SAPA Term Project – Portable Ecg Device

PROJECT TEAM

Hamza Karakuş
Muhammed Samed Özmen
Yavuz Selim Durak
Melih Aslan

Table of Contents

1. Introduction

ECG is an electronic device that works on the principle of measuring the bioelectricity of the heart to detect heart problems for all patients in the world. Electrical movements on the heart help the heart muscle to contract and naturally pump the heart's blood. People with heart rhythm disorders have different diseases, as heart contraction works incorrectly. In some cases, people with other diseases can affect their heart rhythms and these diseases cause heart rhythm to deteriorate. With the help of advanced artificial intelligence technologies, it is possible to examine the heart rhythms of people and identify which heart diseases they have or other diseases in their bodies. In order to examine the heart signal, it must be filtered and amplified and transferred to digital devices.

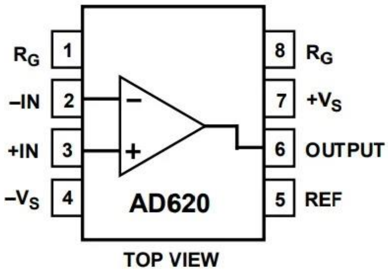
Usually, when receiving a heart signal, people's breathing, and muscle movements, city network signal, and so on, as noise prevent our heart signal from being seen easily. These signals need to be filtered, amplified and transferred to the microprocessors with the help of ADC in an analogous way. The digitalized signals are filtered by digital filters, eliminating noise and providing a clear view of the data. The disease state of the human data can be understood with the obtained data by receiving a heart signal. In this project, we used an AD620 instrumentation amplifier, which includes analog band pass filter, notch filter, amplifier and offset. We converted the analog signal we received from this circuit into a Digital signal using Arduino Uno and performed digital filtering of this signal through Matlab.

We have examined the part that we filtered using FFT in more detail. Using the Graphic User Interface, we plotted the computer screen so that all the signals can be examined by professionals. In this project, all the circuit we use and the intended use are explained.

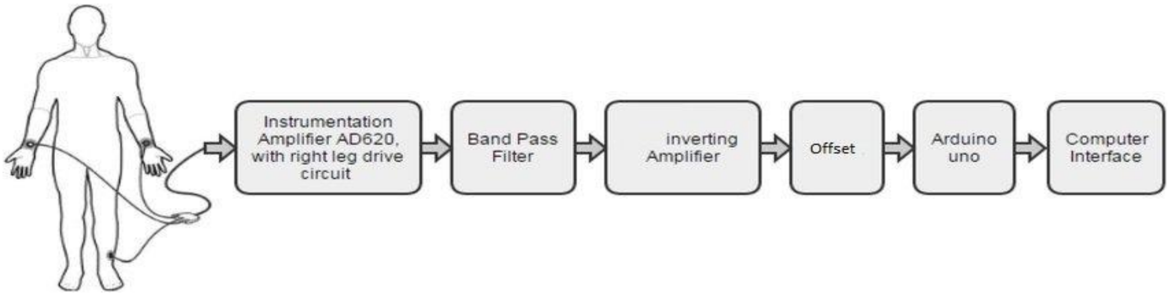
1.1 AD620

AD620 instrumentation amplifier was used in the ECG. The reason to use this Opamp is that it is easy to use, its gain can be adjusted with only 1 resistor, (1 to 10000 ohm), the power supply range is wide (2.3 - 18V), so it can work with low power, 3 opamps it is able to make the gain alone, it has low instrumentation due to its instrumentation, its performance is very high (B Grade) and its bandwidth is high (120 kHz).

Typical Application



(Figure 1.1.1)



(Figure 1.1.2)

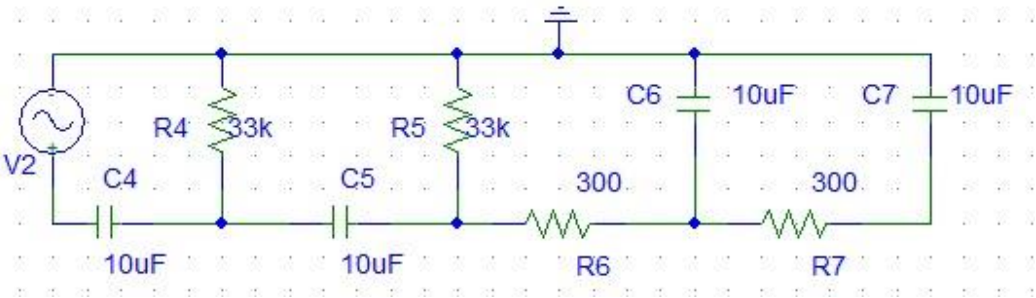
2.Necessary filters, amplifiers and the offset circuit

Introduction

Since our signal was disturbed by many unwanted signals in this process, we began to develop filters to filter these unwanted signal such as high frequency noise, low frequency noise and especially 50Hz. In this part methods and responses of these filter circuits will be outlined.

2.1.1 Passive Bandpass Filter

We used this filter to get the ECG signal more clearly and to process our signal more efficiently in the digital process. So we have used a *2^{nd} order bandpass filter (figure 2.1.1)* which is transmitting signals between 0.5Hz and 50Hz.



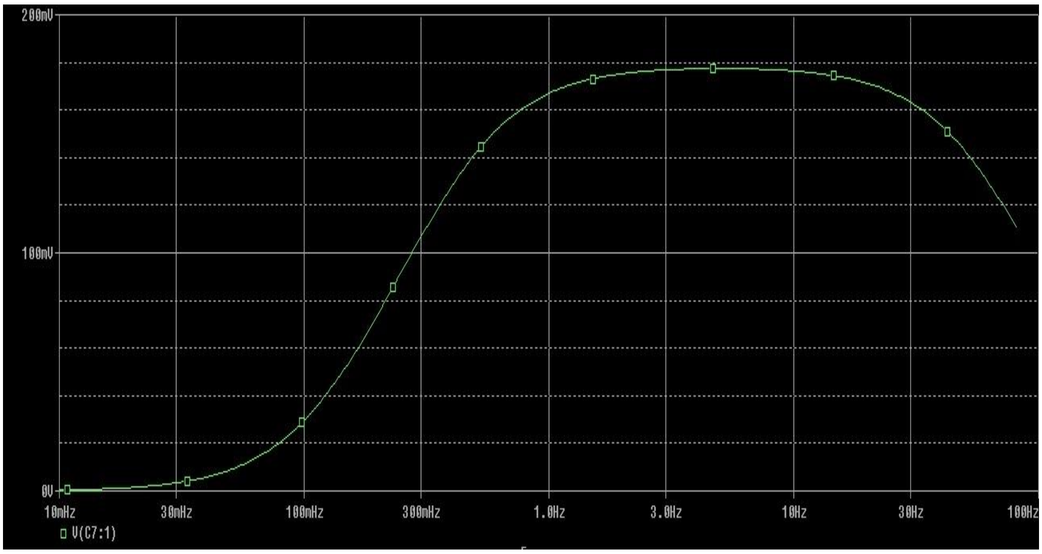
(figure 2.1.1)

As can be seen on figure 2.1.1 we have used *2^{nd}* order low pass filter with cut off frequency 0.5Hz and *2^{nd}* order high pass filter with cut off frequency 50Hz. We have calculated our resistor and capacitor values by using formula as noted in figure 2.1.2

$$f_{cutoff} = \frac{1}{2\pi RC}$$

(figure 2.1.2)

After designing our circuit we have simulated it with Pspice and we have checked bandpass filter's response in frequency domain to observe filter's efficiency. Mentioned in figure 2.1.3

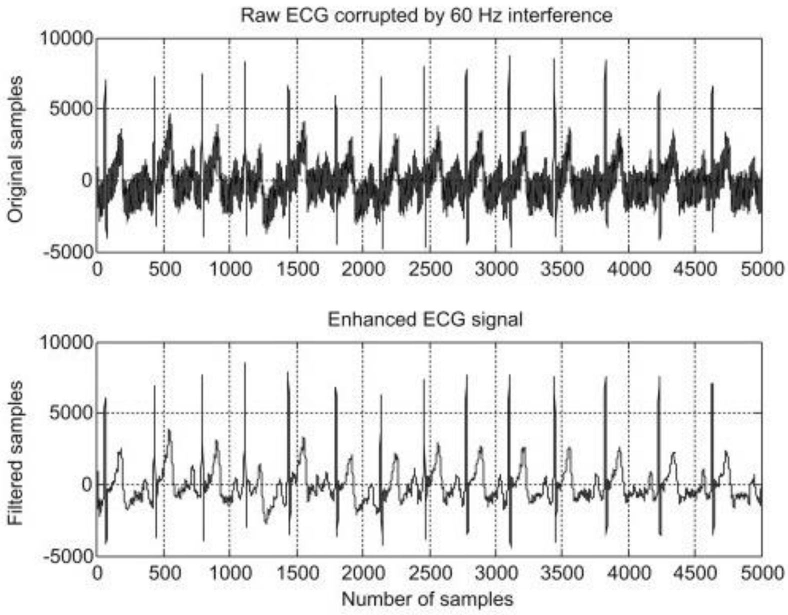


(figure 2.1.3)

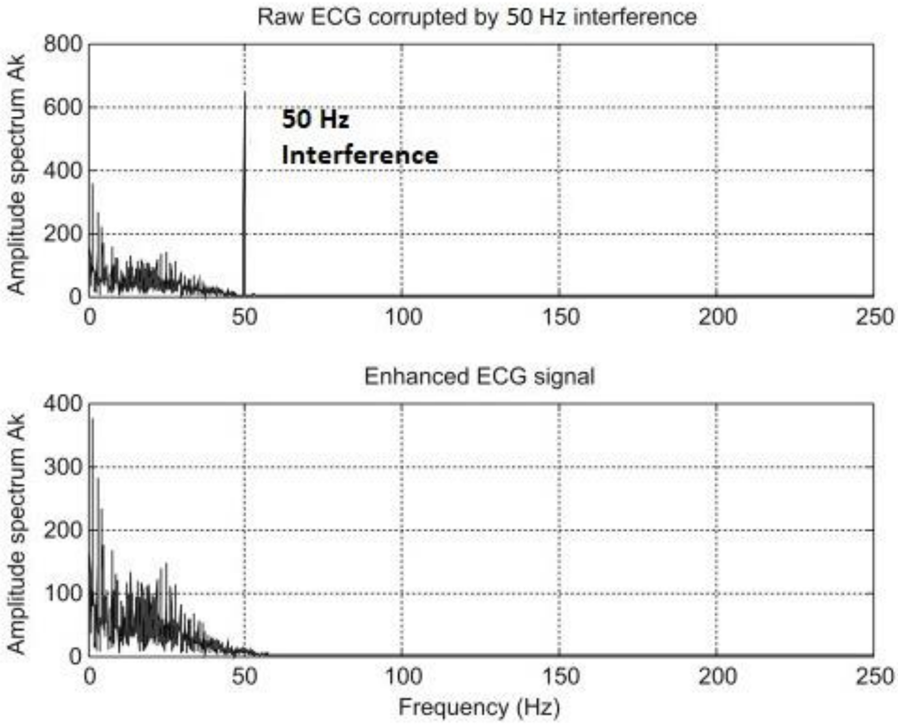
After simulating bandpass filter in Pspice simulation we decided that the filter is working efficient enough to filter our signal between 0.5Hz and 50Hz. Then we built our bandpass filter circuit on pertinaks circuit board.

2.1.2 Notch Filter

During this process, we noticed that our Ecg signal was greatly affected by 50Hz noise, and decided to design a notch filter specifically to block signals at this frequency. We made some pre-research about notch filters and get some information about responses about notch filter applications in ECG. (Figure 2.1.1 and Figure 2.1.2)

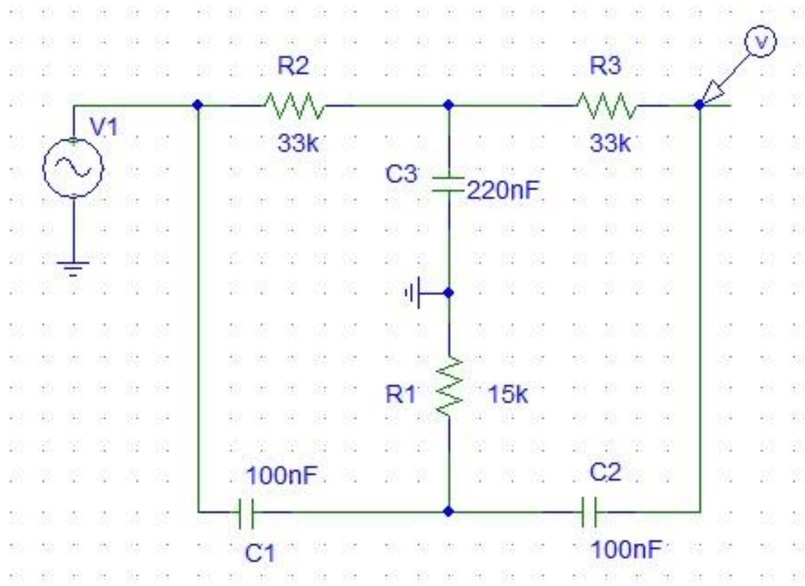


(Figure 2.1.1)



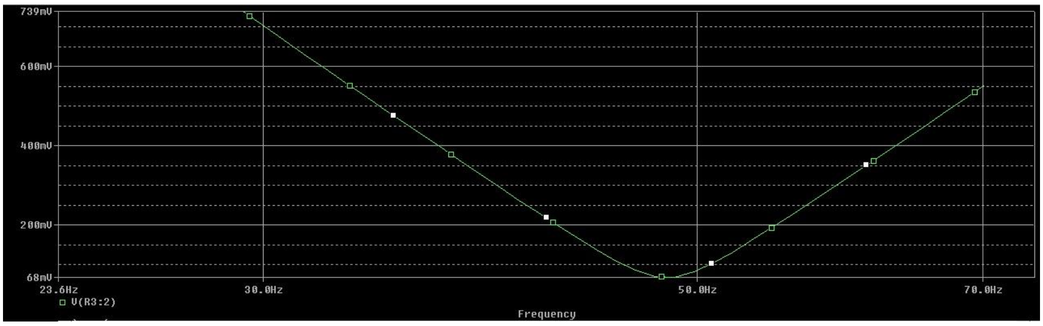
(Figure 2.1.2)

With using these informations we designed our notch filter.(Figure 2.1.*3)*



(Figure 2.1.3)

We calculated our cutoff frequency with using equation 1/4*pi*R*C equation and setted our cutoff frequency to 50Hz. Then we have simulated our signal in Pspice to observe output response.(Figure 2.1.4)



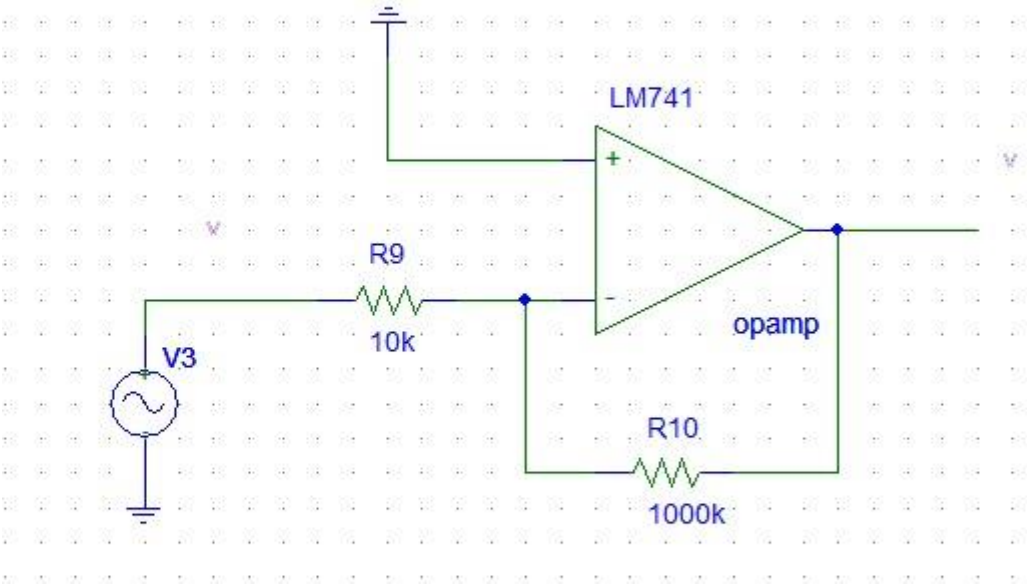
(Figure 2.1.4)

After Pspice simulation we determined our signal is filtered clearly from 50Hz signal and we built our circuit on pertinaks circuit board.

2.2.1 Inverting Amplifier

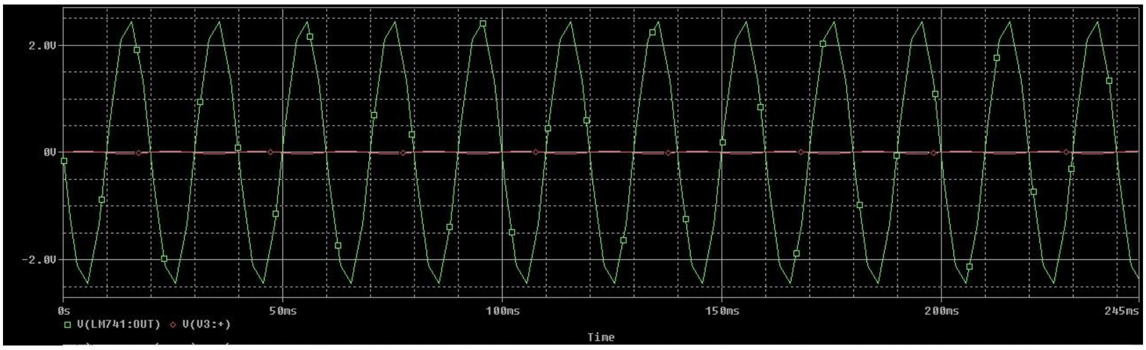
After filtering steps due to voltage loss because of filtering processes we decided to amplify our signal to process more efficiently in digital imaging process such as Arduiuno and MATLAB.

In this amplifier circuit we have used LM741 op amp. We designed our circuit with 100 gain as can be observed in figure 2.2.1.



(figure 2.2.1)

After designing our amplifier circuit we have simulated it with Pspice simulation and tried to observe what kind of response that we are seeing in the output. Response can be seen in figure 2.2.2.



(figure 2.2.2)

Explanation about figure 2.2.2: In this circuit our because our amplification rate is too much (100 Gain) our sinusoidal input is not can be observed clearly in figure 2.2.2

2.2.2 LM 741 Opamp

Information

The LM741 is a general purpose amplifier that can be used in a variety of applications and configurations. In this project, we decided to use it because it is the most suitable op-amp set to 100 gain and this op-amp support between 0.5Hz and 50 Hz frequencies.

Desing Requirement

The gain of the system is determined by the feedback resistor and input resistor connected to the inverting input. We calculated gain with Equation1.;

Equation1 = G = 1 +(R2/R1)

Typical Application

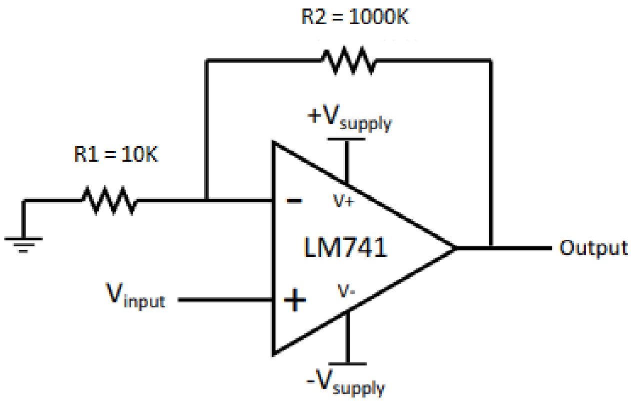
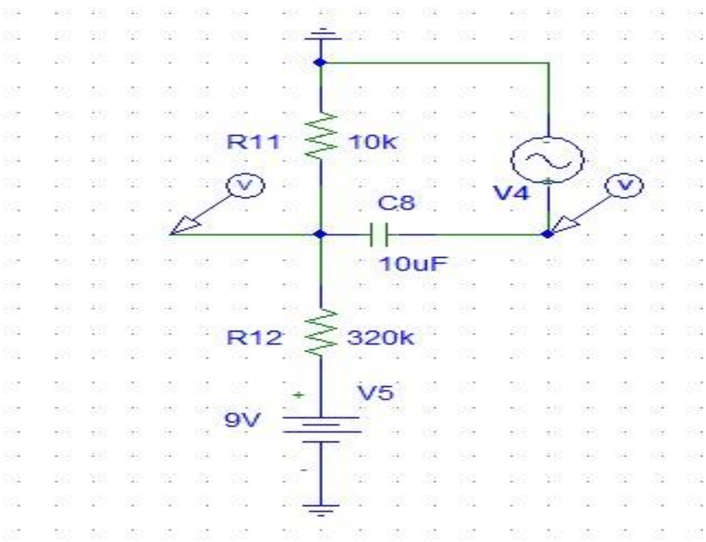


Figure 1. LM741 Noninverting Amplifier Circuit

The gain is set to ≅ 100 with this amplifier.

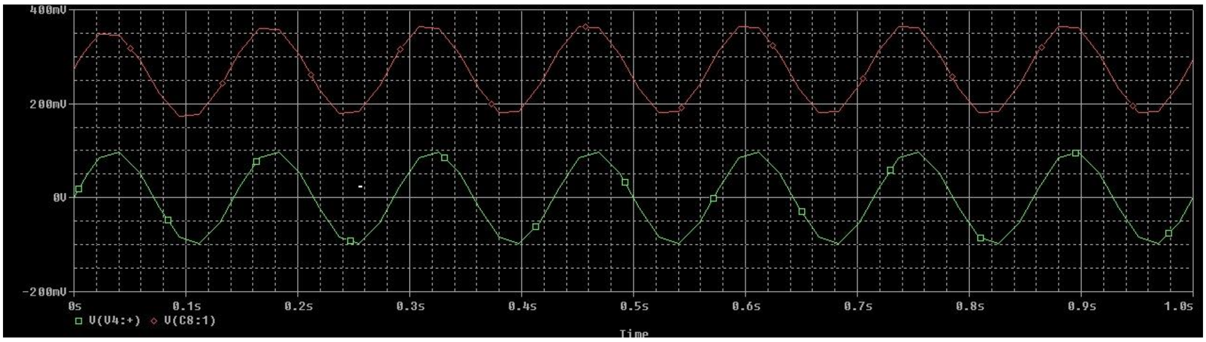
2.3.1 Offset Circuit

After these analog steps, we have observed that our signal has a voltage values below 0. In this project we are using Arduino to transform our analog signal to digital signal and Arduino is not able to process voltage values below 0 Volts. To resolve this problem we decided to use an offset circuit which will carry our signals negative values to positive side. First we observed our signal's minimum value to determine offset ratio. We checked our amplifier circuit's output with oscilloscope and we determined that our minimum value is approximately 300mV. With using this information we designed an offset circuit which will provide 300mV offset to our signal. This circuit is basically based on the voltage divider circuit as can be seen in figure 2.3.1.



(Figure 2.3.1)

Then we have simulated it with Pspice to see offset circuit's response. (Figure 2.3.2)



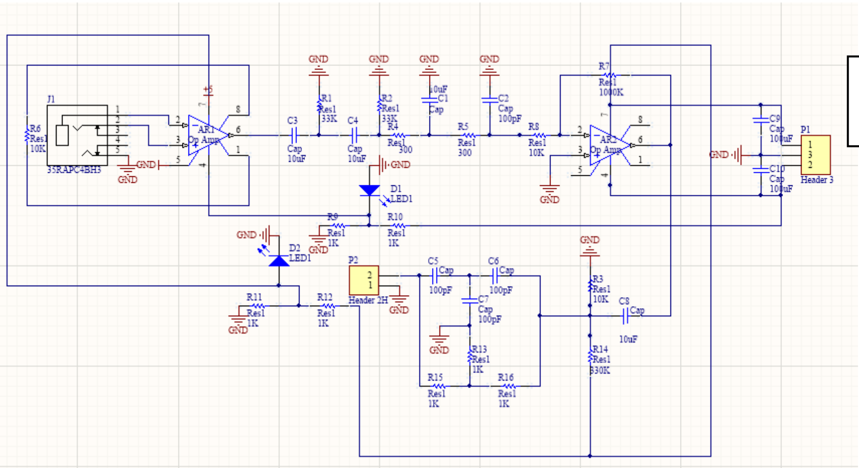
As can be observed from figure 2.3.2 we received 300mV offset and we have built our offset circuit in pertinaks circuit board.

3. ADC Method

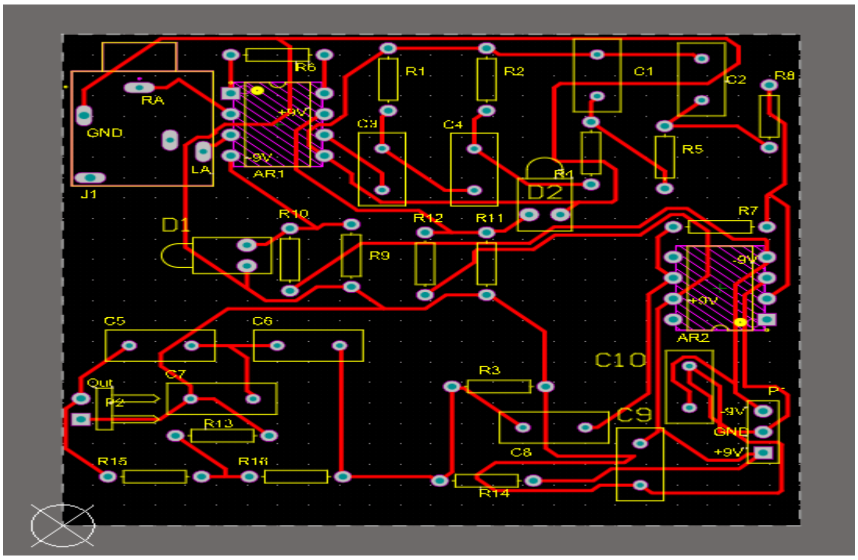
We received the analog signal from port A0 of the Arduino. With the 9600 sampling frequency, Arduino transformed the analog signal into a 32-bit digital signal. We sent this signal to MatLab using the UART communication way, and there we built different operations.

4. PCB, schematic of the total circuit and the components that used in the circuit

4.1 PCB and Circuit schematic



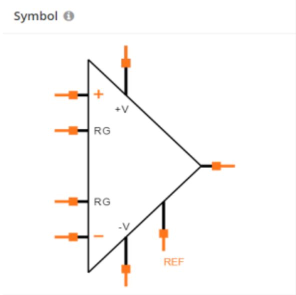
(Figure 4.1.1)

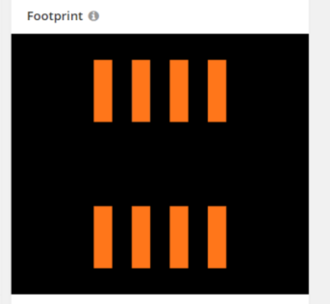


(Figure 4.1.2)

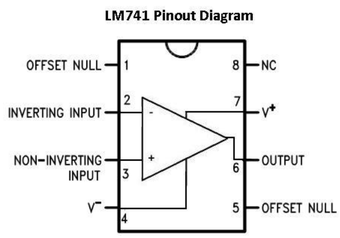
4.2 Components

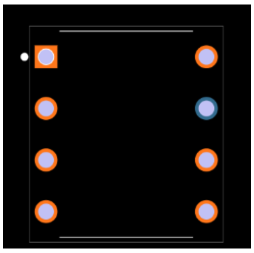
1) AD620





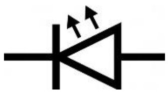
2) LM 741

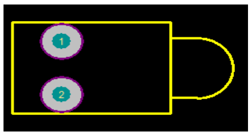




3) LED (Red)

LED (Green





4) Resistor

4 x 33K Ohm

2x 0.3K Ohm

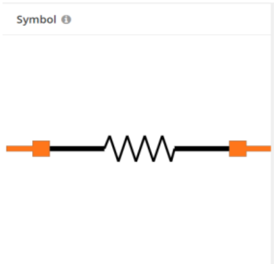
4x 1K Ohm

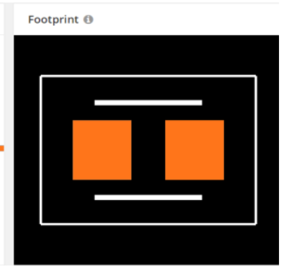
3x 10K Ohm

1x 330K Ohm

1x 15 K Ohm

1 x 1000K Ohm





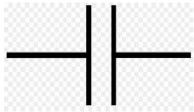
5) Capacitor

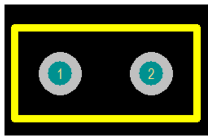
2x 100 uF

5 x 10 uF

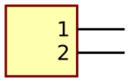
1 x 220 uF

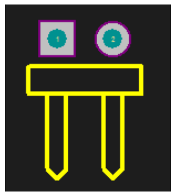
2 x 100 nF



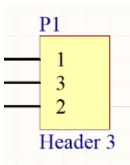


7) Header 2H

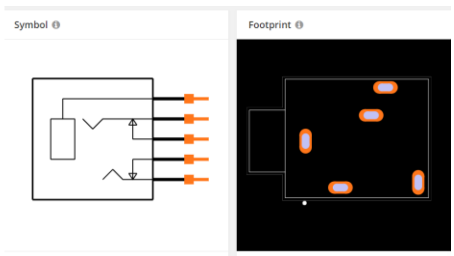




8) Header 3H

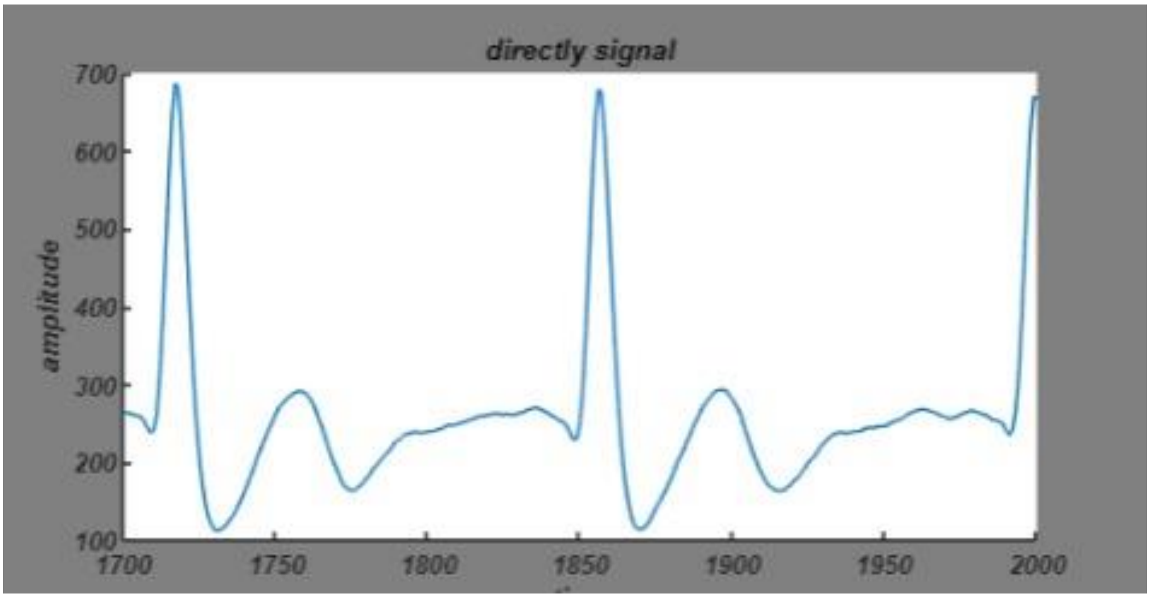


9) Jack (35RAPC4BH3)



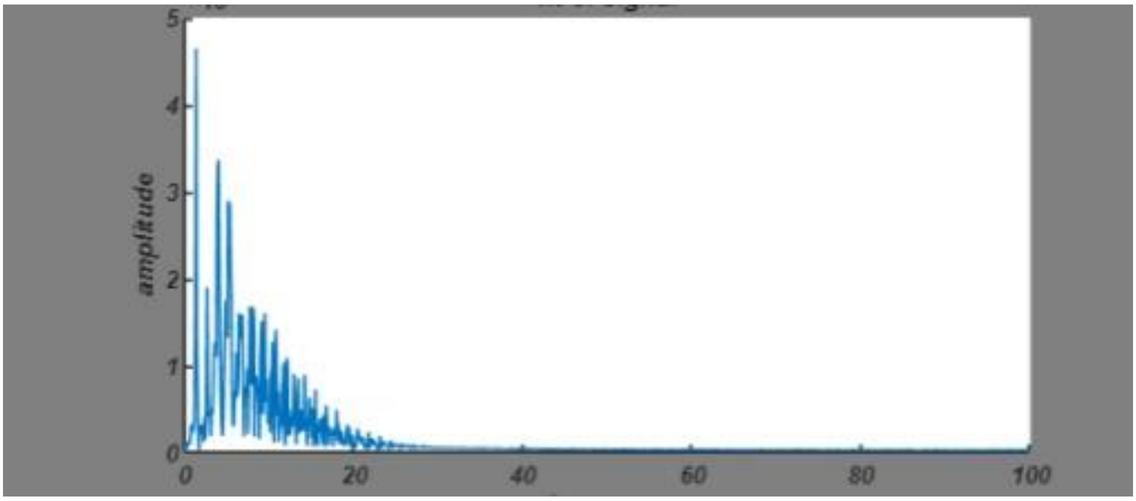
We have designed our PCB on Altium designer as can be seen in figure 4.1.2 but due to troubles about PCB printing process we built our circuit on pertinaks cardboard.

5. Digital Proccesing



(Figure 5.1.1)

In signal processing we used matlab which takes data from arduino and they both have serial communication in order to take data to matlab. After taking data to matlab we decided to use app designer to execute real time signal processing. In this whole signal processing time we utilized from arrays features. The last 300 sample was plotted in our plotting screen but we took 20 sample in every loop that means one loop before my 300th value is equal to 269th value right now so that we obtain a data flowing which is rightside to left side.



(Figure 5.1.2)

Almost the same process is working when fft is plotted. Every 30 value is plotted when one loop is completed. We used fast fourier transformation which is named on matlab like a fft that will take directly fft of our signal but before this step we should delete our direct current value so the array which is storing the mean value of the signal was extracted from ourfundamentel signal. After this step fft can be applied but we have to use abs function in order to take signal which is below zero, to upper side in order to eliminate negative frequency this speacility comes from cosinus theorem.

5.1.1 About fft function;

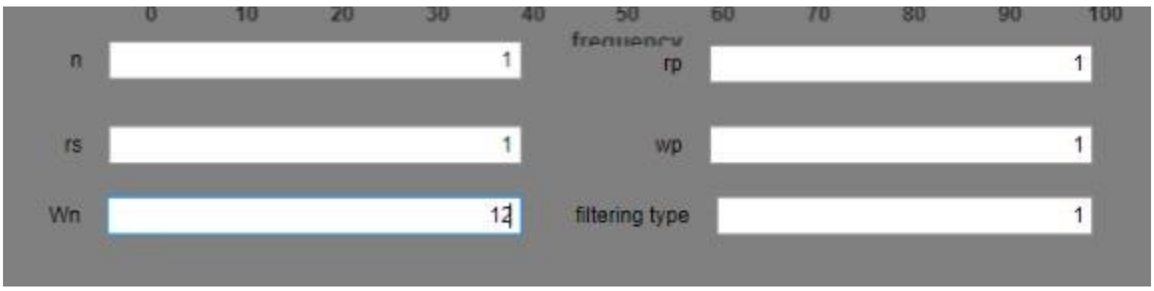
F*ast Fourier Transform (FFT)*

Earlier the method used for ECG signal analysis was time domain method. But the limitation was that it was not sufficient to study all characteristics of ECG signal [6] [9]. So a new method FFT was developed. Fourier transform is a well known method which transforms time domain signal to frequency domain to obtain the frequency coefficients [10]. FFT is an elementary transform in digital signal processing and has various applications in frequency analysis, signal processing etc [11]. It is a fast and more capable algorithm to work out the Discrete Fourier Transform (DFT) and obtains the same effect [12]. FFT is defined by the formula shown in *equation (1):*

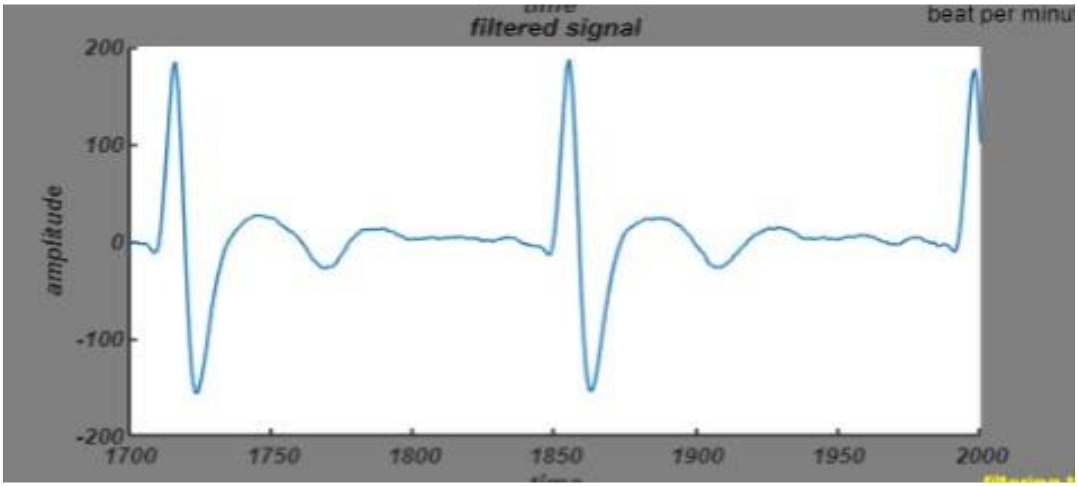
$$Xk = \sum_{n=0}^{N-1} x e^{-nk2\pi i/n}$$

(Figure 5.1.3)

One of the task is the counting of heart rate for per minute. We used the findeaks function in matlab. Firstly peaks were found and max value is also was found. This process provide the max peak value of our signal. After all there is a multipilication operation in order to count peaks which are needed. At this position we faced with a problem that cause of 20 value cause or frequency sampling rate is 200 that means we will can take a peak every 200 sample for normally people. So we applied a different algorithm to solve that. Just it will count the beat rate in every 2 seconds to show purer beat rate and after 10 seconds it will give real values cause our main ecg vector occurs 2000 sampling that means after 10 second it gives last 10 seconds rate, after 2 seconds it will show again last 10 seconds heart beat rate. Finally the counter paired with the edit numeric field and it shows what we have .



(Figure 5.1.4)



(Figure 5.1.5)

Matlab can use also digital filtering part there are a lot of filtering type for that. We used butter, ellip and cheby in order to see better signal. Every part of filtering part includes their own value which is regulate from the users. We used edit field which is numeric on matlab app. And we replace the variable name with the inside of the value in our filtering system. For example butter(app.n,app.wn,(app.bandType))

5.2.1 Why did we use those filters type?

Elliptic filters offer steeper rolloff characteristics than Butterworth or Chebyshev filters, but are equiripple in both the passband and the stopband. In general, elliptic filters meet given performance specifications with the lowest order of any filter type.

`ellip` uses a five-step algorithm:

1. It finds the lowpass analog prototype poles, zeros, and gain using the function `ellipap`.
2. It converts the poles, zeros, and gain into state-space form.
3. If required, it uses a state-space transformation to convert the lowpass filter to a bandpass, highpass, or bandstop filter with the desired frequency constraints.
4. For digital filter design, it uses `bilinear` to convert the analog filter into a digital filter through a bilinear transformation with frequency prewarping. Careful frequency adjustment enables the analog filters and the digital filters to have the same frequency response magnitude at `Wp` or `w1` and `w2`.
5. It converts the state-space filter back to transfer function or zero-pole-gain form, as required.

Butterworth filters have a magnitude response that is maximally flat in the passband and monotonic overall. This smoothness comes at the price of decreased rolloff steepness. Elliptic and Chebyshev filters generally provide steeper rolloff for a given filter order.

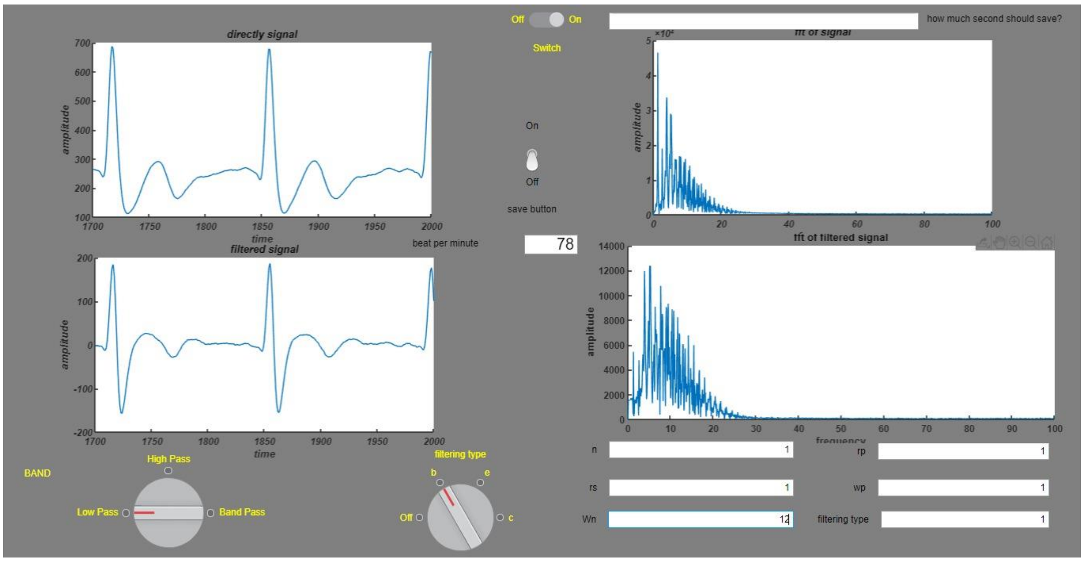
`butter` uses a five-step algorithm:

1. It finds the lowpass analog prototype poles, zeros, and gain using the function buttap.
2. It converts the poles, zeros, and gain into state-space form.
3. If required, it uses a state-space transformation to convert the lowpass filter into a bandpass, highpass, or bandstop filter with the desired frequency constraints.
4. For digital filter design, it uses bilinear to convert the analog filter into a digital filter through a bilinear transformation with frequency prewarping. Careful frequency adjustment enables the analog filters and the digital filters to have the same frequency response magnitude at Wn or at w1 and w2.
5. It converts the state-space filter back to its transfer function or zero-pole-gain form, as required.

Elliptic filters offer steeper rolloff characteristics than Butterworth or Chebyshev filters, but are equiripple in both the passband and the stopband. In general, elliptic filters meet given performance specifications with the lowest order of any filter type.

ellip uses a five-step algorithm:

1. It finds the lowpass analog prototype poles, zeros, and gain using the function ellipap.
2. It converts the poles, zeros, and gain into state-space form.
3. If required, it uses a state-space transformation to convert the lowpass filter to a bandpass, highpass, or bandstop filter with the desired frequency constraints.
4. For digital filter design, it uses bilinear to convert the analog filter into a digital filter through a bilinear transformation with frequency prewarping. Careful frequency adjustment enables the analog filters and the digital filters to have the same frequency response magnitude at Wp or w1 and w2.
5. It converts the state-space filter back to transfer function or zero-pole-gain form, as required.



(Figure 5.1.6)

All this process connected with a open and off switch which is usable by user. There are one more thing in this all gui design. If user want to save the data user can do that in prominent time interval. And program makes this signal approprite for processing. There is one more features in whole system. It would be a surprise for all of us if we can do that.

6.Cardbox and Portable System of ECG Device

Since the wiring of our circuit is vu, we covered it with a coating to protect it from any short circuit. For this circuit to be a portable device, we designed this coating as a portable box. To ensure portability, the power supply had to be portable. So we chose the batteries as the power source. Since the feeds of the opamps we used are -9V, + 9V, -4.5V and + 4.5V, we put two 9V batteries in our circuit and designed the voltage divider for the requirements of -4.5V and + 4.5V.

7.Conclusion

This project was a good start for us to learn about biomedical engineering applications. It also enabled us to know more about the ECG and how it works. We have learned that the ECG (electrocardiogram) tracks the remainder of your heart's electrical activity which provides information on the heart rate and rhythm and indicates if the heart is swollen due to high blood pressure (hypertension) or signs of a previous heart attack (myocardial infarction).

In the first part of the project, we started by just using the AD8232 to acquire the ECG signal directly without constructing any circuit and to process the acquired signal on MATLAB right away using the Arduino serial communication. But for the second part of the project, we had to construct the circuit from scratch using the tools we have in the lab. The most critical part was doing a lot of research about the ECG and the circuit we should use. After some brainstorming, we came up with the circuit we had constructed using AD620 with (highpass low pass whaterver you added to the circuit write it here). After several attempts and using the error and trial method to solve the problems we faced, we were able to get the signal successfully with the least amount of noise. As tough as it may sound but we have fun struggling through this process. Also, we were able to learn about printing circuits using Altium designer and how to design a PCB. Moving on to the digital part, we were able to use Matlab's built-in app designer to design an app to make it easier for the users to take control over the app even they are not experienced in Matlab. We learned how to implement different kinds of filters with different orders and different cutoff frequencies and process the data in real-time. The most intriguing part about this project that it could be an actual product that could be used in the real world and it could solve real-world problems. It possible to work more to take it up to the level which could be used in the medical segment.

8.Reference List

https://www.researchgate.net/figure/Block-diagram-of-3-lead-ECG-machine_fig1_308856444

https://www.slideshare.net/AbdiqaniSidow/ekg-project-oral-presentation

https://www.researchgate.net/publication/272863992_Electrocardiogram_Signal_Analysis_-_An_Overview

https://www.jove.com/science-education/10473/acquisition-and-analysis-of-an-ecg-electrocardiography-signal

http://www.ti.com/lit/ds/symlink/lm741.pdf

https://www.analog.com/en/analog-dialogue/articles/ecg-front-end-design-simplified.html