IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 4, Ser. 5 (July. – August. 2025), PP 01-09
www.iosrjournals.org

ECG Signal Processing Using Python

P. Mallika
Sai Krishna
Koti Womens College Ou
Adress:17-1-391/S/367, Singereni Colony
Saidabad Hyderabad 500059

Date of Submission: 21-08-2025 Date of Acceptance: 31-08-2025

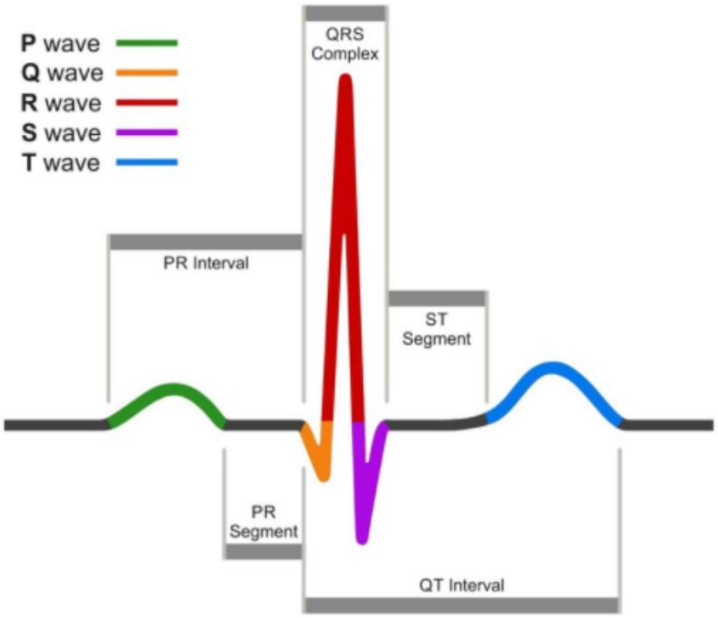
I. Introduction:

ECG Signal:

ECG signal is electrical activity produced due to the activity of the heart. The contractions and expansions of the walls of the heart drive minute electrical currents, which creates different electrical potentials across the body. These changes in electrical potentials can be captured using electrodes, changes in electrical potential are captured by electronic circuitry, which enhances the minute potential differences in the form of varying analog signals,which can be saved for further analysis.

ECG signals contains fiducial points, represented by alphabets **P, Q, R, S** and **T,** these fiducial points provides vital information about the heart condition.

PQRST waveform is depicted in the following image.



Ref:https://www.ems1.com/ems-products/medical-monitoring/articles/quiz-interpreting-cardiac-waveforms-r3tCWbZuBkI5OI9M/

Figure:1

In the above figure QRS complex depicts depolarization of left and right ventricles of the heart. The P wave is the result of the depolarization of the atrium, while the ventricle causes the rest of the peaks.

The information about the morphologies and time duration of the various segments constituting the PQRST waveform provides vital information about the condition of the heart

ECG Signal Processing:

ECG signal can be described as varying voltage in time domain, which represents the activity of heart in electrical form.

ECG signal processing involves activities

- **Signal acquisition**
- **Signal storing**
- **Signal pre-processing**
  - **Baseline wander removal**
  - **Edge Smoothing**
  - **Noise removal**
- **Signal processing**
  - **Feature extraction**
    - **Peaks Detection**
    - **PQRST Waveform Extraction**
    - **PQRST sub segment measurements**

ECG Signal Acquisition:

ECG signal acquisition involves the following components:

- Hardware
  - AD8232 Single Lead ECG Sensor
    - AD8232 is a single lead ECG sensor module. The AD8232 is an integrated signal conditioning block for ECG and other bio-potential measurement applications. It is designed to extract, amplify, and filter small bio-potential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement. This design allows for an ultra-low power analog-to-digital converter (ADC) or an embedded micro-controller to acquire the output signal easily. The AD8232 can implement a two-pole high-pass filter for eliminating motion artifacts and the electrode half-cell potential. This filter is tightly coupled with the instrumentation architecture of the amplifier to allow both large gain and high-pass filtering in a single stage, thereby saving space and cost. (ref: www.analog.com)
  - ESP32 Dev-kit
    - ESP32 is a powerful, generic Wi-Fi MCU modules that have a rich set of peripherals. They are an ideal choice for a wide variety of application scenarios related to Internet of Things (IoT), such as embedded systems, smart home, wearable electronics, etc.ESP32 features a 4 MB external SPI flash and an additional 2 MB SPI Pseudo static RAM (PSRAM) .At the core of the ESP32 is an Xtensa® 32-bit LX7 CPU that operates at up to 240 MHz. You can power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32-S2 integrates a rich set of peripherals including SPI, I2S, UART, I2C, LED PWM, TWAI® controller, ADC, DAC, touch sensor, temperature sensor, as well as up to 43 GPIOs. It also includes a full-speed USB OTG (OTG) interface to enable USB communication. (ref: www.espressif.com)

- Python application
  - Python application to read sensor values from the hardware kit and store in the computer as text file for further processing. Hardware kit sends sensor values to the computer over wifi, the application running on the computer acts as tcp server, where ESP32 acts as tcp client. Tcp client connects to the server using a designated port, once connected the client pushes the data to the server through an assigned socket

Hardware Architecture and Components

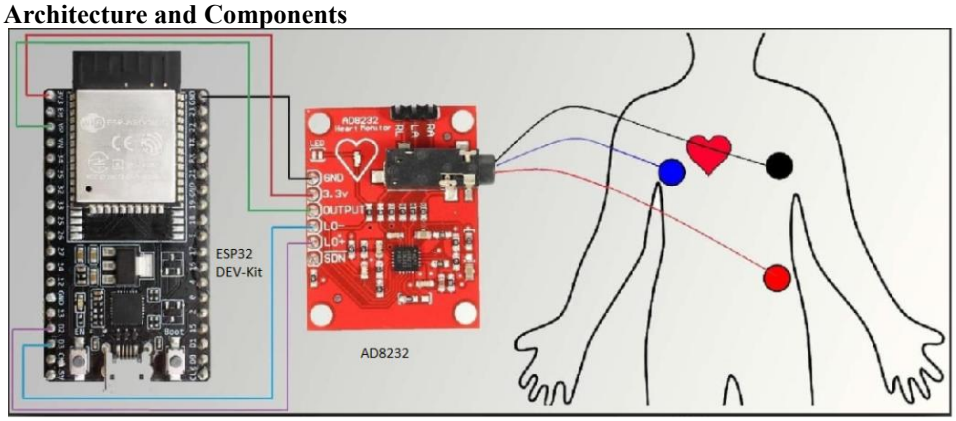


Image Ref: Https://How2electronics.Com/Iot-Ecg-Monitoring-Ad8232-Sensor-Esp32/

ECG Signal Storing:

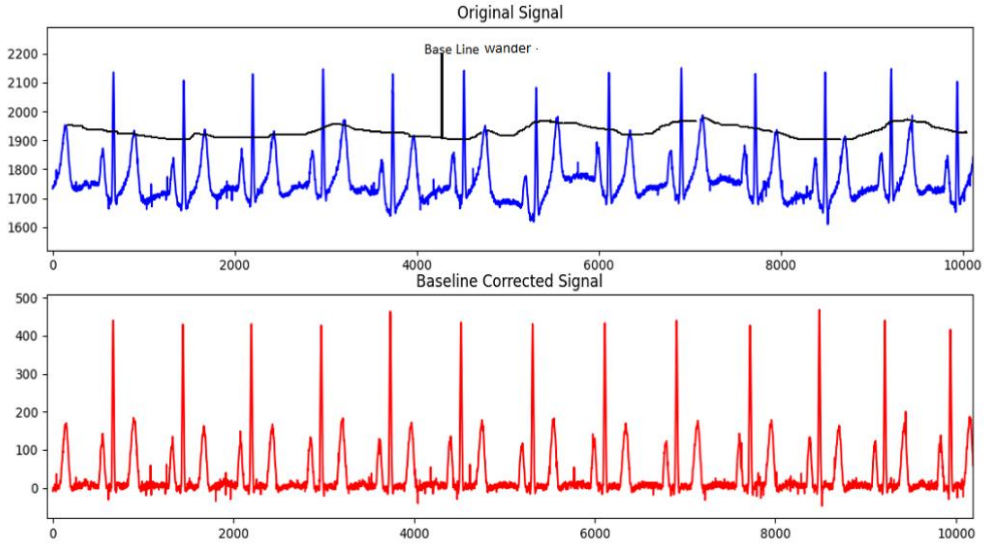
The information captured from ECG sensor is sent to the tcp server as comma separated text, information contains two parameters i.e analog value and time stamp. The parameters received are stored in a text file with new line character as record separator.

ECG Signal Pre-Processing:

Baseline wander removal

There are several kinds of noise present in the ECG signal, noise gets inducted in the actual ECG signal during the process of adc conversion. Mainly this article deals with eliminating noise due to power line(50hz or 60hz) and removal of baseline wander.

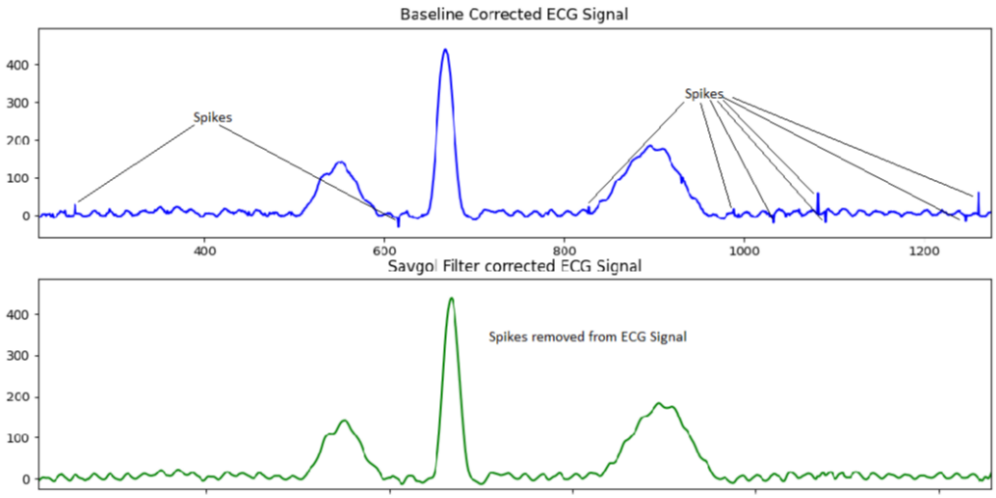
The following image shows how the baseline wandering affects the integrity of ECG signal. The image has two sections, the first section of the image shows ECG signal with wandering baseline and the second section shows ECG signal after removal of baseline wander.



Baseline Removal python library is used for baseline/correction and removal. **Baseline Removal** python library is implementation of Modpoly, IModpoly and Zhangfit algorithms. For Modpoly and Imodpoly polynomial degree needs to be specified. In our case polynomial degree 3 worked well.

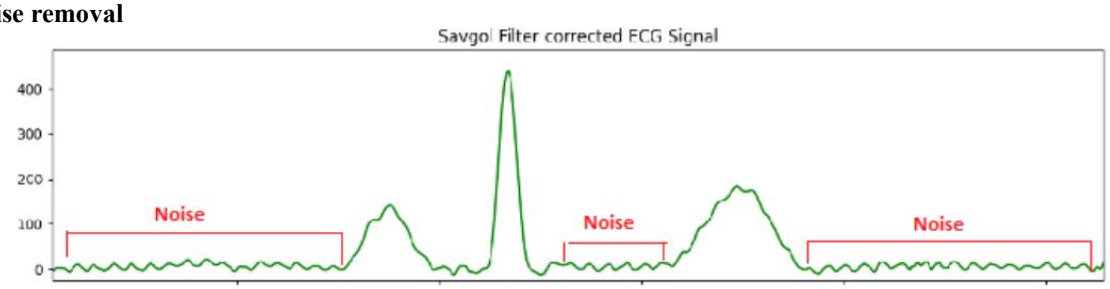
Edge Smoothing

The image below shows the implementation of Savitzky–Golay filter, in short referred as Savgol filter. First half of the image shows spikes in the baseline corrected ECG signal. Second half of the image shows Savgol filtered ECG signal, spikes are eliminated while maintaining the signal precision and integrity



Scipy python package is used and from this package scipy.signal.savgol_filter library is used for the implementation of savgol filter. The function scipy.signal.savgol() takes three mandatory parameters
1. One dimensional array (input data to smoothened)
2. Window size, i.e. No. Of points used to calculate the fit
3. Order of the polynomial function used to fit the signal.

Noise removal

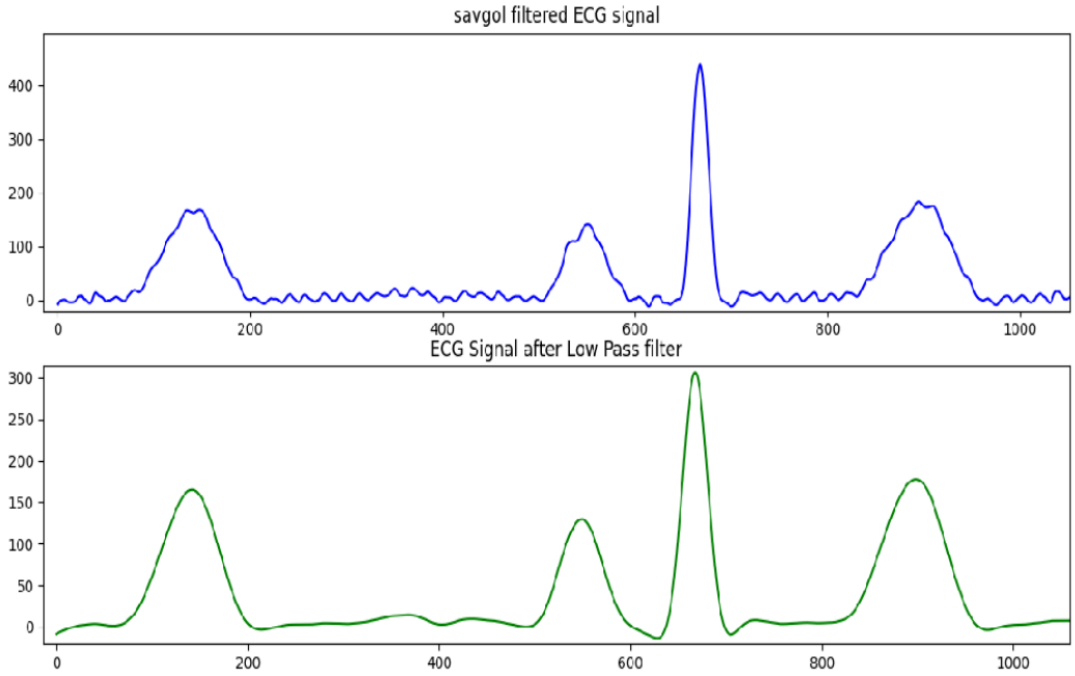


The above image shows high frequency noise marked in red, the noise is mainly due to power-line interference (50 – 60hz frequency). It is clearly evident in the above shown image, high frequency noise riding the ECG waveform. The operating frequency of ECG waveform is between 0.05 – 100 Hz. All the frequencies above this range are considered noise. In order to remove unwanted noise a low pass filter needs to be implemented, which lets low-pass signal to pass and blocks all high frequency signals.

In order to implement low-pass filter, Butterworth low-pass filter is implemented. Butterworth filter eliminates noise,mainly high frequency ripples by allowing low frequency signal to pass and blocking high frequency signals.

From Heartpy package Butterworth low pass filter is used for implementation. Heartpy is python implementation of various filters and pre-procesing functions for bio-signal processing. In order this function, cutoff frequency, sample rate and order needs to be specified to achieve the desired results.
The following shows the implementation of Butterworth low-pass filter on the ECG waveform.

The top-half shows the ECG signal with high frequency noise, the bottom half shows the ECG signal without the noise



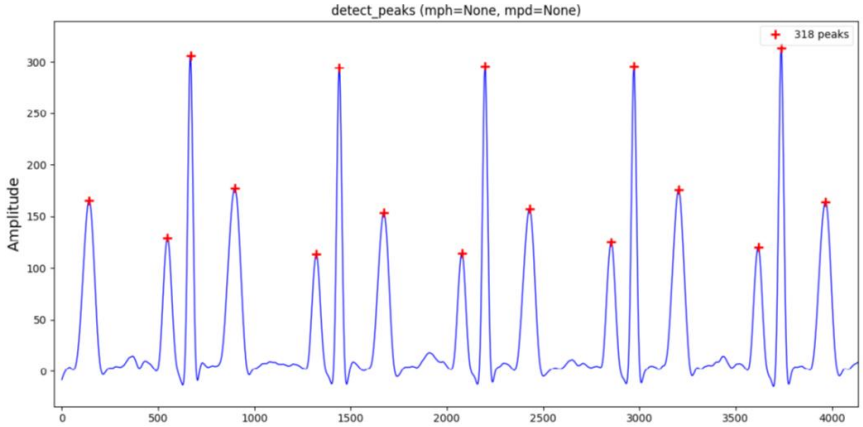
ECG Signal Processing:

After removal of noise noise from the ECG signal, we get a clean ECG signal,with all the features preserved to the maximum extent. As it is clearly visible in the above image,the ECG signal in the color green is carrying all the features, without the noise and baseline wander. Now the signal is ready for further processing.

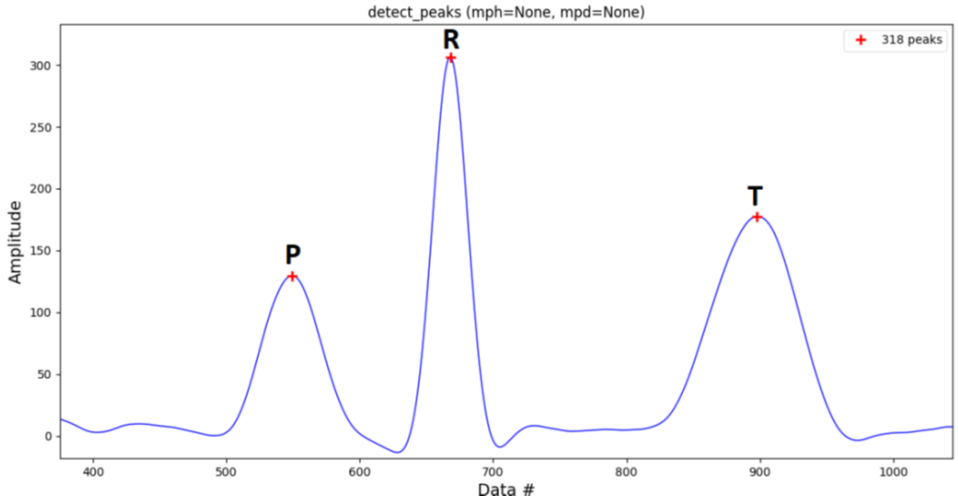
Feature Extraction

Peak Detection

Peak detection is very crucial in ECG signal processing. The main purpose of peaks detection is identify fiducial point in the ECG wave in order to extract the complete PQRST wave form from the signal captured



The above image shows the accurate prediction of **P**, **R** and **T** peaks



The above image is the zoomed in version of the previous image. Clear marking of the peaks can be seen with symbol(+). The highest peak is R peak, the peak on the left is **P** peak and the peak on the right of **R** is **T** peak.

Python library used is detecta (ref: https://nbviewer.org/github/BMClab/BMC/blob/master/notebooks/DetectPeaks.ipynb).

Detectpeaks is the function used from detecta library. It works on the principal of detecting local maxima and minima the function prototype is as below

ind = detect_peaks(x, mph=**None**, mpd=1, valley=**False**)

x – 1d array

mph – minimum peak height

mpd – minimum peak distance

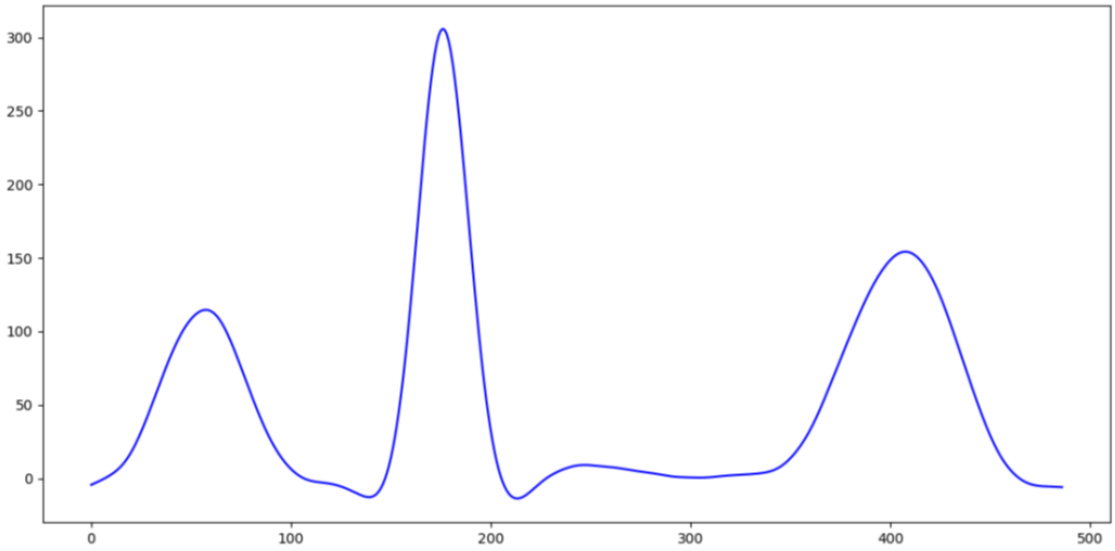
valley – if true detects valleys instead of there are several other parameters, which the function accepts, which we ignored in our implementation

PQRST Waveform Extraction

Once P,R and T peaks are identified, then th process of extracting the complete PQRST waveform is initiated.

The peak indexes are divided into subsets of three indexes, where the middle peak is highest. Based on this criteria, we will be able to get P, R and T peaks of single PQRST waveform

The image below shows the extracted waveform



PQRST Sub segments measurements

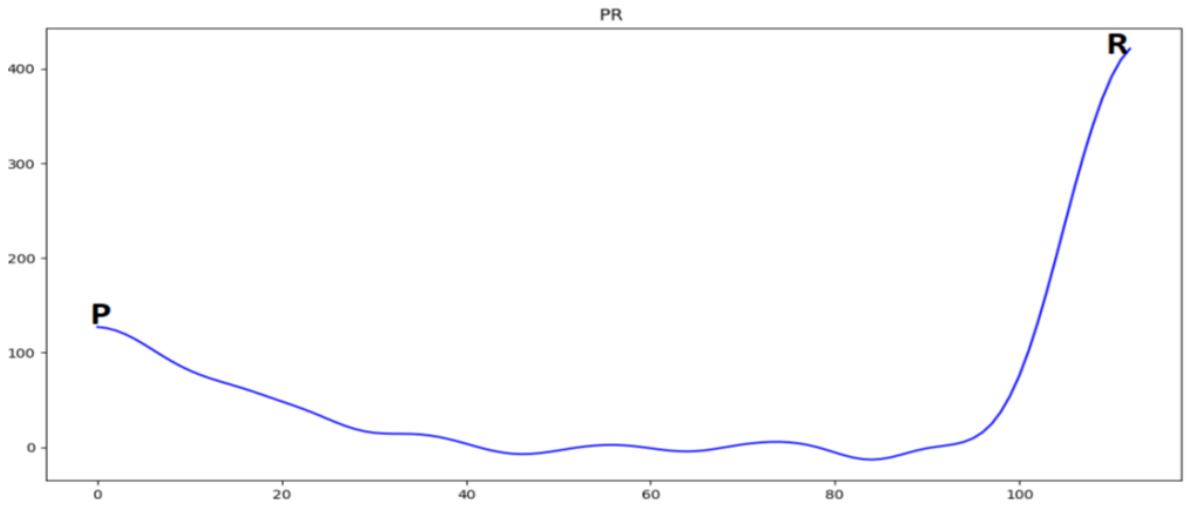
The following are the sub segments to be extracted from PQRST waveform

1. QRS duration
2. PR interval
3. QT interval
4. ST interval
5. ST-T interval
6. PR segment interval
7. P-wave interval
8. T-wave interval

Since the raw data captured from the sensor also contains time stamp, once we identify the start and end index of a particular segment, we can easily calculate the time duration of that particular segment.

The following part will show how segments are extracted, in particular the below example will show the extraction of **QSR segment,** the same logic is applied in the extraction of all other segments.

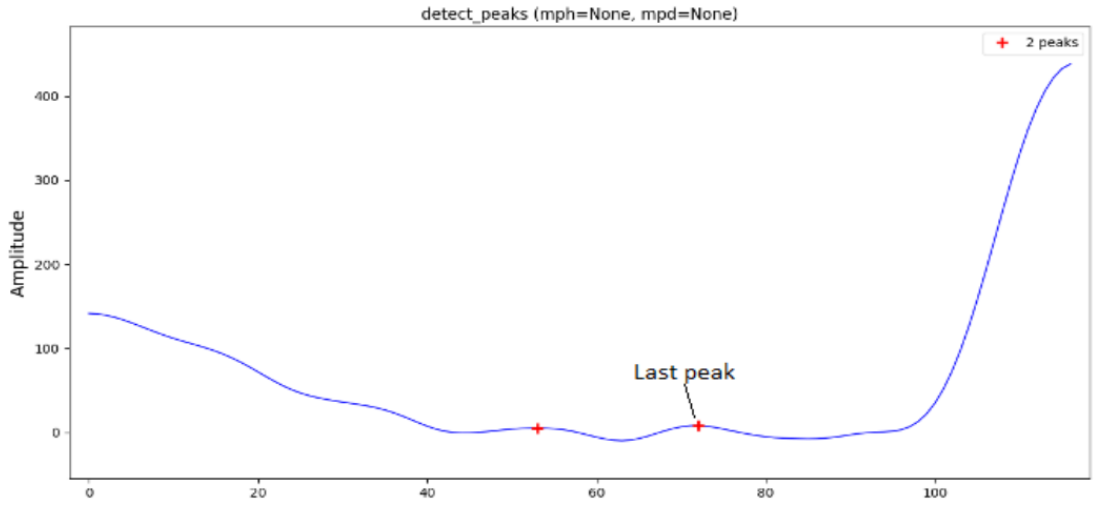
Since the we already have three peaks corresponding to P,R and T, first we need to extract waveform between peak **P** and **R**



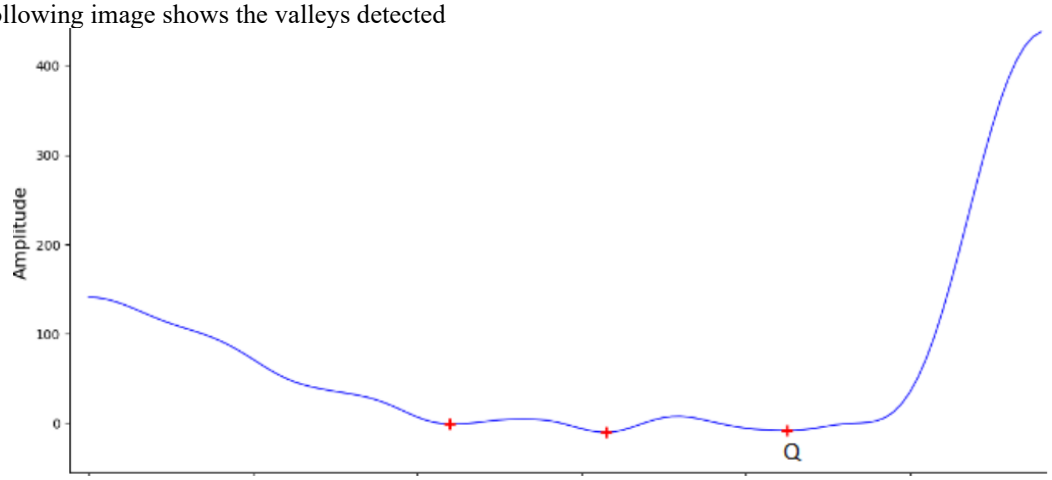
The above image show the waveform between **B** and **R** peak

The next step is to detect peaks within this part of the waveform

The following image shows peaks detected

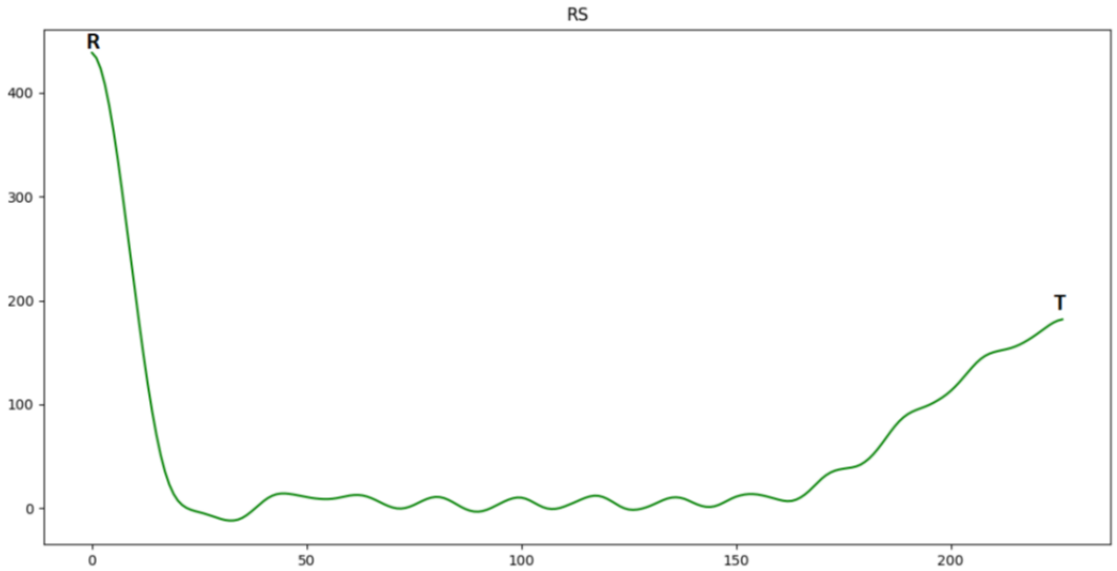


Then detect valleys within this part of the waveform
The following image shows the valleys detected

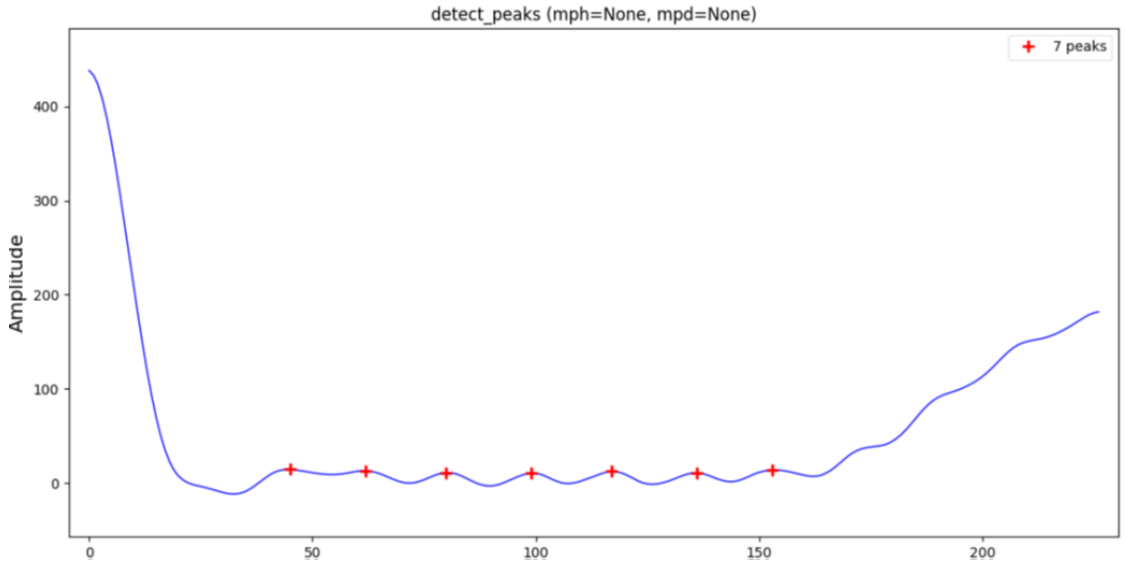


The valley which lies after last peak is the point **Q** of the PQRST waveform

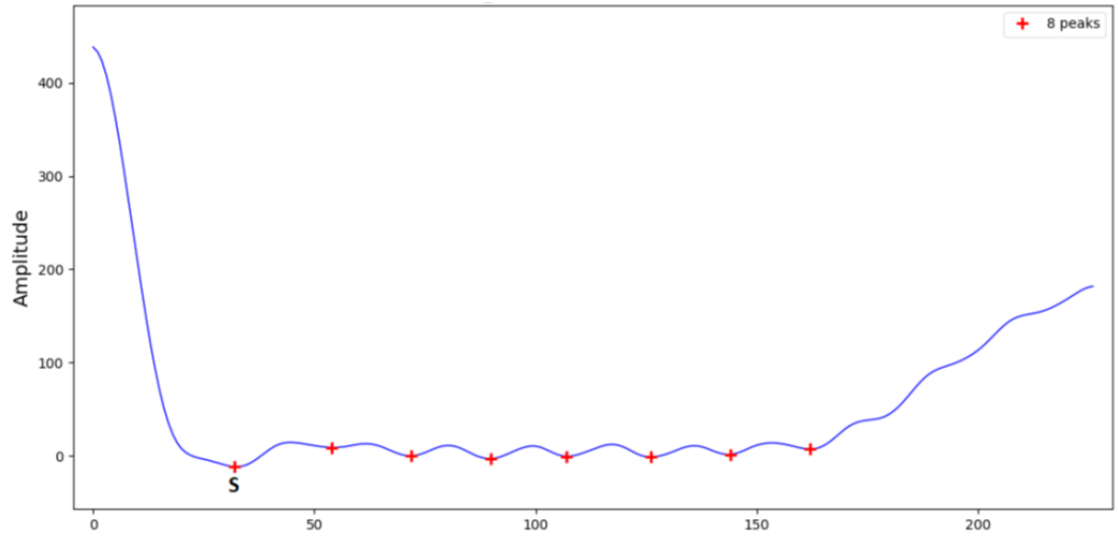
Now **Q** point is estimated and we already have **R,** the next step is to estimate point **S** in the waveform, for that we need to extract RST segment of the PQRST waveform, which is shown in the image below.



Then peak detection within the RT segment as shown in the image below

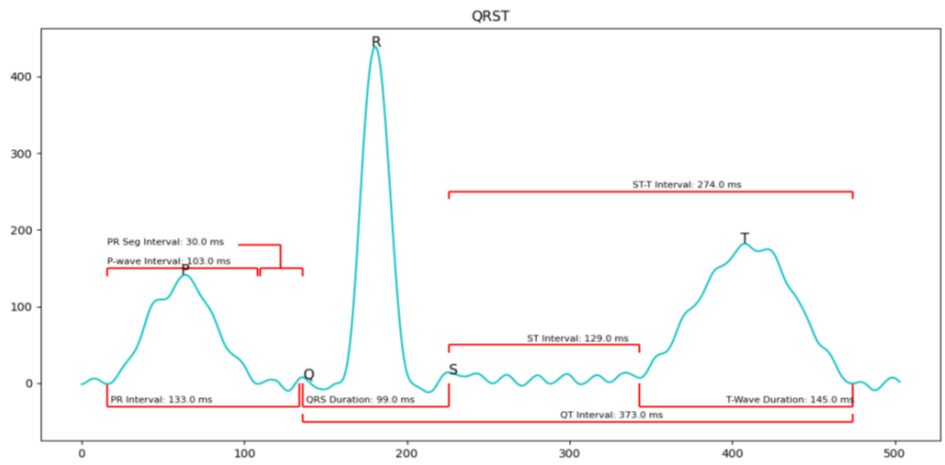


Next valleys needs to be estimated within **RT** segment as shown in the image below



The valley before the first peak represents the point **S** in the **RT** segment

Since we have **Q, R** and **S** points corresponding to ECG waveform, now we can extract timestamp related Q,R and S point in order to calculate QRS complex length in time domain. In the similar way all the other segments can be estimated and length in time domain can be calculated as depicted in the image below.



II. Conclusions:

This article presents one of the may approaches towards processing of ECG signals. Our results based on the data we collected from our hardware produces close to 100 % accurate results. However better electronics design would produce even more accurate results. The focus of this article is on implementing various python libraries in order to extract meaningful information from ECG signal for analysis. The sensor used is single lead ECG sensor which proved heart activity from only one angle, in order to build a medical grade ECG solution , a 12 lead ECG sensor must be used.

References:

[1] Https://Ecgwaves.Com/Topic/Ecg-Normal-P-Wave-Qrs-Complex-St-Segment-T-Wave-J-Point/
[2] Https://Www.Ems1.Com/Ems-Products/Medical-Monitoring/Articles/Quiz-Interpreting-Cardiac-Waveforms-R3tcwbzubki5ol9m/
[3] Www.Espressif.Com
[4] Www.Analog.Com
[5] Https://How2electronics.Com/Iot-Ecg-Monitoring-Ad8232-Sensor-Esp32/
[6] Https://Nbviewer.Org/Github/Bmclab/Bmc/Blob/Master/Notebooks/Detectpeaks.Ipynb