DSP MINI PROJECT
TEAM:MAARI MONSTERS
TOPIC:ANALYSIS OF ECG SIGNAL FOR HEART RATE MONITOR

Team introduction:
EDM19B048 - J SHARATH BABU
EDM19B031 - S SAI KRISHNA
EDM19B026 - N DHANUSH
EDM19B027 - V CHANDAN PAWAR

***Abstract*—This Project aims to implement an algorithm for detecting the heart rate of an ECG signal of a person.This heart detection is very much useful for the patients.**

I. INTRODUCTION

A. What is ECG?

• ECG also known as an electrocardiogram or an EKG, an ECG is a test that detects and records the strength and timing of the electrical activity in your heart. This information is recorded on a graph that shows each phase of the electrical signal as it travels through your heart.

• The electrocardiogram (ECG) shows the plot of the bio-potential generated by the activity of the heart and is used by physicians to predict and treat various cardio vascular diseases.

• One cardiac cycle of ECG signal consists of the P wave QRS complex .

• The first peak (P wave) shows how the electrical impulse (excitation) spreads across the two atria of the heart. The atria contract (squeeze), pumping blood into the ventricles, and then immediately relax. The electrical impulse then reaches the ventricles. This can be seen in the Q, R and S waves of the ECG, which is called the QRS complex. The ventricles contract. Then the T wave shows that the electrical impulse has stopped spreading, and the ventricles relax once again.

B. How ECG signals are produced?

• Electrical signals are produced by contractions in the heart walls which drive electrical currents and create different potentials throughout the body. By placing electrodes on the skin at some positions , we can detect and record this electrical activity in an ECG.

C. Information about Heart Rate

• Heart rate is the speed of the heartbeat measured by the number of contractions of the heart per minute.
• A normal resting heart rate for adults ranges from 60 to 100 beats per minute.
• Heart rate can be measured using ECG signal by detecting the peaks over one minute.

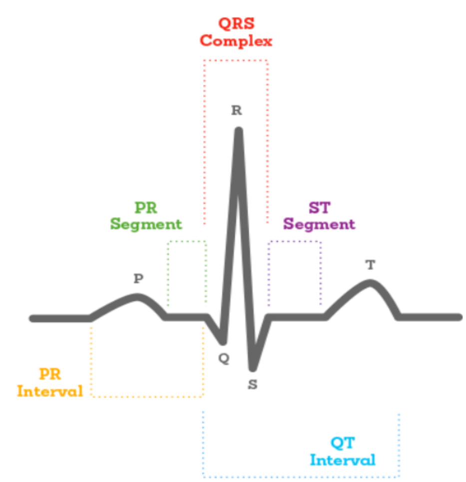

Fig. 1. ECG signal

II. MOTIVATION

There are many applications of ECG signals. It is widely used across the globe to treat the heart patients. Through the ECG signal they determine the heart rate and gives the treatment after analysing it.
This analysis of ECG signals had fascinated us, so we decided to take up this project as it gives the great opportunity for us to get into signal processing in MATLAB. We went with MATLAB to implement an algorithm that detects the heart rate of the ECG signal of a person.

Fig. 2. person taking ECG test

III. Procedure for finding Heart rate in matlab

• Taking an ECG signal of a person as input.
• We have to remove low frequency components of a signal by the following 3 steps:
1.Change the signal into frequency domain using FFT.
2.Removing low frequency components.
3.Bringing back to the time domain using IFFT.
• By using the filter we try to find local maxima of a signal.
• Removing very small values and keeping only significant values.
• Finally using a adjustable filter and recalculating the peaks we get final signal.
Heart rate = (60*sampling rate)/(R-R interval)
R-R means peak to peak distance.

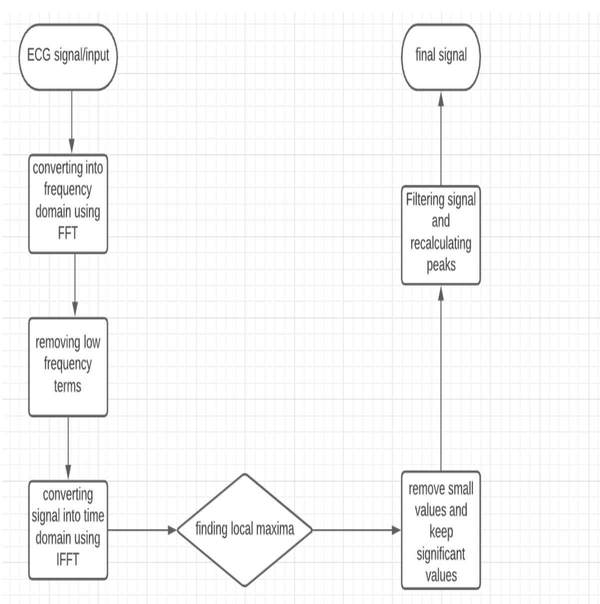

Fig. 3. Block diagram

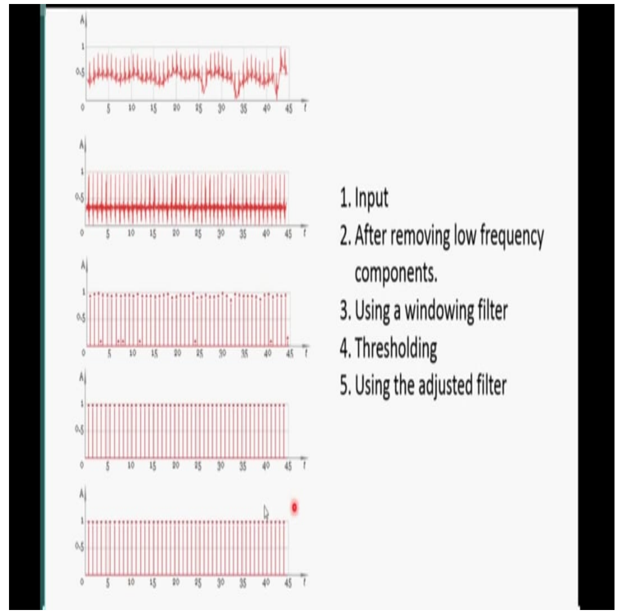

Fig. 4. Block diagram

IV. Matlab code

```
for demo = 1:2:3
    %   Clear our variables
    clear ecg samplingrate corrected filtered1

    %   Load data sample
    switch(demo)
        case 1,
            plotname = ’Sample 1’;
            load ecgdemodata1;
        case 3,
            plotname = ’Sample 2’;
            load ecgdemodata2;
    end

    %   Remove lower frequencies
    fresult=fft(ecg);
    fresult(1 : round(length(fresult)*5/samplin
    fresult(end - round(length(fresult)*5/sampl
    corrected=real(ifft(fresult));

    %   Filter - first pass
    WinSize = floor(samplingrate * 571 / 1000);
    if rem(WinSize,2)==0
        WinSize = WinSize+1;
    end
    filtered1=ecgdemowinmax(corrected, WinSize)

    %   Scale ecg
    peaks1=filtered1/(max(filtered1)/7);
    %   Filter by threshold filter
    for data = 1:1:length(peaks1)
        if peaks1(data) < 4
            peaks1(data) = 0;
```

```
    else
        peaks1(data)=1;
    end
end
positions=find(peaks1);
distance=positions(2)-positions(1);
%   Returns minimum distance between two peaks
for data=1:1:length(positions)-1
    if positions(data+1)-positions(data)<distance
        distance=positions(data+1)-positions(data);
    end
end
% Optimize filter window size
QRdistance=floor(0.04*samplingrate);
if rem(QRdistance,2)==0
    QRdistance=QRdistance+1;           end
end
WinSize=2*distance-QRdistance;

% Filter - second pass
filtered2=ecgdemowinmax(corrected, WinSize);
peaks2=filtered2;
for data=1:1:length(peaks2)
    if peaks2(data)<4
        peaks2(data)=0;
    else
        peaks2(data)=1;
    end
end

positions2=find(peaks2);
distanceBetweenFirstAndLastPeaks = positions2(length(positions2))-positions2(1);

averageDistanceBetweenPeaks = distanceBetweenFirstAndLastPeaks/length(positions2);

averageHeartRate = 60 * samplingrate/averageDistanceBetweenPeaks;

disp('Average Heart Rate = ');
disp(averageHeartRate);


%   Create figure - stages of processing
figure(demo); set(demo, 'Name', strcat(plotname, ' - Processing Stages'));
%   Original input ECG data
subplot(3, 2, 1); plot((ecg-min(ecg))/(max(ecg)-min(ecg)));
title('\bf1. Original ECG'); ylim([-0.2 1.2]);
%   ECG with removed low-frequency component
subplot(3, 2, 2); plot((corrected-min(corrected))/(max(corrected)-min(corrected)));
title('\bf2. FFT Filtered ECG'); ylim([-0.2 1.2]);
%   Filtered ECG (1-st pass) - filter has default window size
subplot(3, 2, 3); stem((filtered1-min(filtered1))/(max(filtered1)-min(filtered1)));
title('\bf3. Filtered ECG - 1^{st} Pass'); ylim([0 1.4]);
%   Detected peaks in filtered ECG
subplot(3, 2, 4); stem(peaks1);
title('\bf4. Detected Peaks'); ylim([0 1.4]);
%   Filtered ECG (2-d pass) - now filter has optimized window size
subplot(3, 2, 5); stem((filtered2-min(filte
title('\bf5. Filtered ECG - 2^d Pass'); yli
%   Detected peaks - final result
subplot(3, 2, 6); stem(peaks2);
title('\bf6. Detected Peaks - Finally'); yl
%   Create figure - result
figure(demo+1); set(demo+1, 'Name', strcat(
%   Plotting ECG in green
plot((ecg-min(ecg))/(max(ecg)-min(ecg)), '-
%   Show peaks in the same picture
hold on
%   Stemming peaks in dashed black
stem(peaks2'.*((ecg-min(ecg))/(max(ecg)-min
%   Hold off the figure
hold off
```

V. MATLAB CODE FOR WINDOW FITER

```
function Filtered=ecgdemowinmax(Original, WinSize)

    %initialising variables
    WinHalfSize = floor(WinSize/2);
    WinHalfSizePlus = WinHalfSize+1;
    WinSizeSpec = WinSize-1;
    FrontIterator = 1;
    WinPos = WinHalfSize;
    WinMaxPos = WinHalfSize;
    WinMax = Original(1);
    OutputIterator = 0;

    % Finding the postion of the largest value in window
    for LengthCounter = 0:1:WinHalfSize-1
        if Original(FrontIterator+1) > WinMax
            WinMax = Original(FrontIterator+1);
            WinMaxPos = WinHalfSizePlus + LengthCounter;
        end
        FrontIterator=FrontIterator+1;
    end
    % if the first point is the highest, set ouput 1
    if WinMaxPos == WinHalfSize
        Filtered(OutputIterator+1)=WinMax;
    else
        Filtered(OutputIterator+1)=0;
    end
    OutputIterator = OutputIterator+1;

    % search next half of signal
    for LengthCounter = 0:1:WinHalfSize-1
        if Original(FrontIterator+1)>WinMax
            WinMax=Original(FrontIterator+1);
            WinMaxPos=WinSizeSpec;
        else
            WinMaxPos=WinMaxPos-1;
        end
        if WinMaxPos == WinHalfSize
            Filtered(OutputIterator+1)=WinMax;
        else
            Filtered(OutputIterator+1)=0;
        end
        FrontIterator = FrontIterator+1;
        OutputIterator = OutputIterator+1;
    end
    for FrontIterator=FrontIterator:1:length(Original)-1
        if Original(FrontIterator+1)>WinMax
            WinMax=Original(FrontIterator+1);
            WinMaxPos=WinSizeSpec;
        else
            WinMaxPos=WinMaxPos-1;
            if WinMaxPos < 0
                WinIterator = FrontIterator-WinSizeSpec;
                WinMax = Original(WinIterator+1);
                WinMaxPos = 0;
                WinPos=0;
```

```
            for WinIterator = WinIterator:1:FrontIterator
                if Original(WinIterator+1)>WinMax
                    WinMax = Original(WinIterator+1);
                    WinMaxPos = WinPos;
                end
                WinPos=WinPos+1;
            end
        end
    end
    if WinMaxPos==WinHalfSize
        Filtered(OutputIterator+1)=WinMax;
    else
        Filtered(OutputIterator+1)=0;
    end
    OutputIterator=OutputIterator+1;
end
WinIterator = WinIterator-1;
WinMaxPos = WinMaxPos-1;
for LengthCounter=1:1:WinHalfSizePlus-1
    if WinMaxPos<0
        WinIterator=length(Original)-WinSize+LengthCounter;
        WinMax=Original(WinIterator+1);
        WinMaxPos=0;
        WinPos=1;
        for WinIterator=WinIterator+1:1:length(Original)-1
            if Original(WinIterator+1)>WinMax
                WinMax=Original(WinIterator+1);
                WinMaxPos=WinPos;
            end
            WinPos=WinPos+1;
        end
    end
    if WinMaxPos==WinHalfSize
        Filtered(OutputIterator+1)=WinMax;
    else
        Filtered(OutputIterator+1)=0;
    end
    FrontIterator=FrontIterator-1;
    WinMaxPos=WinMaxPos-1;
    OutputIterator=OutputIterator+1;
end
```

VI. OUTPUT PLOTS AFTER RUNNING THE CODE IN MATLAB

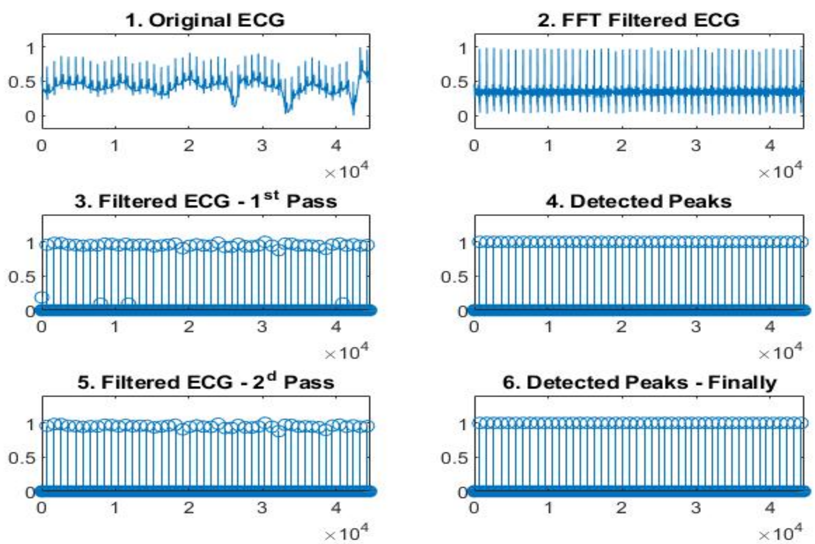

Fig. 5. ECG plot for several operations

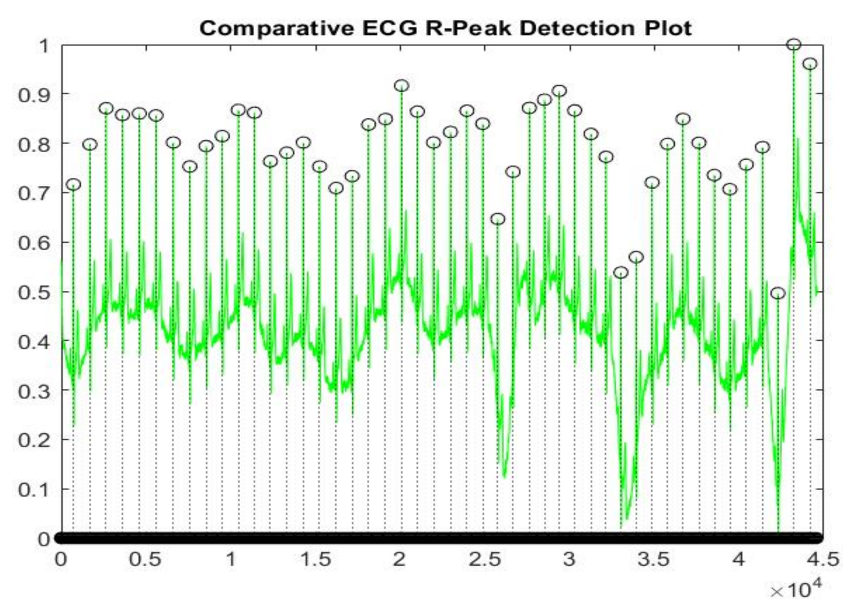

Fig. 6. Final ECG signal

VII. REFERENCES

https://www.theengineeringprojects.com/2016/09/ecg-simulation-using-matlab.html
https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-13-160
www.github.com
The ECG Made Easy by John R Hampton