plt.legend(loc='upper left')
```

6.2.4 Temperature Sensor Graph

Temperature Sensor button on the main window allow for the real time capture of the temperature and plot it on a graph. The resulting graph for one such case is as shown in the figure10.

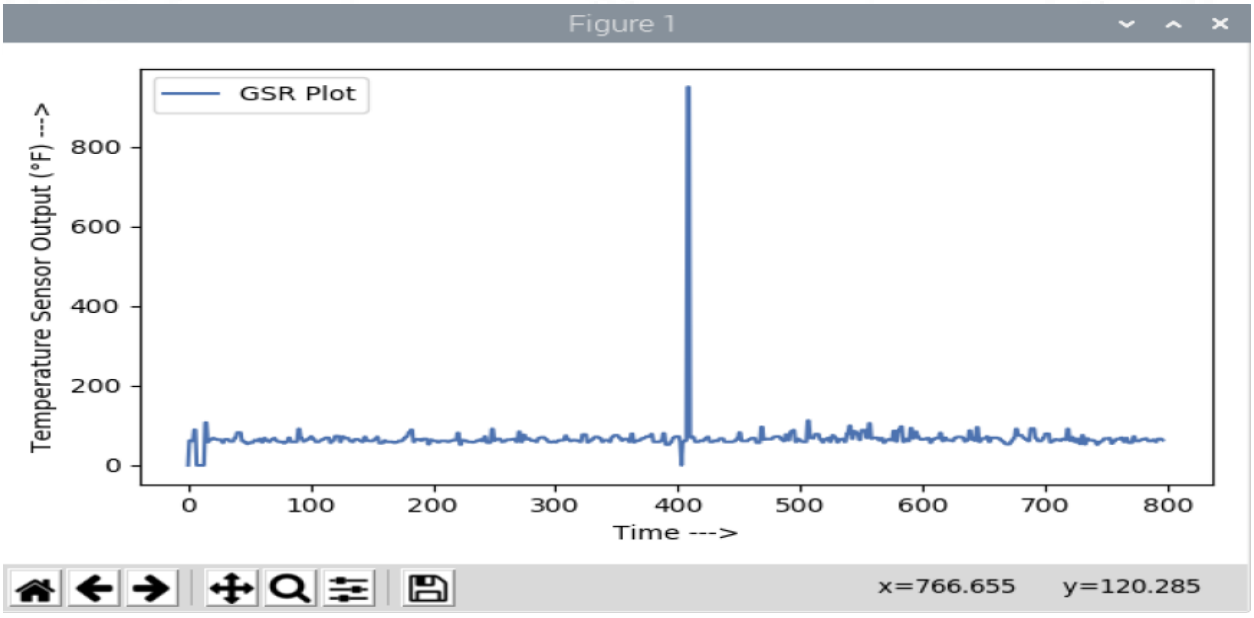

Figure10: Temperature Sensor plot

Some time useless data might creep in the plot value which creates a spike as seen the graph above.

6.2.5 Pulse Oximeter Sensor

Pulse Sensor button on the main window allow for the real time capture of the pulse oximeter sensor output and plot it dynamically. Figure11 shows one such graph.

Pulse oximeter's only job is to measure hemoglobin saturation and not ventilation. It does not in a complete sense provides respiratory sufficiency measurement.

One disadvantage of this device is that since they are calibrated with healthy subjects in mind so, the for critically ill patients and newborns (preterm) the accuracy is poor.

A very common pulse oximeter makes use of an electronic processor and a pair of small LEDs (Light Emitting Diodes) that is facing a photodiode passing through a patient's body part that is translucent like fingertip or an earlobe.

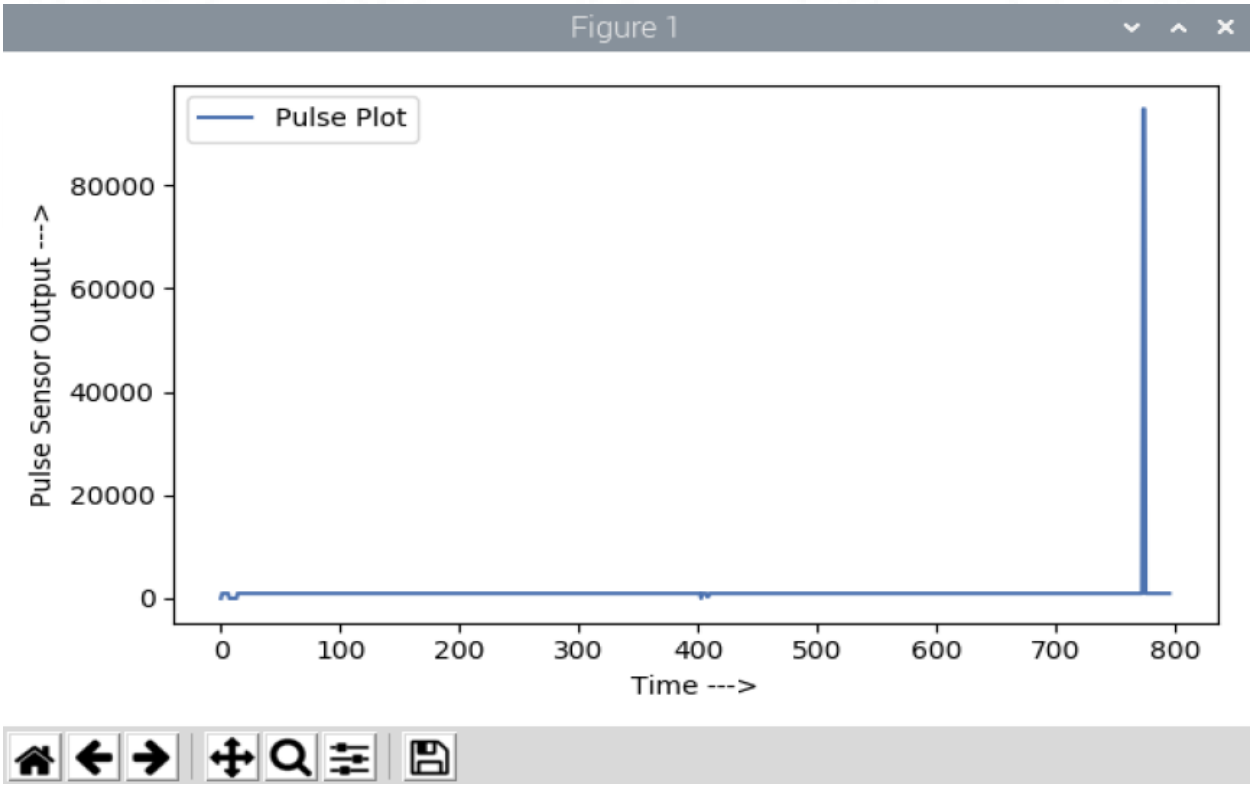

Figure11: Pulse Oximeter Sensor plot

6.2.6 ECG Sensor

ECG button on the main window maps the real time data of the ECG sensor output and provides a plot as shown in figure12.

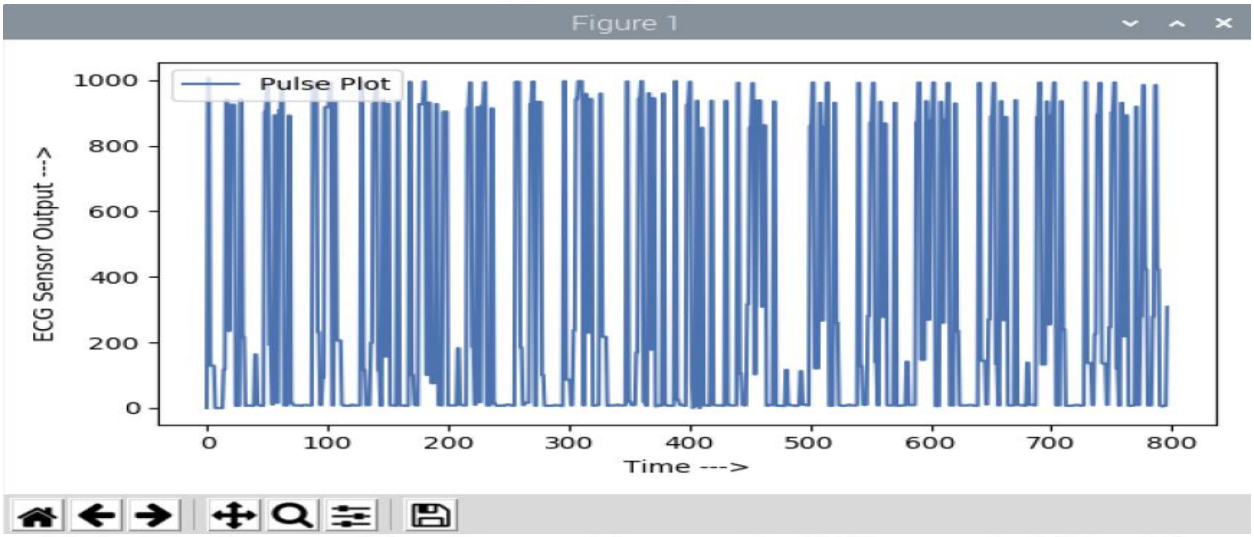

Figure12: ECG Sensor plot

7 GRAPH FUNCTIONALITY

Every graph is equipped with the functionality to save a snippet of it as an image as shown in figure13 below.

Figure13: Save functionality

The circled button with red color when clicked allows the user to save the plot snippet as png file at desired location as shown in figure14 below.

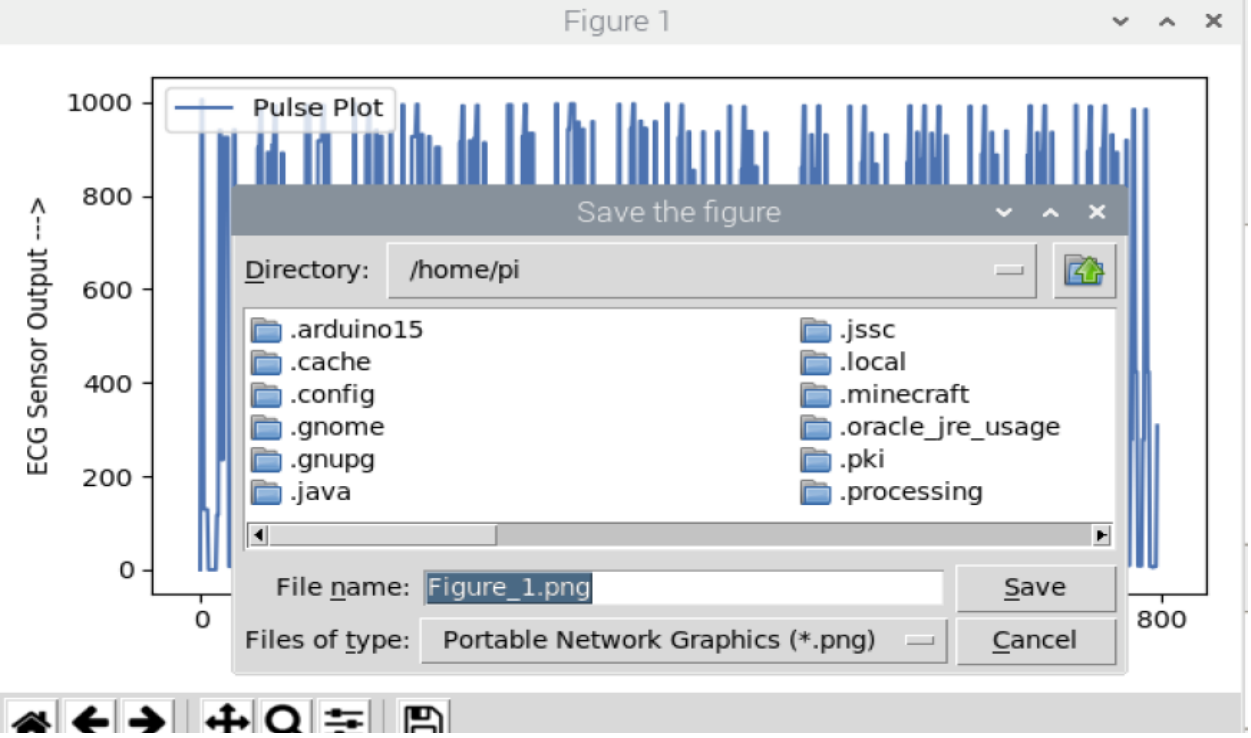

Figure14: save dialog box

So, even though the data is saved is only of the last run, but plots can be saved providing information about previous runs as well.

8 HARDWARE EQUIPMENTS

The hardware mainly consists of a touch screen device with various USB ports connected to it. It has outlet for various Sensors like temperature, GSR, ECG and pulse Oximeter.

It is connected to a power supply of around 5V.

The connection is as visible in the figure15 to figure 17 shown below.

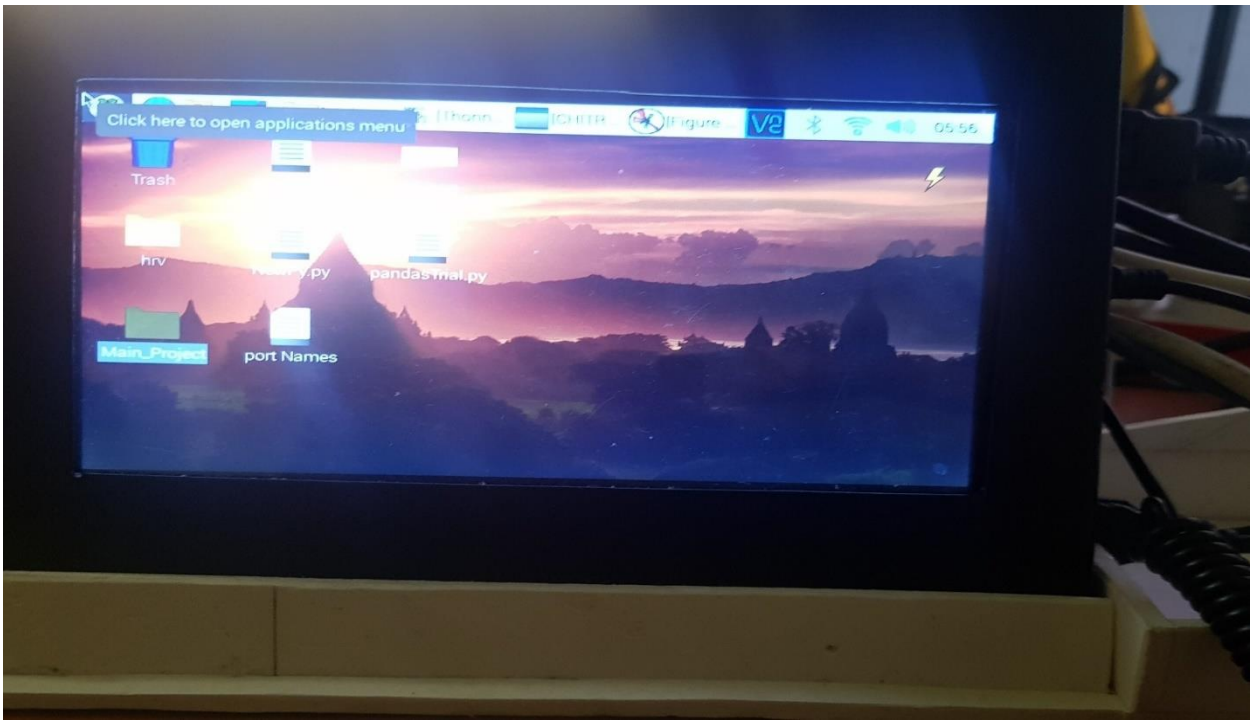

Figure15: Touch screen

The operating system being used on it is Raspbian. It has various applications on it that allow the program to run smoothly. Some these programs that were used in building of this project are:

Mu

Thonny Python IDE

Python (v3.70)

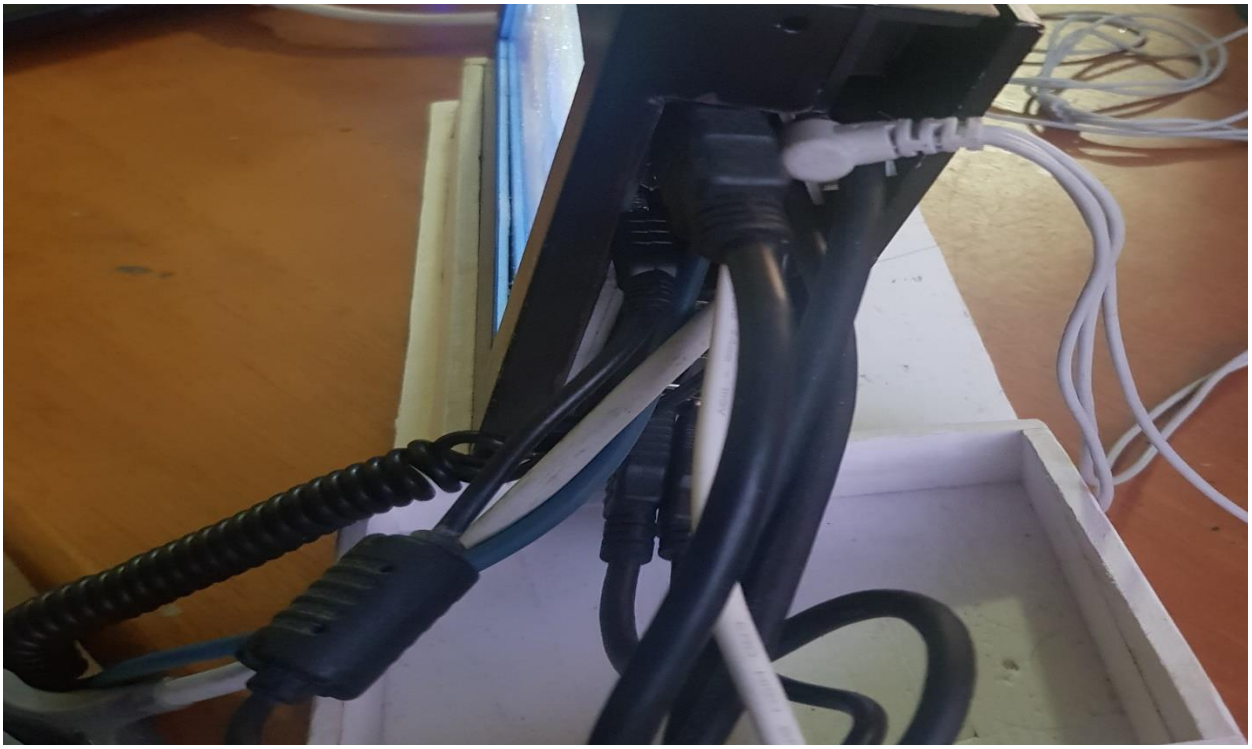

Figure15.1: showing various port outlets from the hardware

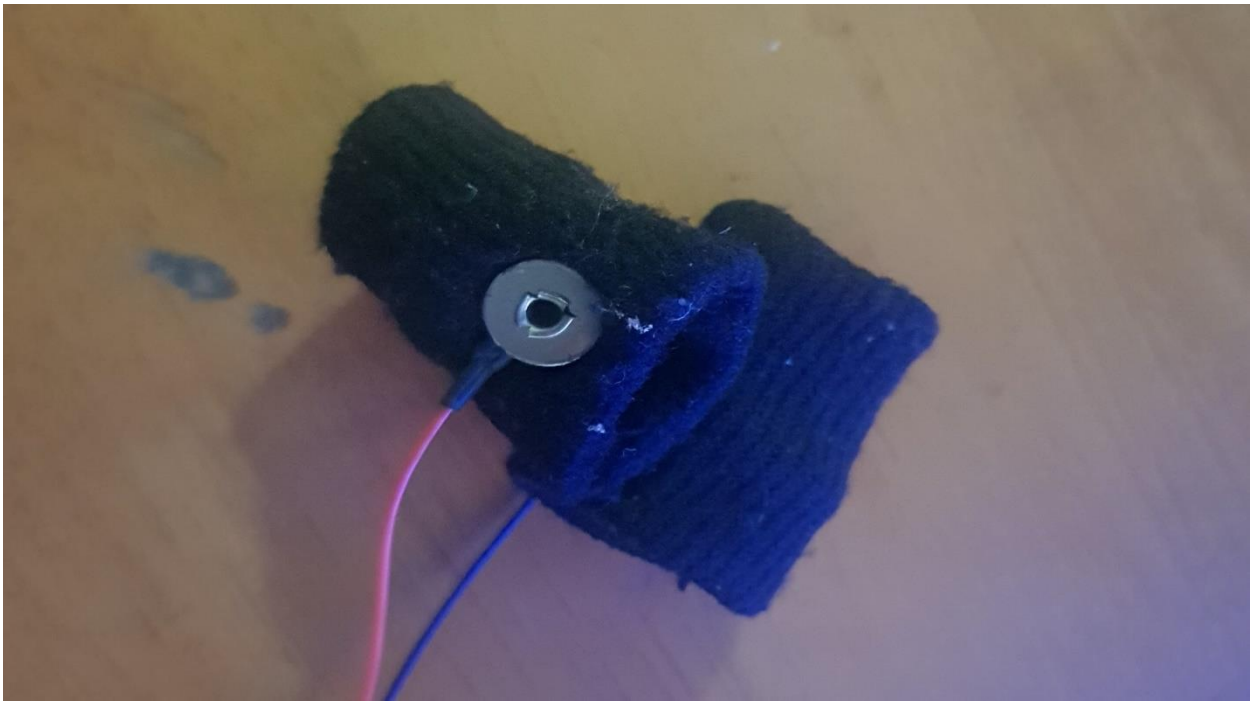

Figure16: GSR Sensor having Ag/AgCl skin Contact part as being shown above.

Figure17: This is the temperature sensor for measuring real time data

Figure18: Pulse Oximeter sensor with LED lights. The black arrow on the right side is where the finger tip is inserted for measurement.

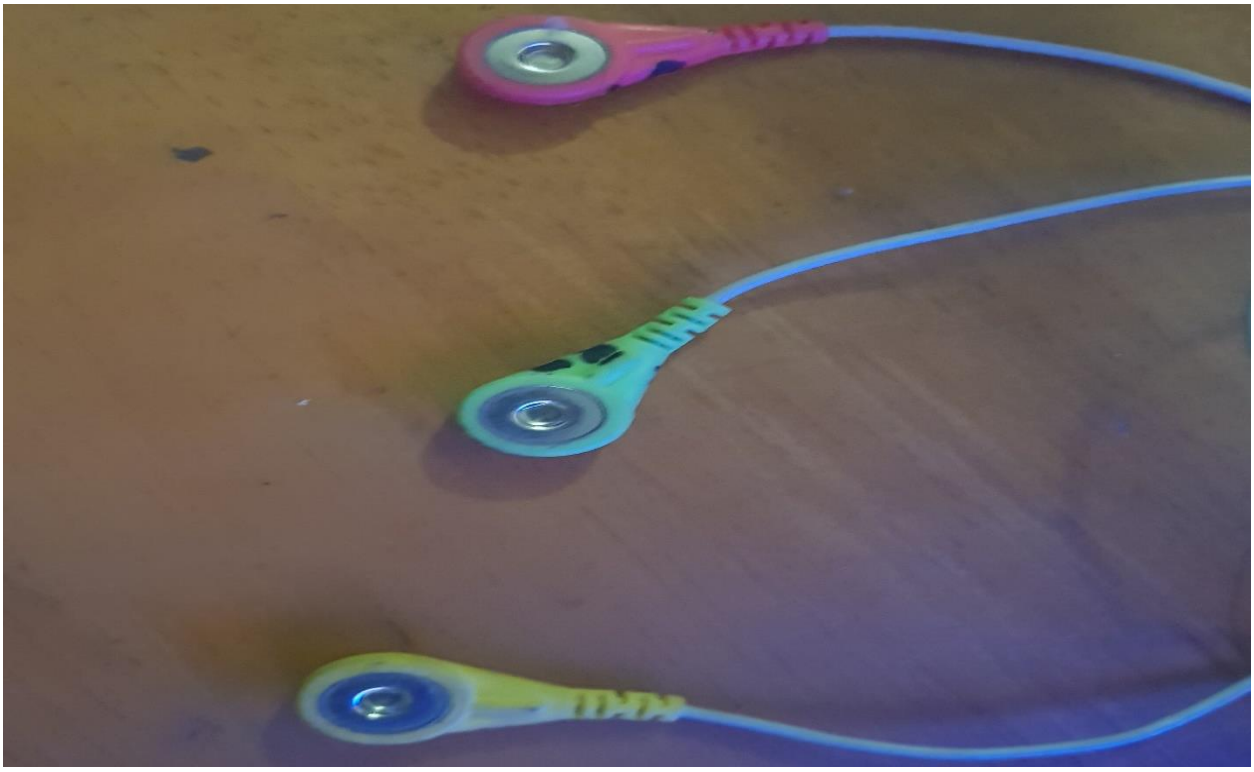

Figure19: ECG sensor (three lead device)

9 RESULTS and DISCUSSION

9.1 Results

The project result consists of software application capable of running on Raspberry Pi system which serially read data over USB connections and store these data in a CSV file and then plot the same data dynamically. There is no need to connect a laptop to provide extra memory or no additional hardware is needed to run the application. Stability is guaranteed as there is no unnecessary updates on this system that allows the application to run smoothly for a longer period of time.

The performance is in lieu with the desired output where currently the data is being fetched every 0.3 seconds and stored on the system itself. There is no need to have an external or even a third-party to store the data. Later on, the interval to fetch the data of sensor output can be changed manually as the requirements of the user.

The application was built to withstand faulty data where sometimes the value received is none or the integer object is not of ‘utf-8’ encoder. To deal with such cases the decoder was set to ‘ISO-8859-1’ which provides an encoding every 255 characters and hence avoids any kind of error.

So, the system along with the application performs in a stable and maintained way. The requirements are met for the said application.

9.2 Discussion

The project goal was a bit hard to achieve. Small ideas and ways to fetch and clean the data took a lot work where errors occurring in the data like [‘’, ‘23’] where the first value is empty also in case of [‘1’, ‘98S3’] where the second item is a non-integer value which is not right. So dealing with all the incorrect data and passing the values accordingly helped in achieving the main window layout.

Proficiency in python and its various libraries like numpy, pandas, PySide2, PyQt, matplotlib and more have significantly increased. The problems encountered were usually due to lack of experience in terms of handling the language as well working on a project from scratch.

A better plot can be drawn if the data being fetched can be cleaned with the right subscripts. For example, say data [‘S34’] should go in pulse sensor data storage as S indicates Pulse, similarly data value[‘G57’] should be appended to GSR (Galvanic Skin Response) sensor data storage as G indicates GSR and so on for Temperature (T) and ECG (P).

As the value of pain can also depend on the various other factors surrounding an individual like if the environment that the individual is currently in, is it beautiful, then pain may subside, in another case if the individual is in a precarious situation, then the pain level might go up.

So, this project can achieve a better result for a person is unconscious. Then no external stimuli will be there and then it can be determined if the value obtained truly determines if the person is in pain or not?

10 CONCLUSION

The splash screen and the Main Window were successfully created with the plot needing some work with whether they need to a line graph or rectangular graph or numerical plots only. With the use of single board computer, the Raspberry Pi made it easier to create an application that can be managed by anyone and everyone having just the basic knowledge of python language and its libraries as well.

The system built has protection against faulty data or redundant data or missing data or even excess data. The system displays GUI (Graphical User Interface) with interaction on touch screen that is built using the Raspberry Pi and Arduino.

So, I have a better understanding of what is to be desired from the said program.

11 FUTURE WORKS

Even though the project has resulted in a functional system. There is scope of improvement.

11.1 Additional Feedback System

In case when the user is not able to interact with the system then facility should be provided that gives user an idea as to what caused the error. The help can be in the form of a message or an email, allowing user to take further and appropriate action.

11.2 Size of the Hardware

Hardware size can be reduced based on what components are needed and then a slimer touch screen can be developed that is easier to carry.

References

[1] https://medium.com/swlh/python-gui-with-pyqt-pyside2-5cca38d739fa

[2] "Raspberry pi 3 model b," [Accessed 02-04-2017]. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[3] Schlaeger JM, Cain KC, Myklebust EK, Powell-Roach KL, Dyal BW, Wilkie DJ. Hospitalized quantify verbal pain intensity descriptors: methodological issues and values for 26 descriptors. Pain. 2020 Feb;161(2):281-287. doi10.1097/j.pain.0000000000001716. PMID: 31599851; PMCID: PMC8109226.

[4] Caldwell JT, Wardlow GC, Branch PA, Ramos M, Black CD, Ade CJ. Effect of exercise-induced muscle damage on vascular function and skeletal muscle microvascular deoxygenation. Physiol Rep. 2016 Nov;4(22):e13032. doi: 10.14814/phy2.13032. PMID: 27884955; PMCID: PMC5358004.

[5] Gablenz EV, Heinen B, Kirsch D, Lanz E. Schmerzerfassung : Beschreibung einer neuen Methode [Measurement of pain.]. Schmerz. 1988 Sep;2(3):144-50. German. doi: 10.1007/BF02528613. PMID: 18415325.

[6] Clark P, Lavielle P, Martínez H. Learning from pain scales: patient perspective. J Rheumatol. 2003 Jul;30(7):1584-8. PMID: 12858463.

[7] Lund I, Lundeberg T, Sandberg L, Budh CN, Kowalski J, Svensson E. Lack of interchangeability between visual analogue and verbal rating pain scales: a cross sectional description of pain etiology groups. BMC Med Res Methodol. 2005 Oct 4;5:31. doi: 10.1186/1471-2288-5-31. PMID: 16202149; PMCID: PMC1274324.

[8] Williamson A, Hoggart B. Pain: a review of three commonly used pain rating scales. J Clin Nurs. 2005 Aug;14(7):798-804. doi: 10.1111/j.1365-2702.2005.01121.x. PMID: 16000093.

[9] Larson AG, Marcer D. The who and why of pain: analysis by social class. Br Med J (Clin Res Ed). 1984 Mar 24;288(6421):883-6. doi: 10.1136/bmj.288.6421.883. PMID: 6423128; PMCID: PMC1441695.

[10] RasPi.TV, accessed 1st June 2021, https://raspi.tv/2013/how-to-use-interrupts-with-python-on-the-raspberry-pi-and-rpi-gpio-part-3

[11] John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib Development team, accessed 21st May 2021, https://matplotlib.org/stable/contents.html