ANIMAL HEART RATE MONITORING

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Bachelor’s thesis

Electrical and Automation Engineering

Spring 2019

Trung Van

LIST OF FIGURES

LIST OF EQUATIONS

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Electrical and Automation Engineering
Valkeakoski

Author Trung Van **Year** 2019

Subject Animal heart rate monitor equipment

Supervisors Juhani Henttonen and Raine Lehto

ABSTRACT

Most of heart diseases and health conditions are reflected by the heartbeat. Especially in the case of surgery, tracking the heart signal is vital to diagnosing the general situation and for making precise predictions in the surgery process. The topic of the thesis derived from a real situation when it is needed to track the heart signal of anaesthetized animal in veterinary facilities. The purpose of this thesis was to create a practical solution which would allow us to monitor the heart rate of an animal.

This design included studying circulatory systems, popular methods to archive the heart rate and relative knowledge of electrical automation. The thesis can be divided into two mains parts: hardware and software. The hardware part includes a low-cost stethoscope chest piece connected to a microphone, an amplifier-filter circuit for the heart signal and an Arduino to send data to a webpage. The software was Arduino program to process the input signal from circuit and a website-form to display the heart signal along with other information.

Keywords Heart rate monitor, signal processing, digital stethoscope, lowpass active filter, electrical circuit, data visualizing, web interface.

Pages 50 pages including appendices 0 pages

Contents

1 INTRODUCTION

1.1 Outline of thesis

There are six main chapters in this thesis as follows.

Chapter one is about the background of the topic, the author's target on this project.

Chapter two covers the basic knowledge about cardiovascular issues and different methods to track the heart signal which played a huge role in making technology selection later.

Chapter three reviews the electronics and physics theory required to work on the project. Chapter four discusses automation and programming skills applied in the thesis.

Chapter five deep dives into real work on thesis including: selecting suitable method, design system, designing electrical schematic and the programs. Some preliminary results of the work are introduced side by side the implementation enhance perspicuity.

Chapter six describes the results of data visualization and heart beat calculation on the website user interface.

Chapter seven discusses present limitations of the project as well as recommendation future work which can be done to improve the efficiency and practicality of the project. Chapter eight is an overview of the project after the development process.

1.2 Background of project

The target objects of the thesis were small to medium-sized animals on operation table. They are anaesthetized and located on a soft surface hence there is no significant movement during the surgery operation. An example for the operation was illustrated in Figure 1. As these animals stay still, it is hard to observe their regular life signal. The heart rate becomes vital parameter in this case for the vet to follow operation status.

Figure 1 Animal on operation table

There is a lot of high-tech equipment in veterinary hospital, but it is quite expensive and sometimes rather excessively functional. Professional sensors may cost €1000 which may be too high investment. There were attempts looking for alternative solution from devices for human. Even market of human medical equipment is large and diverse still it is challenging to find an applicable equipment for animal. Therefore, we need a simple and reasonable-cost solution which can afford basic requirement specific facility. Design process asked collaboration between engineer and end user also brings us ability to update or maintain in detailed situation.

Figure 2 Schiller resting ECG machine ($3600) (USA medical and surgical supplies)(n.d)

As there is no straightforward map for solution development, the topic was expected to start from research about the heart biological mechanism where the signal was generated. The next step was investigating different techniques to measure the heart rate. After that the electrical hardware and software for a protype both had to be designed and had to double checked in simulation environment to optimize efficiency. In the last stage, a test model was deployed.

1.3 Project targets

The targets of the thesis project included:

- Investigating suitable technique to monitor the heart rate
- Creating a design for the heart rate monitoring device to meet the following functions and requirements:
    - Collect the heart rate signal
    - Filter noise from the original input signal and amplify filtered signal
    - Convert the heart rate digital signal into audio sound
    - Process signal to calculate the heart rate
    - Display EKG(Electrocardiography) and the heart rate on screen
- Deploying a web application to monitor the heart rate and a test model for the device.

2 THEORETICAL BASIS OF BIOMEDICAL ENGINEERING

2.1 Cardiovascular system

This thesis focuses on small to medium sized animal for example dogs and cats. A dog's circulatory system is quite similar to the circulatory system of humans so some reference of human can be applied on these animals.

2.1.1 Cardiovascular system

The cardiovascular system includes blood vessels(veins and arteries) and the heart. For a dog, the heart sits in the chest cavity between 3rd and 6th ribs as can be seen in Figure 3. This may vary depends on breeds, species, age, etc.

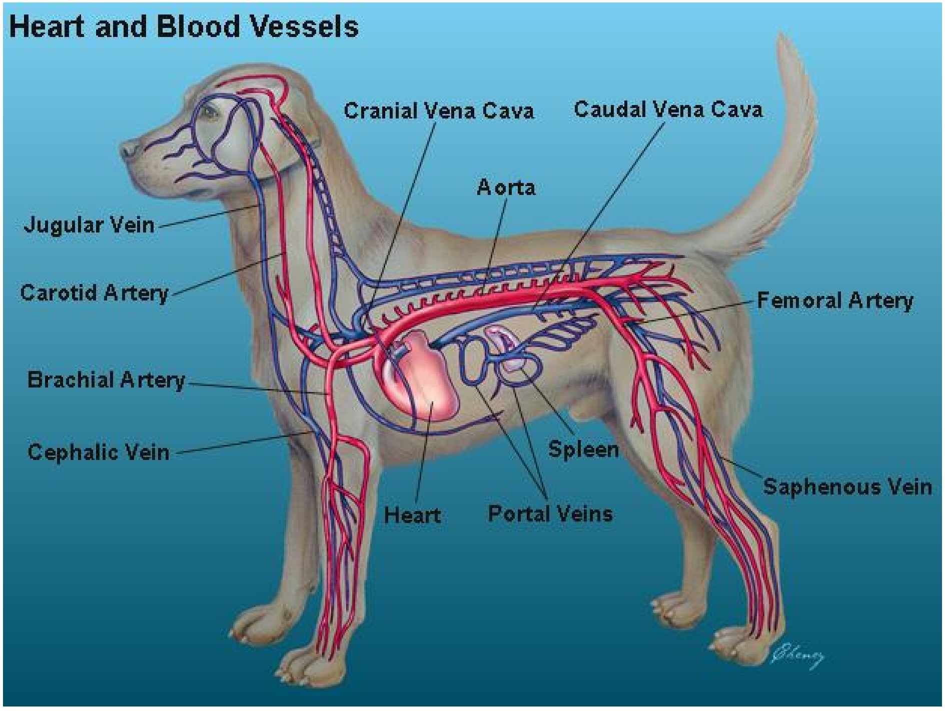


Figure 3 Dog cardiovascular system (Dog cardiovascular, n.d)

2.1.2 Anatomy of heart

The heart illustrated in Figure 4 is one of the most important internal organs of an animal. Considered as the central part of the circulatory system, its function is to pump blood and distribute oxygen to other part of body. The right side of the heart pumps blood to the lungs so to gain oxygen into blood as well as remove carbon dioxide from blood cell. On the other hand, the left side of the heart pumps blood to the rest of tissues along with oxygen and nutrients. The blood cell then leaves bring waste with them to other organs(Washington State University, n.d). Accordingly, even a small dysfunction of heart may lead to huge effects in the body.

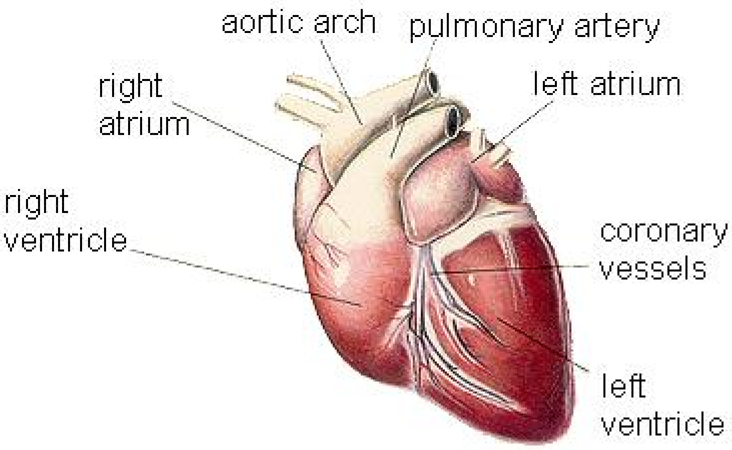


Figure 4 Outside view of heart(Dog cardiovascular, n.d)

Heart is a hollow muscular organ .The normal heart has four chambers; the two uppers are left and right atria. The rest are two lower champers(left and right ventricles) which are stronger. The right atrium and right ventricle make up the “right heart” so does the “left heart” is form left atrium and left ventricle which can be seen in Figure 5. There is a muscle wall between two sides of the heart called septum. Blood flows in one direction through the heart thanks to series of valves. Between the atria and the ventricles there are atrioventricular valves. Between the heart and the aorta and between the heart and the pulmonary artery are semilunar valves. On the left side, inlet valve is mitral valve and outlet valve is aortic valve, both connected to ventricle. On the right side, inlet valve and outlet valve are tricuspid valve and pulmonary valve respectively.(Cunningham,2018).

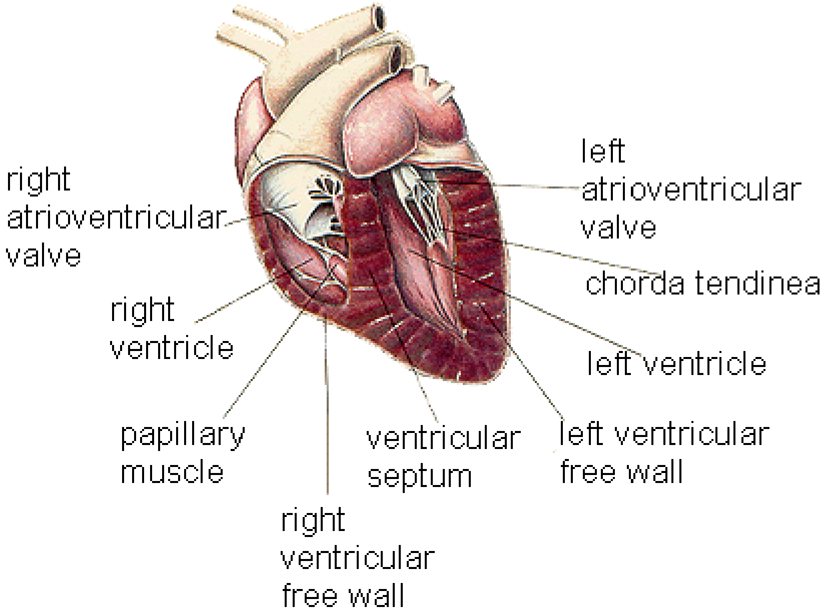

right atrioventricular valve
left atrioventricular valve
chorda tendinea
right ventricle
left ventricle
papillary muscle
ventricular septum
left ventricular free wall
right ventricular free wall

Figure 5 Inside view of heart(Dog cardiovascular, n.d)

2.1.3 Heart beat

Each heart beat consists of two parts or one pumping actions: diastole and systole which are shown in Figure 6.

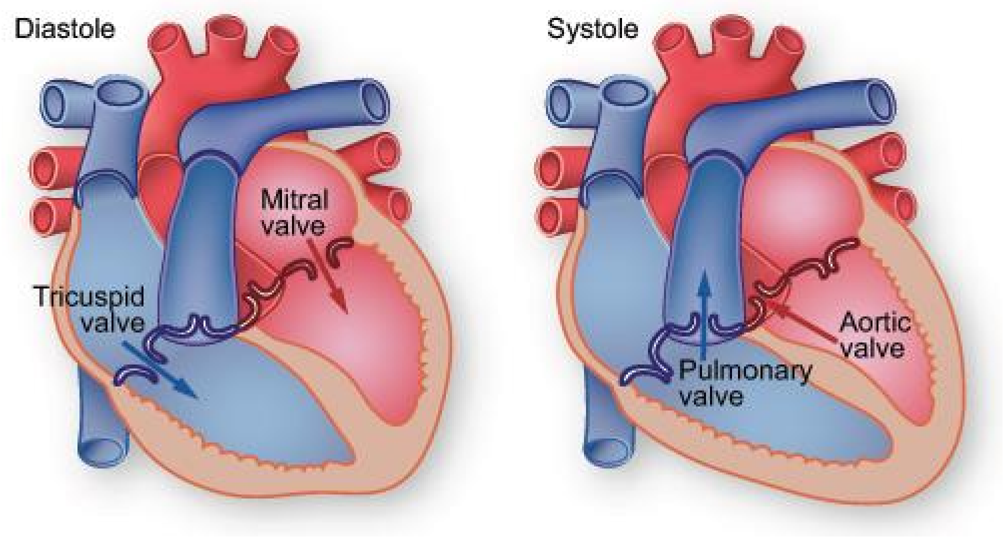


Figure 6 Heart beat illustration(Dog cardiovascular, n.d)

The first sound of the heart beat is generated when the mitral and tricuspid valves are closing. During diastole phase, the ventricles(resting lower chamber right and left) are at relax situation and filled with blood. The second half comes when aortic and pulmonary valves closing. At the start of this phase, the ventricles are full of blood, ventricles contract and pump blood out. The tricuspid and mitral valves keep shut tight, prevent blood backflowing. After that, blood moves into the aorta and pulmonary artery then the ventricles relax, the pulmonary and aortic valves close. The cycle starts over again as there is low pressure in the ventricles, tricuspid and mitral valves open. (Abbas,2009).

Heart beating happens due to a microscopic electrical current derives from heart’s pacemaker called the sinoatrial node. Muscle fibers in the heart contracts as a result when rhythmic electrical impulses or discharges. The larger an animal are, the lower the heart rate it gets. For example: 30 times/ min in the horse, over 120 times/min in the cat.

For dog, usually, the heart rate is irregular as it increases and decreases while breathing in or out. Changing in the heart rate corresponding with breathing can be considered as good sign of health. (Abbas,2009).

2.1.4 Heart sound and murmurs

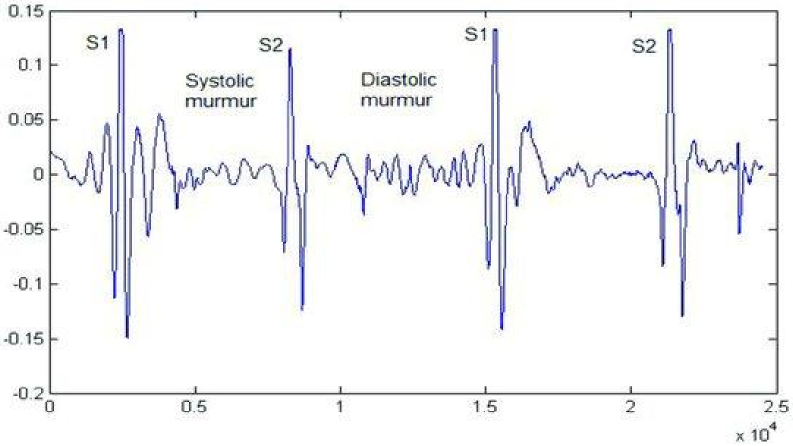


Figure 7 Graphing heart sound and heart murmur (Amiri 2013)

Heart sound is generated by sudden acceleration or deceleration of blood and the heart vibration because of blood circulation. With a stethoscope, we are already able to distinguish 2 heart sounds(thuds) known as S1 and S2 which repeat one after other during a beat. Systole phase is the time between S1 and S2 while between S2 and S1 diastole phase. Heart murmur originates from turbulent blood flow or heart structures vibrations. They can be heard from the heart or major blood vessels.(Ahlstrom ,Jan2018).

2.2 **Different methods to measure the heart rate**

Various methods are being used nowadays to measure the heart rate: manual(pulse palpation), electrocardiography, oscillometry, Photoplethysmography and phonocardiography. Each of them detects different phenomena occurring in the human body in a cycle of heart beat.

2.2.1 Manual methods:

Pulse palpation:
This is a physical examination. Traditionally, the pulse rate is achieved by counting beats during a period (varies from 15-60 seconds) then multiply result with coincide number to get the beat per minute(BPM). Until now, this is still a popular method used by healthcare professionals and doctors in daily routine. At any point of the body where there is an artery, it’s possible to measure the heart rate using this method. Radial, carotid, branchial and femoral arteries are some of the most common positions (Walker,1990).

Using stethoscope:
A stethoscope is an acoustic medical device for auscultation or listen to internal sounds of human or animal body. It is often used to listen to heart sound, breathing sound, intestines or blood flow in arteries and veins.

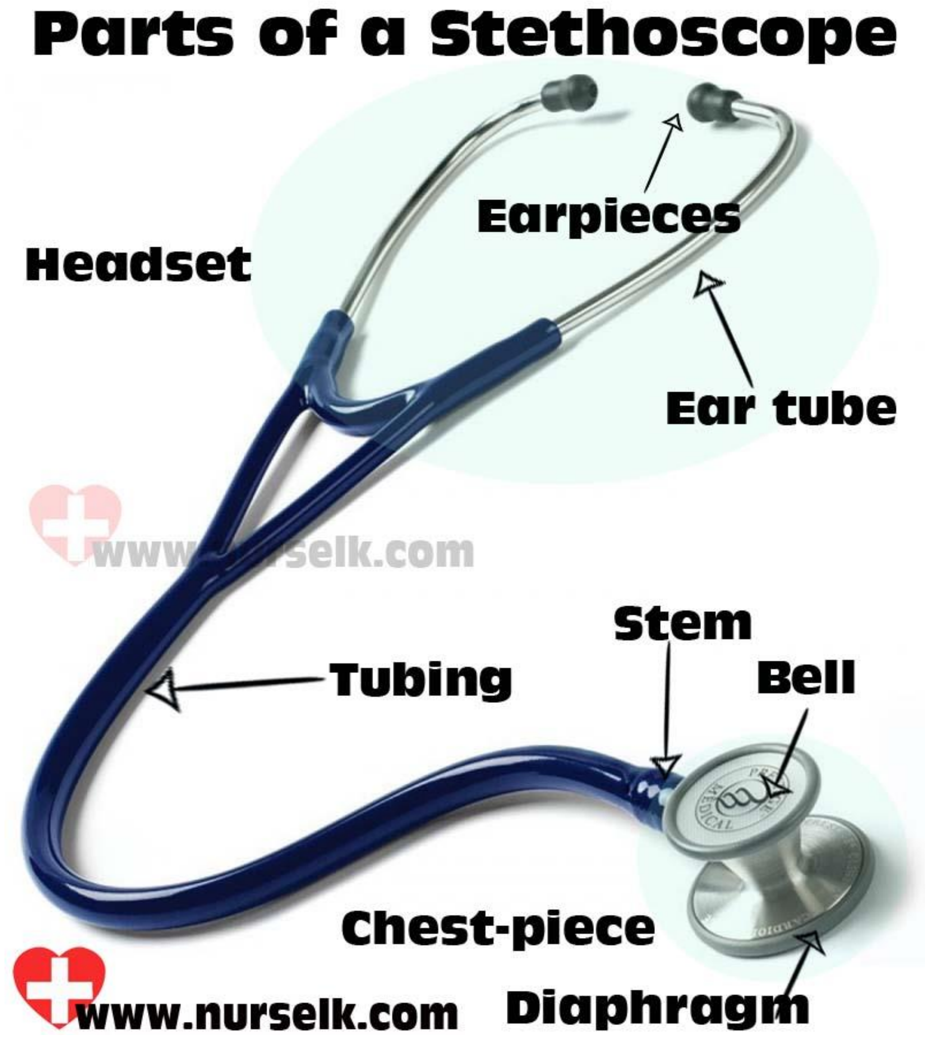


Figure 8 Stethoscope parts design (Stethoscope, n.d)

A stethoscope illustrated in Figure 8 has the following components:

- The chest-piece or head of the stethoscope is composed of the connected stem, diaphragm and bell. The diaphragm is the large circular end of the chest-piece which is used to collect sound from contacted surface. While using stethoscope, diaphragm should be held firmly against the skin. The bell is the smaller circular of the chest-piece. The bell is designed as physical amplifier for low frequency sounds when applied with low pressure.
- The stem is basically the metal/steel part of the stethoscope that connects the stethoscopes tubing to the chest-piece.
- Headset is combined by components from upper half of the stethoscope that includes ear tubes, tension spring and ear tips. They are designed to bring comfort in usage and maximize the sound quality.
- Tubing is the soft flexible line whose purpose is to transfer the sound captured by the diaphragm to the ear tube at the lowest distortion.

- Ear tubes are the metal tubes of the stethoscope connecting ear tips and tubing. They are designed to isolate the sound and transfer it with the small quality loss.
- The part that contacts with user ears is eartip. This part is usually made from rubber or silicone material to effectively fit inside user ears. (Nelson,2015).

Cardiac auscultation should be implemented in a quiet room. The heart beat is best heard on the left side of the dog, behind the point of the elbow. If it is hard to catch the sound, move the stethoscope to different areas without lifting the chest piece off the body. Apply mild pressure on the pressure on the chest piece. Since it’s possible to collect heat sound from the stethoscope, use the same calculation as pulse palpation technique to achieve the heart rate.

2.2.2 Electrocardiography(ECG)

Electrocardiographic recording is the most precise method of the heart rate measurement and is routinely carried out in many clinical settings, especially in critical care cases. This invasive method is more suitable to track the heart rate during long period. Figure 9 illustrated an overview of ECG machine.

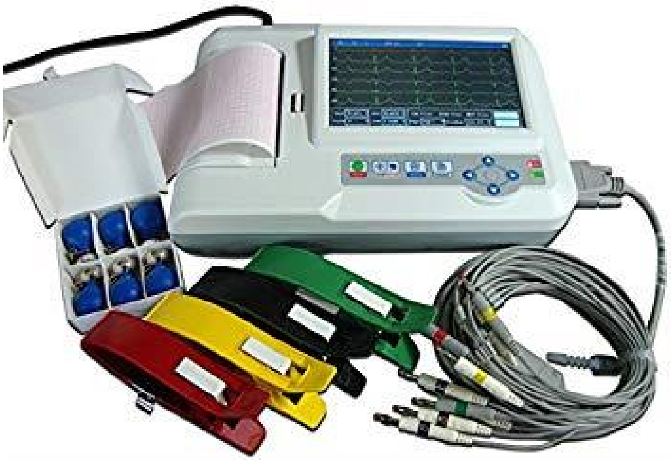


Figure 9 ECG machine (ECG n.d)

During the cycle of contractions and relaxation of cardiac muscle, some tissues on the heart call pacemaker generates electrical impulses which spread through all heart. It is possible to detect these impulses. However, the amplitude of the electrical wave is quite small, ECG electrodes must be placed on some specific position on the body to collect these signals. Electrodes must be attached on target skin, sometimes they require special jelly. Current picked up by electrodes goes through amplifier and display on paper/ screen(for modern type) or storage devices.(Kumar,n.d).

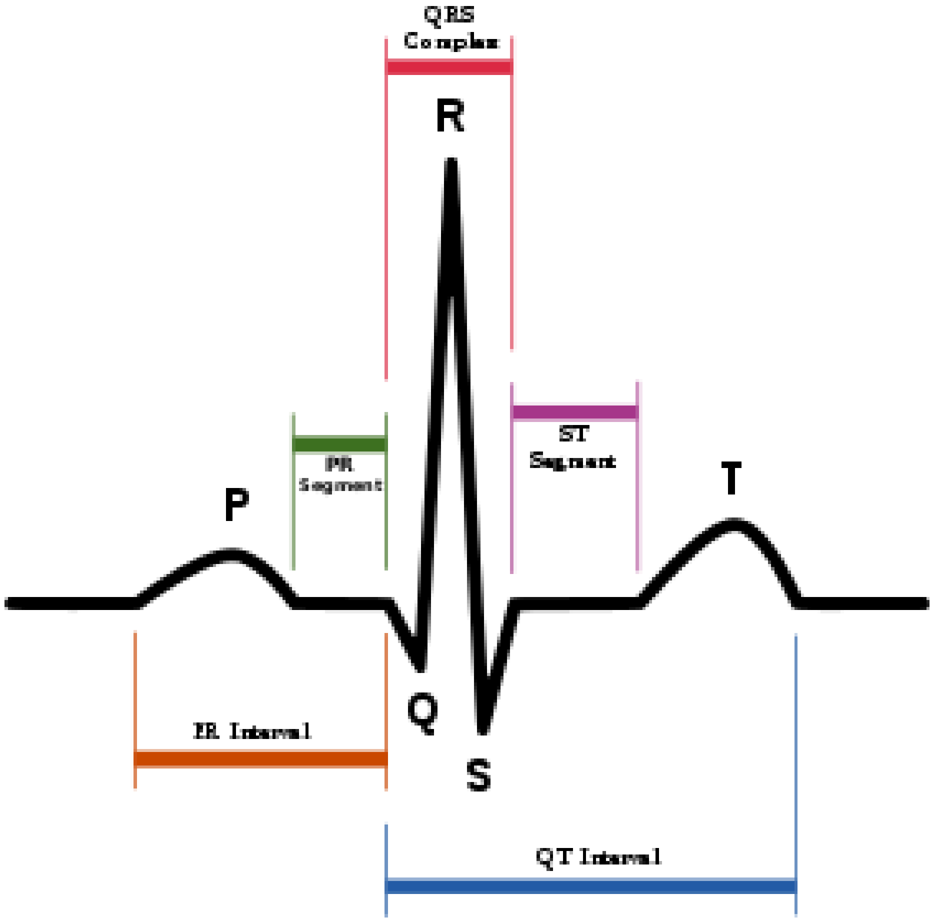


Figure 10 ECG of a heart in normal sinus rhythm (ECG signal, n.d)

An ECG signal is characterized by five main peaks and valleys marked as: P, Q, R, S, T as can be seen from Figure 10. The peak P is the result of muscle contraction of the atria. At the end of the atrial contraction and the start of ventricular contraction, the R peak shows up with the magnitude of 0.1mV to 1.5mV. The end of a ventricular contraction is marked by the T-peak.

The heart rate is achieved by calculating the time interval between two consecutive R peaks, denoted RR interval. Taking the average of this value over fixed window and then scale it to units to beats per minute (BPM) we can get the desired result. The general working principle of ECG was described in Figure 11.

There are several factors can interrupt or interfere the measurement: electrical interference from other equipment, measurement noise, electromyogram noise, movement of target, baseline drift, instrument noise, …
To calculate the heart rate, there must be processing in both analog and digital domains. A microcontroller can handle these functions in “real time”. (Kumar,n.d).

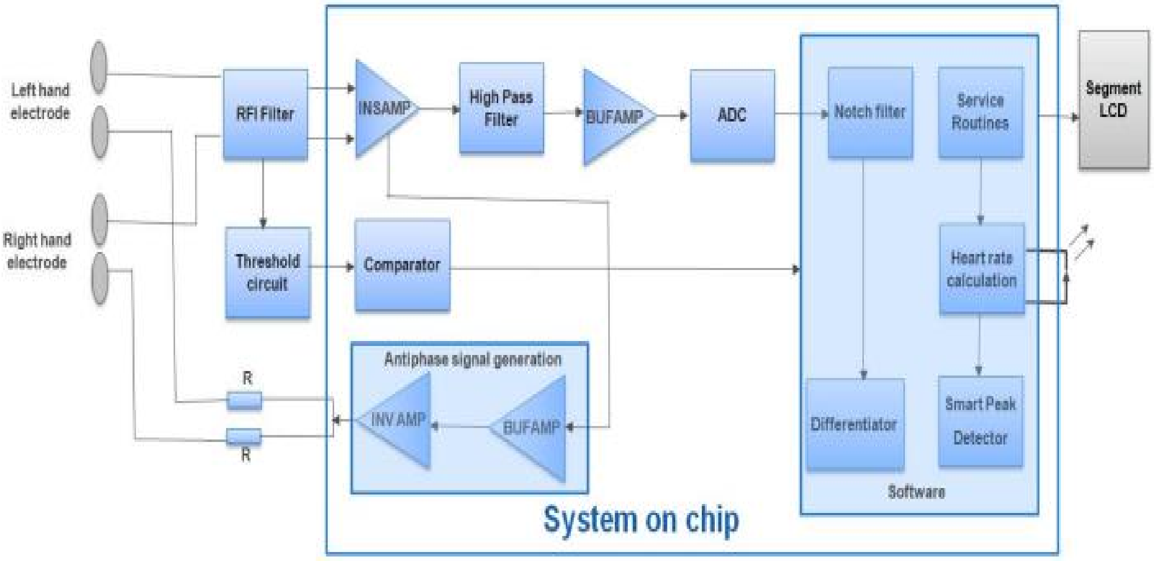


Figure 11 Typical ECG system (Measurement system, n.d)

2.2.3 Photoplethysmography

Photoplethysmography(PPG) is an optical technique used to detect blood volume changes in the microvascular bed of tissue. This method offers a non-invasively measurement at the skin surface as shown in Figure 12. A basic form of PPG system includes several opto-electronic components(light source to illuminate tissues + photodetector to determine the variation in light intensity correspond to the perfusion from the blood volume change). The light diffuses through human tissues and gets detected by photodetector placed on the opposite side of the light emitter. Usually, the chosen light is at red or a near infrared wave length. Peripheral pulse is the most recognized waveform as it is synchronized to heart beat. (Kumar,n.d).

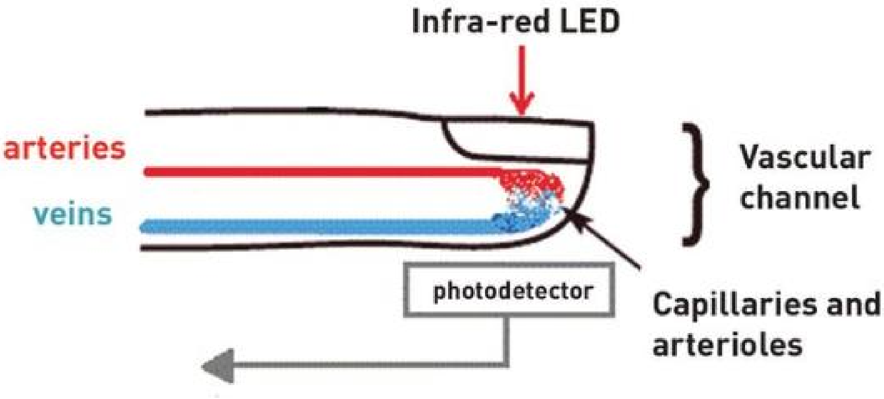


Figure 12 Photoplethysmograph sensing diagram

Signal achieved by sensing the light then filtered, amplified and processed through an analog to digital converter .The heart rate is calculated using a peak detector peak detector. The basic system was shown in Figure 13.

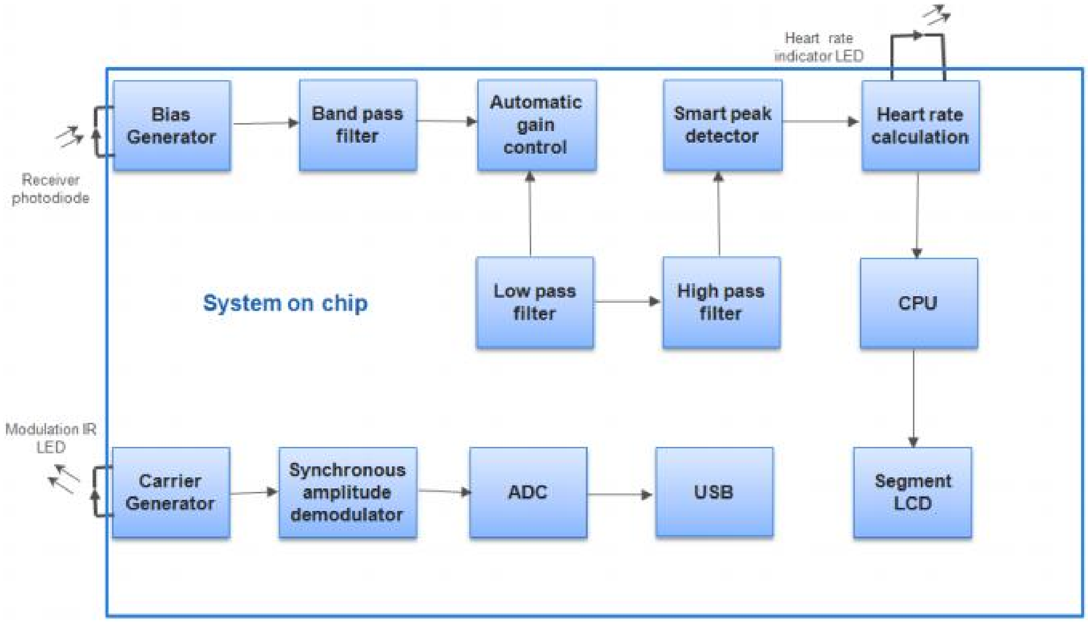


Figure 13 Photoplethysmograph (Measurement system, n.d)

2.2.4 Phonocardiograph

When valves in the heart open or close, they generate sounds that are rhythmic to the heart beat and audible through a stethoscope. Hence, it is possible to use sensitive microphone to collect these sounds.

In signal acquisition process, other sounds can also be picked up with desired heart sounds. Murmur, which can be seen from Figure 14, is an example for abnormal signals. These signals have different spectral characteristics thus it is potential using proper filters to visualize heartbeat. (Kumar, n.d).

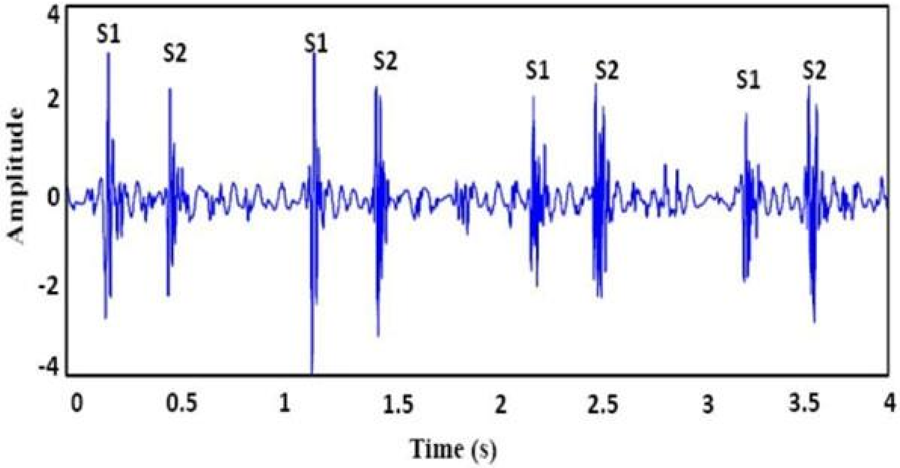


Figure 14 Phono Cardiographic signal (Christer Ahlstrom, 2008)

The audio recording chain requires a sequence of signal transformation:

- Sensor converts sound(vibration) to electrical signal
- Preamplifier
- High frequency filter
- Analog to digital converter

A processing unit is applied to calculate the heart beat from the time interval between heart thuds and represent the heart signal(pulse) on the screen. An example for phonocardiograph system was demonstrated in Figure 15.

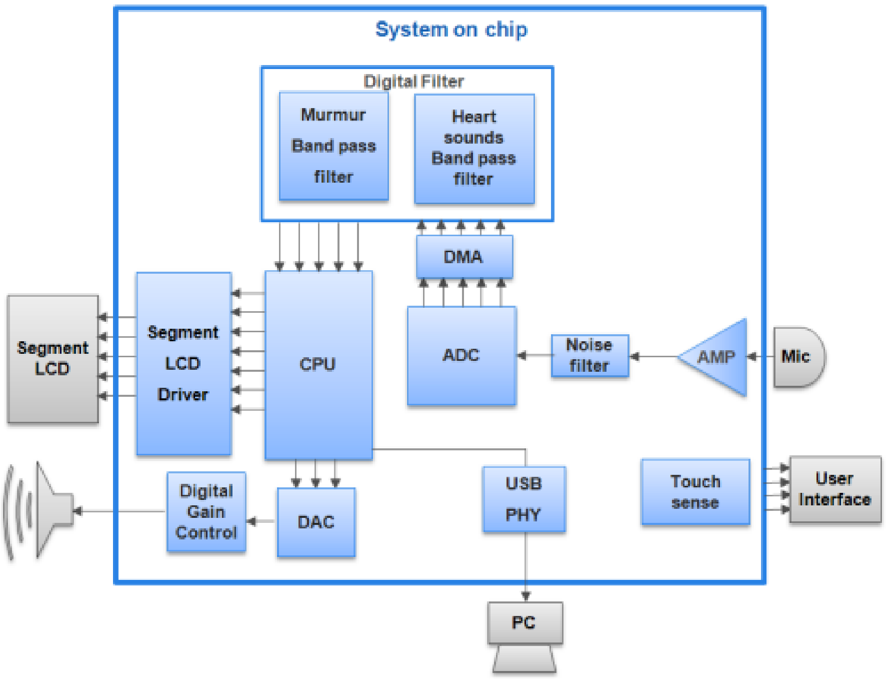


Figure 15 Phonocardiograph system (Measurement system, n.d)

3 ELECTRICAL THEORETICAL BASIS

3.1 Operational amplifier circuit

An Operational Amplifier or op-amp is a voltage amplifying component designed to be used with external feedback components such as resistors and capacitors between its output and input . It is a high-gain electronic voltage amplifier with a differential input and usually a single-ended output. Op-amps are among the most widely used electronic devices nowadays.

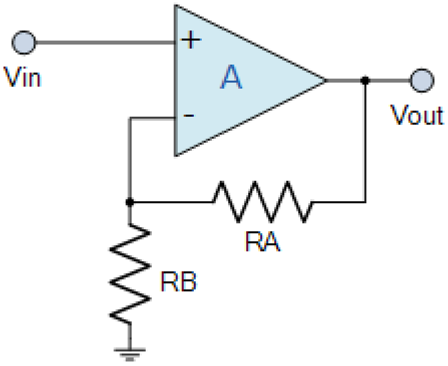
3.1.1 Non-inverter amplifier

This circuit does not invert the input signal or produce inverting signal while applying gain ratio:

$$\frac{Vout}{Vin} = 1 + \frac{RA}{RB}$$

Equation 1 Non-inverter amplifier voltage ratio

The input signal is connected to (+)non-inverting input



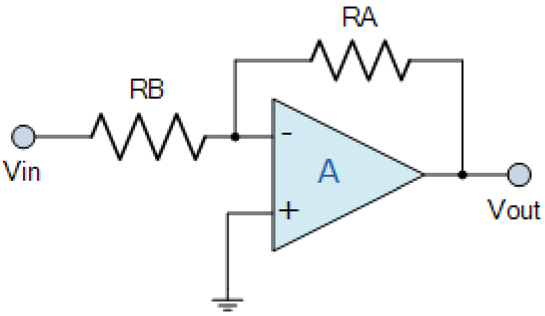


Figure 16 Inverting Op-amp Circuit (Amplifier n.d)

Figure 17 Non-inverting Op-amp Circuit (Amplifier n.d)

3.1.2 Inverter amplifier

This circuit does both inverting and amplifying the input signal with the ratio of :

$$\frac{Vout}{Vin} = -\frac{RA}{RB}$$

Equation 2 Inverted amplifier voltage ratio

The gain is controlled by negative feedback using feedback resistor RA . The input signal is fed to inverting input(-).

3.2 **Filters**

Electronic filters are circuits perform signal processing functions which in most of the cases are removing unwanted frequency signal from the original signal or enhance desired signal. These filters can be passive or active, analog or digital.
Passive filters are made up from passive components for example resistor, capacitor, inductors, … without amplifying components(transistors, operational amplifier…). Their output levels are always less than input.
Active filters are implemented using both passive and active electronic components along with external power source. They can provide voltage gain.

There are four primary types of filters which were named base on the frequency they remain after filtering process: lowpass filter, high pass filter, bandpass filter, notch filter which were listed in Figure 18. “Low” or “High” are regardless to any absolute values of frequency but relate to value of cutoff frequency.

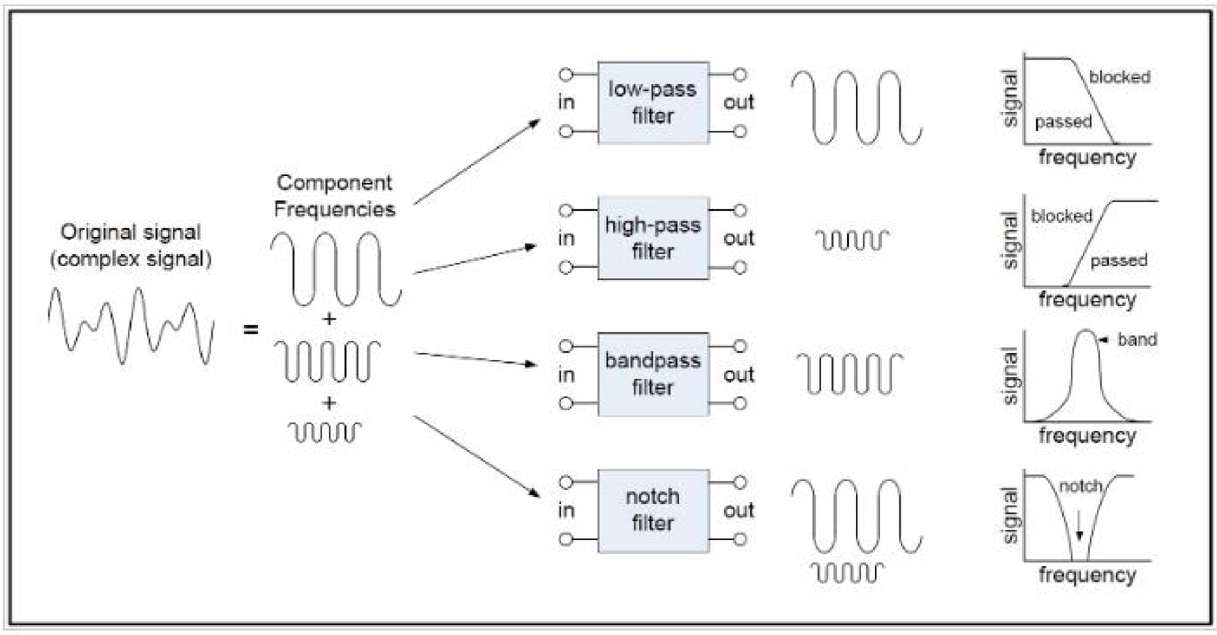


Figure 18 Basic depiction of four major filter types (Filter types n.d)

Response curve is used to describe a filter behave. This graph shows attenuation ratio (V_{IN}/V_{OUT}) over frequency. Attenuation is expressed in units of decibels(dB).

Below are some popular technical terms using for response curve(Figure 19):

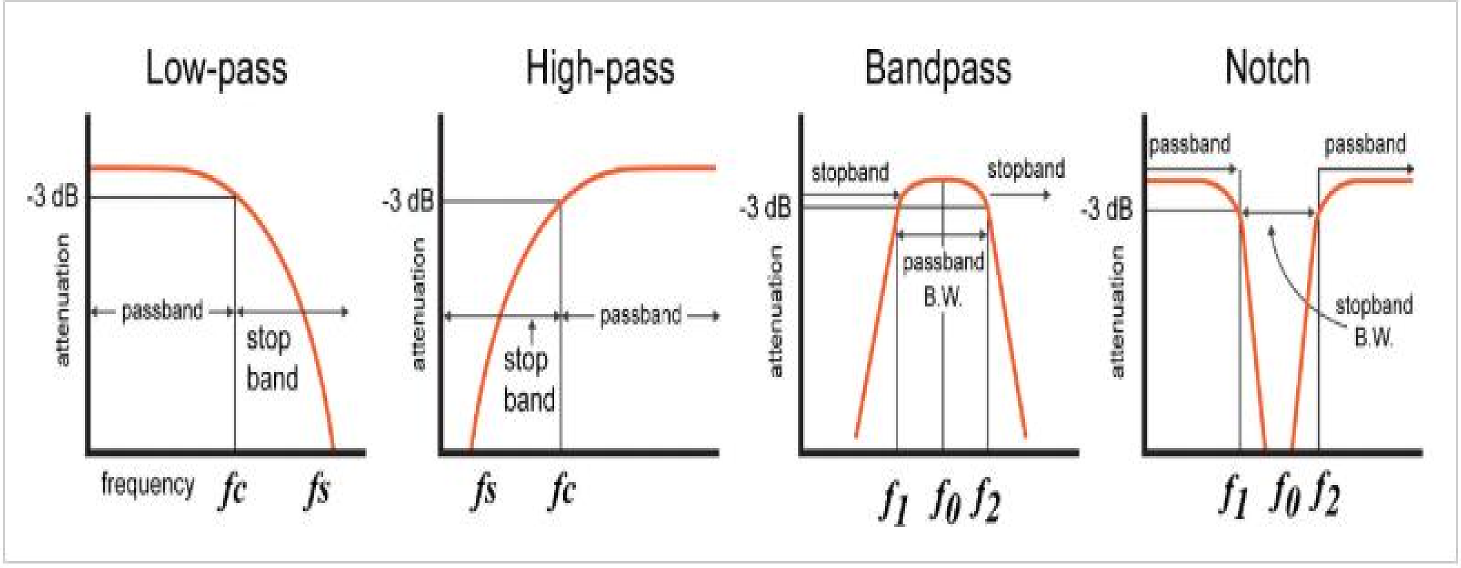


Figure 19 Response curves for the four major filter types (Filter types n.d)

- Cutoff frequency or “-3dB frequency” is the value of input frequency that cause the output signal amplitude a drop by -3dB in compare to the input signal amplitude. This is also the frequency where the output power reduced 50%. For lowpass and high pass filter there is only one cutoff frequency while for stopband and notch filter there two.
- Stopband frequency(f_S) is a particular frequency where the attenuation achieves a specified value.
- Gain of a filter is function of the output amplitude divided by coinciding input, usually displayed in Decibels:

$$Gain\ in\ dB = 20\log\frac{Vout}{Vin}$$

Equation 3 Filter gain in dB

3.2.1 Lowpass filter

Lowpass filter is made from capacitance, resistance and inductance to apply high attenuation for frequency above a specific value(cutoff frequency) as well as does no attenuation for signal below that frequency.
Components of a lowpass filter can vary from a resistor plus a capacitor for simple circuit or more pairs of resistor-capacitor and inductor.

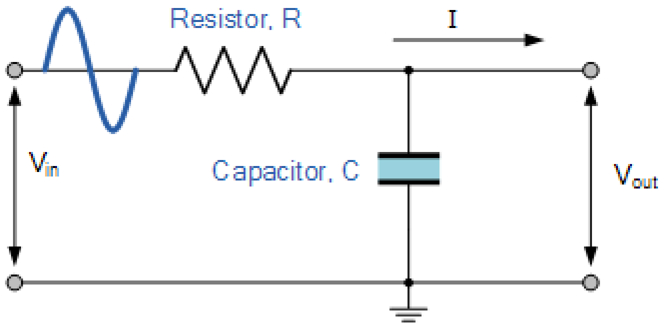


Figure 20 Simple RC circuit (Passive LPF n.d)

RC potential divider equation

$$\mathrm{Vout} = \mathrm{Vin} \times \frac{Zc}{Z} = \mathrm{Vin} \times \frac{Zc}{\sqrt{R^2 + Zc^2}}$$

Equation 4 RC potential divider equation

Where $\mathrm{Zc} = \frac{1}{2\pi\mathrm{fC}}$

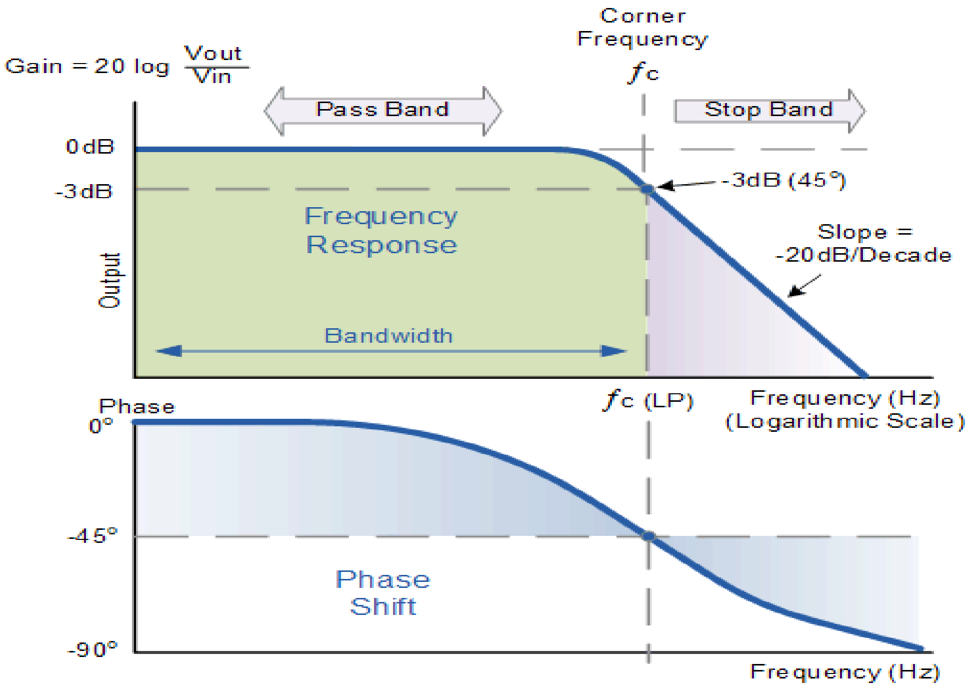


Figure 21 Frequency Response of a 1st-order Low Pass Filter (Passive LPF n.d)

The Bode Plot shows the frequency response of the filter: no significant change for low frequency until the it reaches cutoff frequency. After f_c point, the response of the circuit goes down to 0 at the slope of -20dB/Decade or “roll-off”.

Cutoff frequency

$$fc = \frac{1}{2\pi RC}$$

Equation 5 Cutoff frequency of low pass filter

Phase shift

$$\varphi = -\arctan(2\pi fRC)$$

Equation 6 Phase shift of low pass filter

3.2.2 Second order lowpass passive filter

Sometimes, the roll-off slope of -20dB/decade attenuation is not enough then two stages of filtering can be implemented.

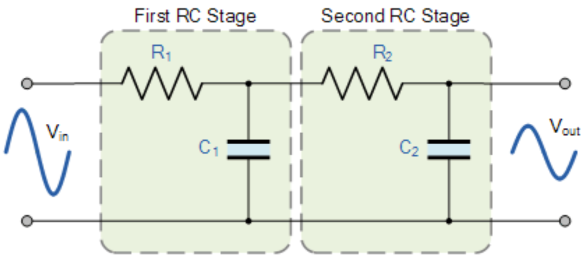


Figure 22 2nd order passive LPF (Passive LPF n.d)

Passive second order LPF can be installed by connecting two first order LPF in series.

Passive lowpass filter gain at f_c $(\frac{1}{\sqrt{2}})^n$ where n is number of filter stages.

Cutoff frequency

$$fc = \frac{1}{2\pi\sqrt{R1C1R2C2}}$$

Equation 7 Cutoff frequency of 2nd order lowpass filter

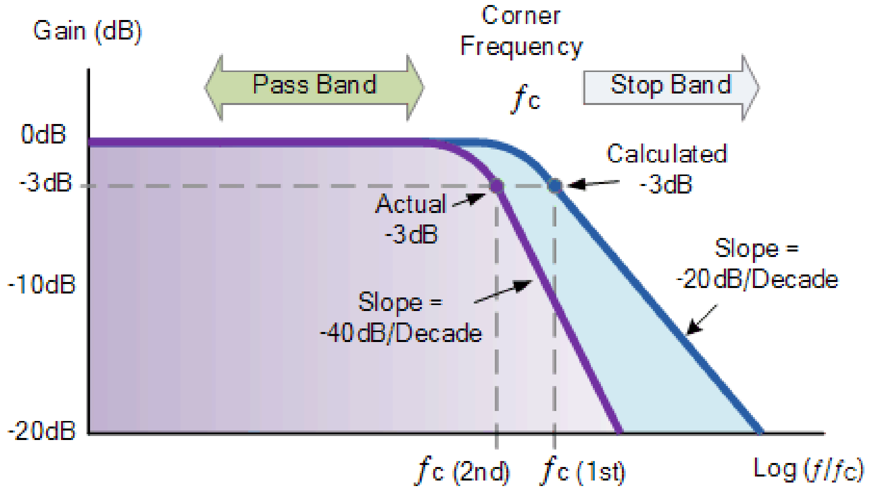


Figure 23 2nd order LPF frequency response (Passive LPF n.d)

3.2.3 Lowpass active filter

An active lowpass filter can be implemented by combining basic RC LPF with and operational amplifier.

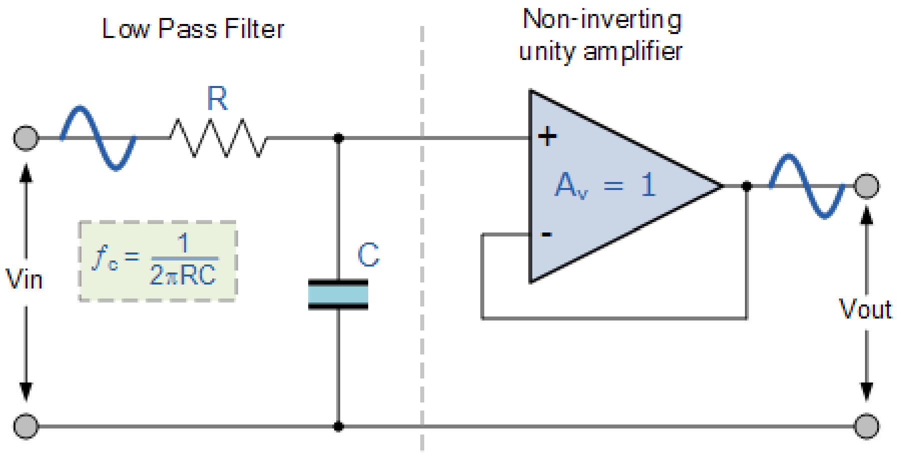


Figure 24 1st order active LPF (Active LPF n.d)

The op-amp is configured as a voltage buffer to achieve the gain of 1. The advantage of this configuration is that the op-amps high input impedance prevents over-loading on the filters output at the same time low output impedance prevents the filters cut-off frequency point from being affected by changes in the impedance of the load.

Incase greater voltage gain is required, amplification can be applied

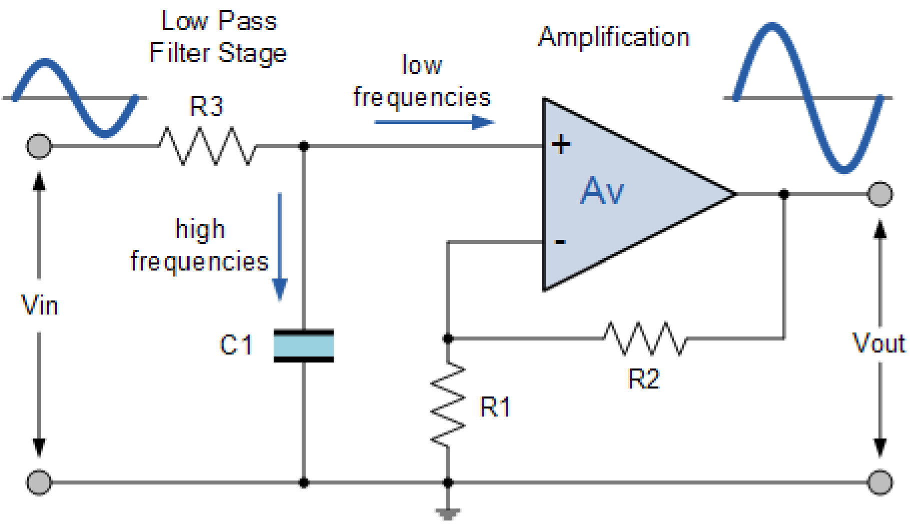


Figure 25 Active Low Pass Filter with Amplification (Active LPF n.d)

Frequency response of the circuit remains the same for passive LPF, however amplitude of output is increased by the gain of the amplifier A_F. For non-inverting amplifier:

- Gain from amplifier:

$$K_F = 1+R_2/R_1$$

Equation 8 Gain from amplifier

- Gain of 1 order active gain

$$Voltage\ gain(Kv) = \frac{Vout}{Vin} = \frac{KF}{\sqrt{1 + (\frac{f}{fc})^2}}$$

Equation 9 Voltage gain of 1st order active gain

Where f the input signal frequency in Hertz
f_C cutoff frequency in Hertz

The formula to calculate cutoff frequency is the same as for passive RC LPF:

$$fc = \frac{1}{2\pi RC}$$

Equation 10 Active LPF cutoff frequency

The gain corresponding to different input frequencies can be determined:

1. At low frequency, f<f_C $\frac{Vout}{Vin} \cong KF$
2. At cutoff frequency, f = f_C $\frac{Vout}{Vin} = \frac{KF}{\sqrt{2}}$
3. At high frequency, f> f_C $\frac{Vout}{Vin} < KF$

3.2.4 Second order active lowpass filter

It’s possible to connect two first orders passive LPFs to get second order passive LPF and add an op-amp with gain to the circuit to implement a second order active LPF. Stop band roll-off with be twice the first order filter at 40dB/decade.

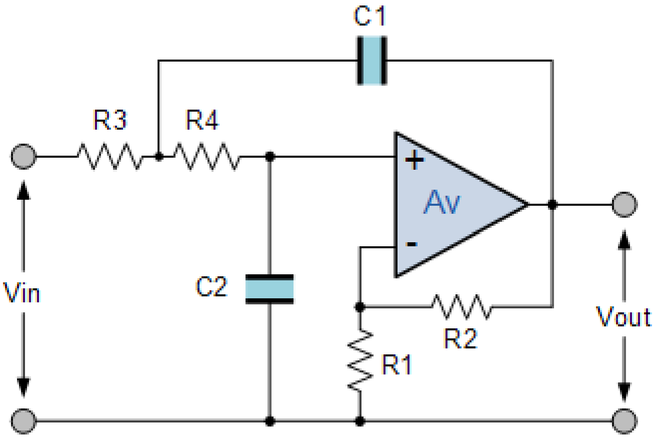


Figure 26 Second-order Active Low Pass Filter Circuit (Active LPF n.d)

$$Gain\ \ K = 1 + \frac{R2}{R1}$$

Equation 11 2nd order active LPF gain

Cutoff frequency $fc = \frac{1}{2\pi\sqrt{R3R4C1C2}}$

Equation 12 2nd order active LPF cutoff frequency

Cascading voltage gain: $K = K1 \times K2 \times \ldots K\,n$
The circuit above is also known as the lowpass Sallen-Key circuit which has transfer function:

$$\frac{Vout}{Vin} = \frac{K}{s^2(R3R4C1C2) + s\big(R3C2 + R4C2 + R3C1(1-K)\big) + 1}$$

Equation 13 Sallen-Key transfer function

Where s = j2πf

$$fc = \frac{1}{2\pi\sqrt{R3R4C1C2}}$$

$$Q = \frac{\sqrt{R3R4C1C2}}{R3C1 + R4C1 + R3C2(1-K)}$$

4 BASIS OF AUTOMATION THEORETICAL

4.1 Microcontroller

An embedded system is a computer system with a specific, dedicated function that is not designed so that it should ever need to be reprogrammed (i.e. engine control units, implantable medical devices, appliances, etc.) The most common type of embedded system is a microcontroller, which is a small computer system on a single integrated circuit. (Tyler, 2017).System in Figure 27 was an example of microcontroller.
Microcontroller includes three main parts: CPU(central processing unit), memory and communication. CPU's task is to perform calculation and other implementation of input data. Memory part involves ROM(read-only memory) and RAM(random access memory) which serve different data saving purposes. The communication part of a microcontroller includes

input and output port to receive and send digital and analog signal. Figure 27 illustrate the basic structure of a microcontroller.

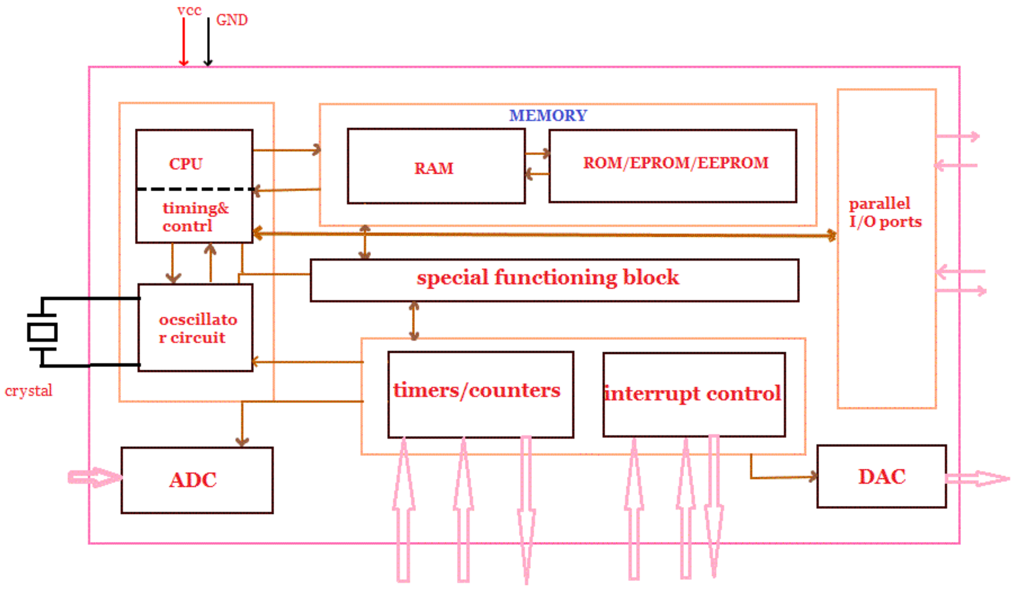


Figure 27 Structure of microcontroller (Vysakh, n.d)

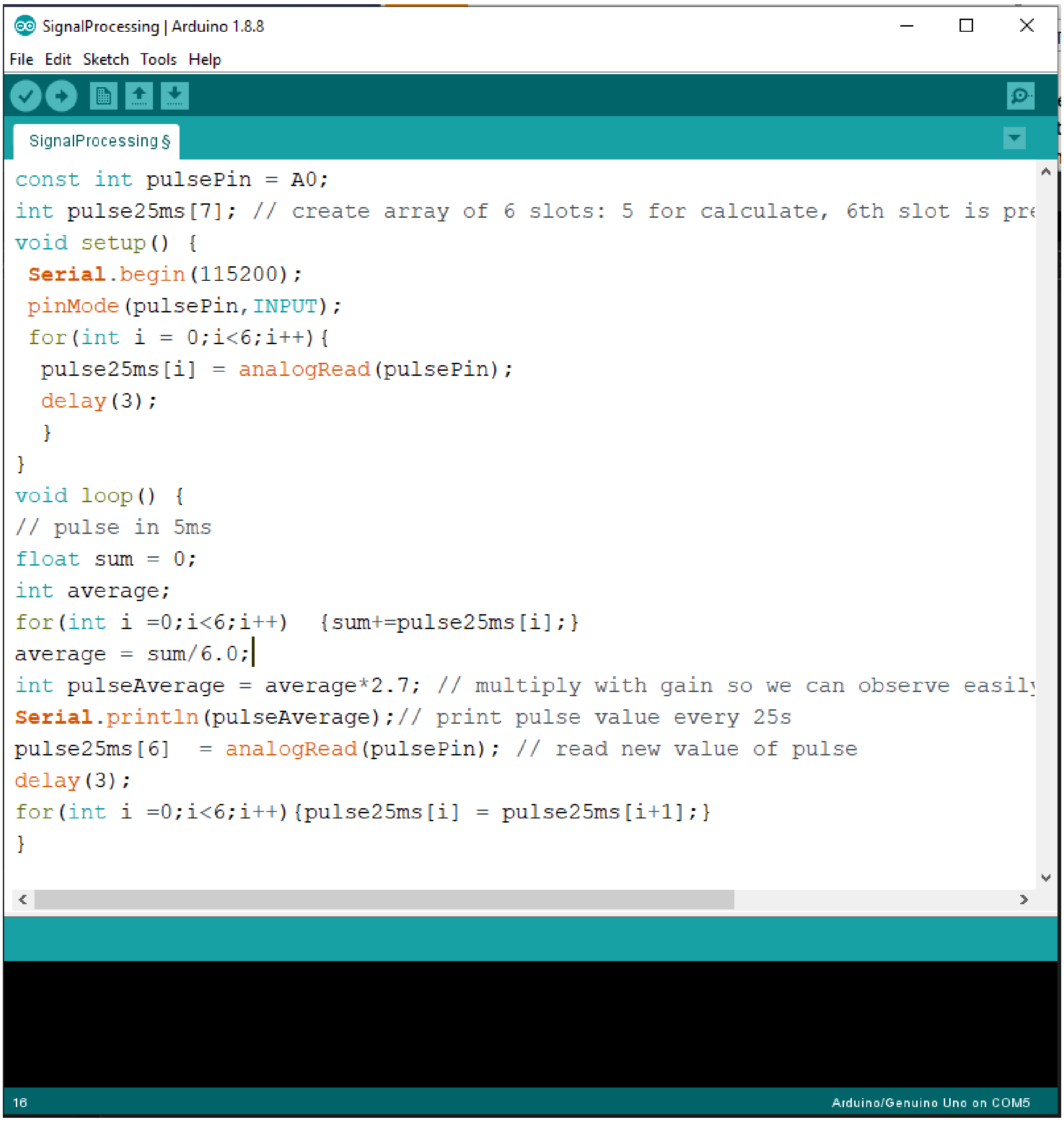
4.2 Programming background

4.2.1 Arduino

Arduino programs are written in the Arduino Integrated Development Environment (IDE) which is a special software allows user to write sketches for different Arduino boards. The programming language of Arduino called processing, similar to C/C++ language. An Arduino sketch must be compiled and upload on an Arduino board to implement.
There are two main functions used in nearly every Arduino sketch:

- Void setup(){ } : is used to set up initial condition for the sketch. This function runs only once(must be the first function) during the life of a program execution.
- Void loop(){} : is where Arduino essential functions are located. All the code between the curly braces is repeated as long as the Arduino is still working.

Figure 28 Shows an example of Arduino code sketch used in this thesis.


SignalProcessing | Arduino 1.8.8
File Edit Sketch Tools Help
SignalProcessing §
16
Arduino/Genuino Uno on COM5

```
const int pulsePin = A0;
int pulse25ms[7]; // create array of 6 slots: 5 for calculate, 6th slot is pre
void setup() {
 Serial.begin(115200);
 pinMode(pulsePin,INPUT);
 for(int i = 0;i<6;i++){
  pulse25ms[i] = analogRead(pulsePin);
  delay(3);
  }
}
void loop() {
// pulse in 5ms
float sum = 0;
int average;
for(int i =0;i<6;i++)  {sum+=pulse25ms[i];}
average = sum/6.0;
int pulseAverage = average*2.7; // multiply with gain so we can observe easil
Serial.println(pulseAverage);// print pulse value every 25s
pulse25ms[6]  = analogRead(pulsePin); // read new value of pulse
delay(3);
for(int i =0;i<6;i++){pulse25ms[i] = pulse25ms[i+1];}
}
```

Figure 28 Arduino sketch example

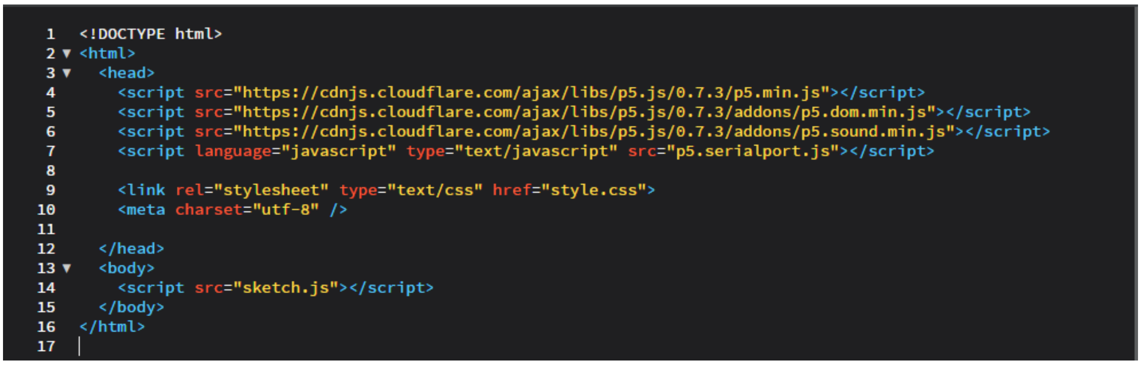
4.2.2 HTML(Hyper Text Markup Language)

HTML is a programming language created to build website which can be view by anyone else via the Internet. It is considered easy to learn and quite powerful for web creation. Nowadays, HTML is still under the revision and evolution from user and W3C, the company is responsible to develop and maintain this programming language.

HTML consists of group of short codes(tags) written in a text-file by the site author. The tags are what distinguish HTML code and a regular text. Each tag type implements distinct function in HTML page and only its effect is on the web display.

This text-file is saved as a HTML file and viewed through a browser.

File is read and translated into visible form by browser. Tags are involved to be used strictly correctly to create a website. However, any text-editor can be used to create these HTML pages.(Ross Shannon, 2012)
Figure 29 demonstrates an example HTML code tried in this thesis



```html
<!DOCTYPE html>
<html>
  <head>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.3/p5.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.3/addons/p5.dom.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.3/addons/p5.sound.min.js"></script>
    <script language="javascript" type="text/javascript" src="p5.serialport.js"></script>

    <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8" />

  </head>
  <body>
    <script src="sketch.js"></script>
  </body>
</html>

```

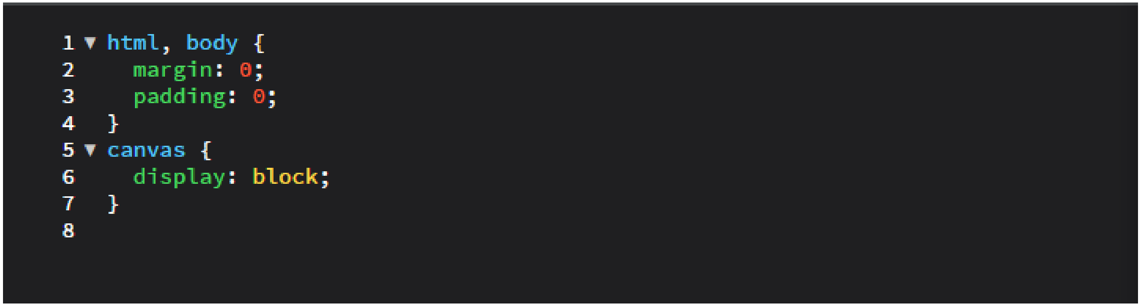
Figure 29 HTML Code Example

4.2.3 CSS(Cascading Style Sheets)

CSS is a language to specify how the documents are displayed in a webpage for example the layout, style, size,…
Web browsers apply CSS rules to affect the display of a document. CSS has the general form:

- Set of properties update the appearance of HTML content.
- Selector to select the element or group of elements website author want to apply the properties on.

These rules are enclosed inside a stylesheet or .css file.(MDN web docs CSS, n.d)
Figure 30 shows an example css code used in in thesis.



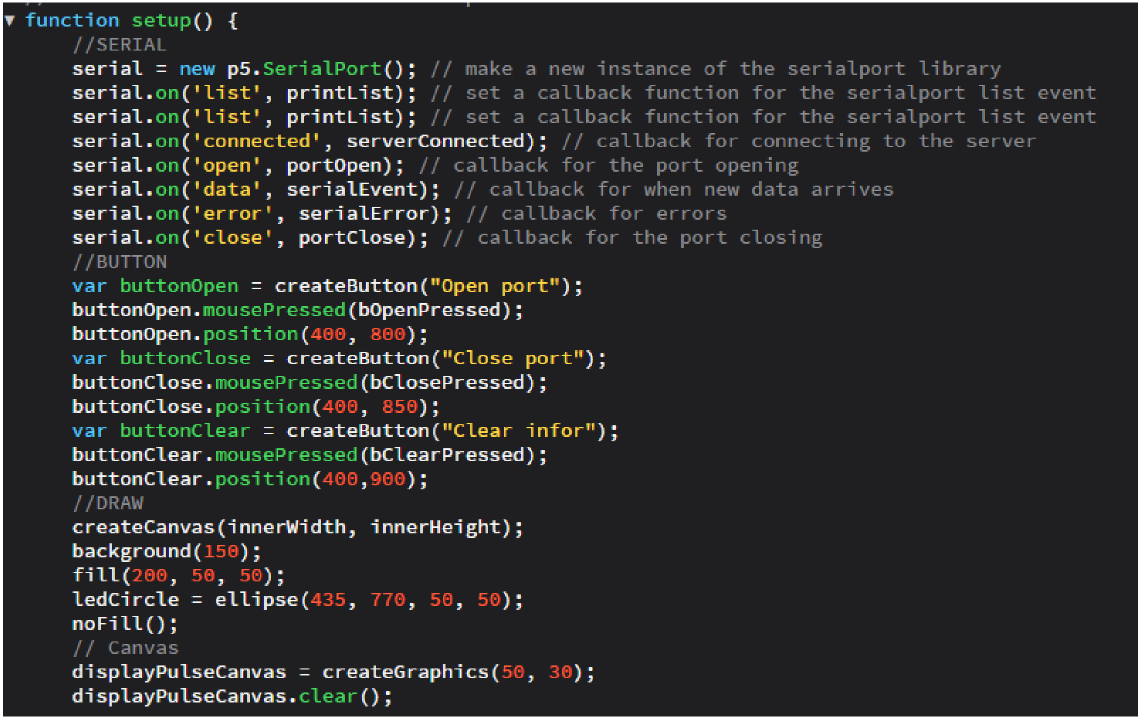
```css
html, body {
  margin: 0;
  padding: 0;
}
canvas {
  display: block;
}

```

Figure 30 CSS Code Example

4.2.4 JavaScript (JS)

JavaScript is a programming language for website interaction implementation. Thanks to JavaScript layer, the website becomes more dynamic and interactive. This language is supported by a large number of libraries and frameworks to make the work easier. JavaScript is fully integrated with HTML/CSS, supported by popular browsers and most of the time execute simply. Therefore, it is one of the most popular programming languages for web creation and design.

Figure 31 is an example of JavaScript code used in the thesis.



```
▼ function setup() {
    //SERIAL
    serial = new p5.SerialPort(); // make a new instance of the serialport library
    serial.on('list', printList); // set a callback function for the serialport list event
    serial.on('list', printList); // set a callback function for the serialport list event
    serial.on('connected', serverConnected); // callback for connecting to the server
    serial.on('open', portOpen); // callback for the port opening
    serial.on('data', serialEvent); // callback for when new data arrives
    serial.on('error', serialError); // callback for errors
    serial.on('close', portClose); // callback for the port closing
    //BUTTON
    var buttonOpen = createButton("Open port");
    buttonOpen.mousePressed(bOpenPressed);
    buttonOpen.position(400, 800);
    var buttonClose = createButton("Close port");
    buttonClose.mousePressed(bClosePressed);
    buttonClose.position(400, 850);
    var buttonClear = createButton("Clear infor");
    buttonClear.mousePressed(bClearPressed);
    buttonClear.position(400,900);
    //DRAW
    createCanvas(innerWidth, innerHeight);
    background(150);
    fill(200, 50, 50);
    ledCircle = ellipse(435, 770, 50, 50);
    noFill();
    // Canvas
    displayPulseCanvas = createGraphics(50, 30);
    displayPulseCanvas.clear();
```

Figure 31 JavaScript example code

5 METHODOLOGY

5.1 Selection process for measurement method

Choosing the suitable method to measure the heart rate is one of the challenges in this project. As the targets of this project are animals, photoplethysmography is not ideal. The thick fur of these animal blocks most of infrared light hence the measurement is extremely hard. For ECG, it usually requires shaving these animals in at least 3 places to place electrodes and obtain proper measurement by a non-invasive method is preferable. Electrodes can be also attached to target’s tongue, but these sensing components may fall out because of wet surface in the tongue. Simple pulse palpation or stethoscope cannot be used in this situation as we need a continuous and more precise method. Phonocardiograph is the most promising method for our needs as sound can propagate though solid environment(fur) well. Moreover, with proper sensing process, the data can be collected continuously and more precise.

Our solution is to use a stethoscope to acoustically collect the heartbeat sound and use electrical circuit to make this tiny sound audible. Further signal processing and display implemented on this sound will be illustrated in detail in the next section.

5.2 Heartbeat calculations

There were technical terms and abbreviations will be used frequently in this section and thesis:

- BPM: beat per minute.
- IBI: inter beat interval, the time interval between 2 points that have the same amplitude and direction.
- P: peak value of signal.
- T: trough or bottom value of signal.
- Amplitude = P -T.

Figure 32 presents the value collected by sensor and notation reasoning the method the calculate heartbeat.

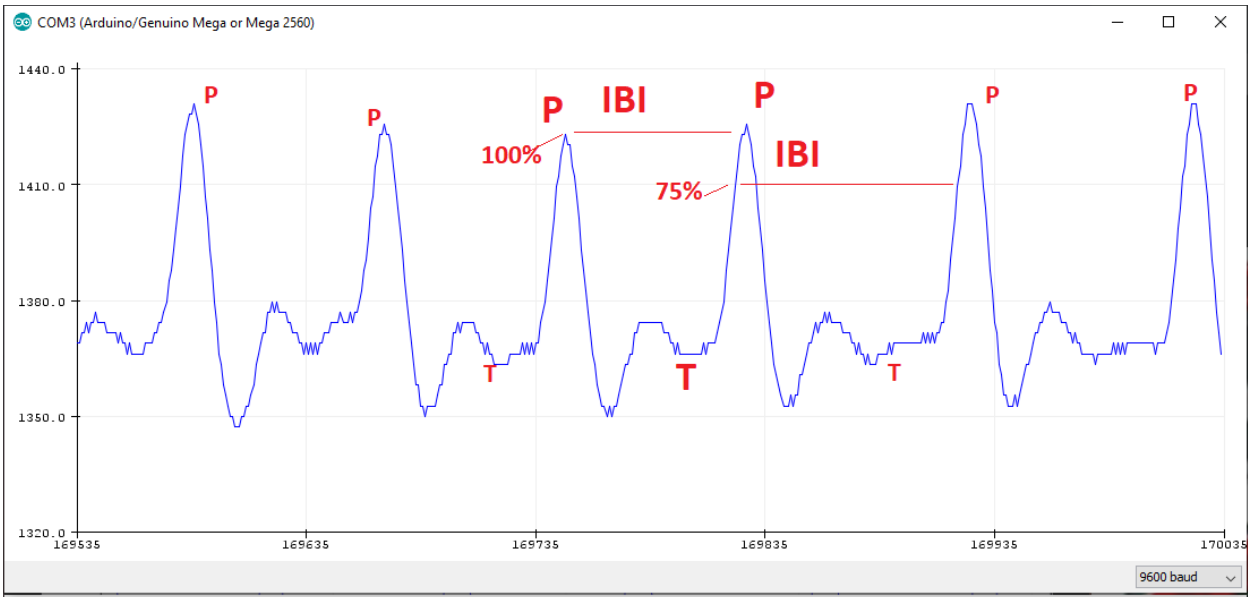


Figure 32 Signal from circuit graphed by Arduino

Base on phonocardiograph method and the data shown in figure 32, each heart beat sound can be recognized by T-point, P-peak and one small curve. P-peak was the top of a rapid upward raise, after that signal decline dramatically. There was a small curve of the second heartbeat sound before the low T-point. This pattern repeated periodically so the desired signal was predictable.

The peak value might differ due to variety reasons such as: measured target, condition of sensor, … Therefore, it was not ideal to pick a fixed high value for peak detection. Our effort was to determine the raising edge, update P-peak and then do calculation on IBI to find the heart rate. In practice, it was easier to confirm a raising edge on the way up than on P-peaks. If the signal was 75% of the amplitude or higher and the signal was increasing, then it was safe to consider the signal was at rising edge. Measuring the time between these points and calculating average number was a way to find the heart rate.

Moving average was an effective calculation to analyze data like this. The principle was simple, values were fed into an array and average number was compute. After that, the oldest value would be removed from array.

All value in the array would be shifted to the left before adding a new value to last position of the array. The computation went on after updating. Note that, the P-peak and T-bottom-value must be updated during the measurement to make sure no heartbeat was missed out.

5.3 **Architecture**

System architecture is a conceptual model to construct the structure, behavior and general view of a system. An architecture description represents system in a formal way while still supports clear explanation about structures and behaviors.

A system architecture usually contains system components and sub-system if needed to run the whole system. Formalized languages are now in construction which called architecture description languages(ADLs).
The next figure illustrates hardware architecture of the system in this thesis.

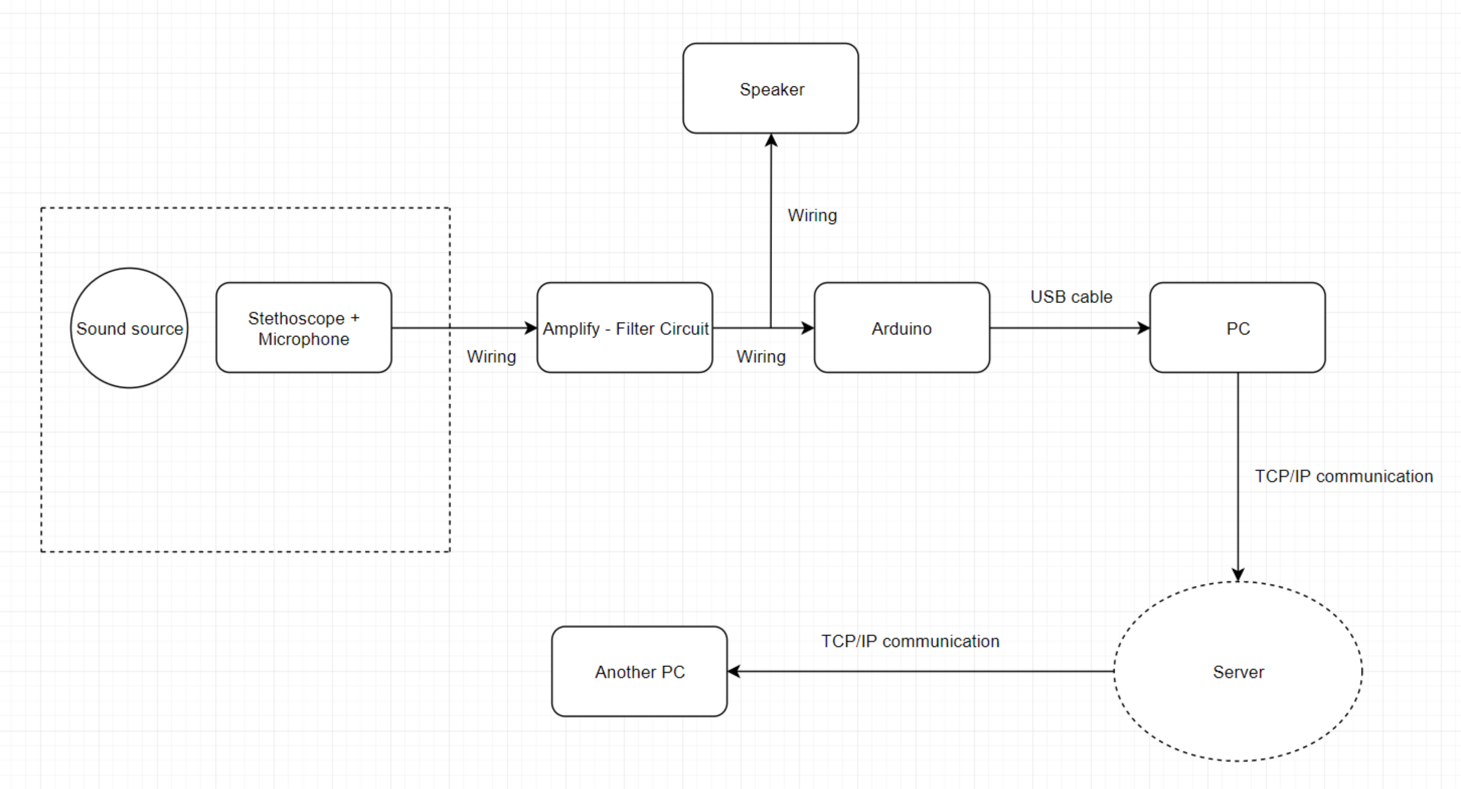


Figure 33 Hardware system architecture

As can be seen from the figure, sound signal from the cardiac system will be collected and acoustically amplified by a stethoscope. A sensitive microphone is connected to the end of stethoscope stem to convert physic vibration of sound into voltage signal. This signal then goes through a pre-amplifier, a filter and then another amplifier. Audible sound is generated after electrical signal activate the speaker. An Arduino receives signal from circuit, process to the signal to get BPM value. The Arduino is power by PC via a USB cable which is also used to power this board. Data after processing is send to PC continuously, and from PC to a local host server. This information can be viewed by other PCs which are allowed to connect to the network.

5.4 **Electrical design**

In this step of project, multiple works related to electronic components have been implemented for example: planning, drafting, simulation, testing, installation, …
For circuit design and simulation, used program is Multisim, an industrial-standard SPICE simulation and circuit design software for analog, digital and power electronics in education and research.
Final circuit contains 3 smaller circuits: preamplifier circuit, filter circuit, amplifier circuit

5.4.1 Sensor (Microphone)

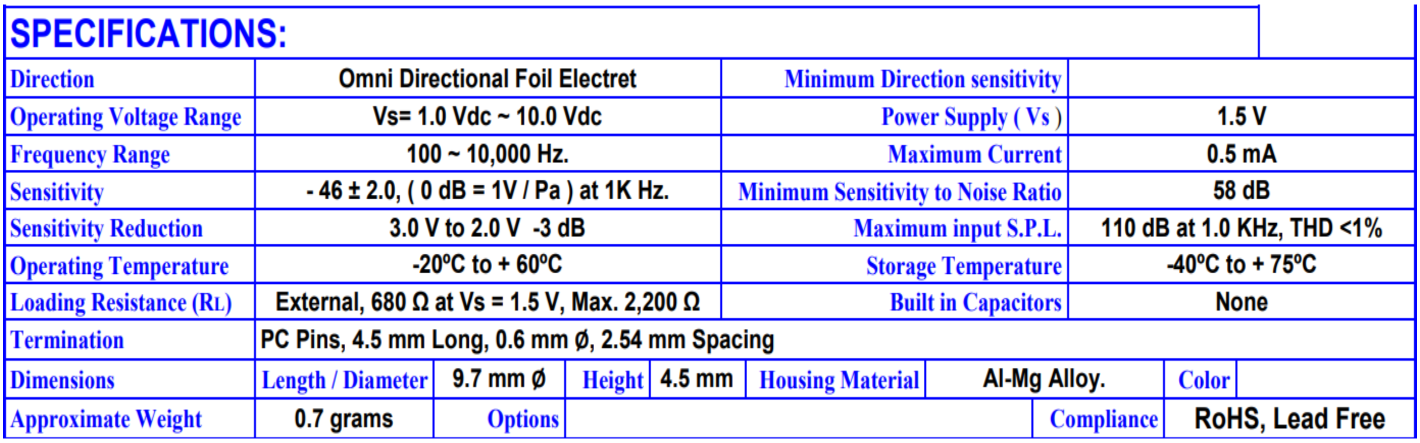
The key specification of a microphone is its sensitivity. The sensitivity of an analog microphone is the correspondent between physical input and electric voltage:

$$Sensitivity(dBV) = 20\log_{10}\left[\frac{sensitivity(mV/Pa)}{1000\,mV/Pa}\right]$$

Microphone sensitivity measured 94dB sound pressure level at 1 Pascal. From practical cases, the stethoscope can amplify tiny sound from cardiac system to level of 60-70 dB SPL. Base on this information, a microphone which has sensitivity lower than 50 is acceptable for the project. In this case, electret microphone was chosen. Figure 34 presented the structure of the microphone.

Figure 34 Electret microphone (sparkfun microphone, n.d)

Electret is a type of condenser microphone comes with permanent charge built inside. The microphone includes a pair of charged plates(+ and -) to record sound. Even there is no need to charge the plates again, the microphone still requires voltage for operation.
Table 1 includes the specification of chosen electret microphone.



<table>
<tr><th colspan="9">SPECIFICATIONS:</th></tr>
<tr><td>Direction</td><td colspan="4">Omni Directional Foil Electret</td><td colspan="2">Minimum Direction sensitivity</td><td colspan="2"></td></tr>
<tr><td>Operating Voltage Range</td><td colspan="4">Vs= 1.0 Vdc ~ 10.0 Vdc</td><td colspan="2">Power Supply (Vs)</td><td colspan="2">1.5 V</td></tr>
<tr><td>Frequency Range</td><td colspan="4">100 ~ 10,000 Hz.</td><td colspan="2">Maximum Current</td><td colspan="2">0.5 mA</td></tr>
<tr><td>Sensitivity</td><td colspan="4">- 46 ± 2.0, (0 dB = 1V / Pa) at 1K Hz.</td><td colspan="2">Minimum Sensitivity to Noise Ratio</td><td colspan="2">58 dB</td></tr>
<tr><td>Sensitivity Reduction</td><td colspan="4">3.0 V to 2.0 V -3 dB</td><td colspan="2">Maximum input S.P.L.</td><td colspan="2">110 dB at 1.0 KHz, THD <1%</td></tr>
<tr><td>Operating Temperature</td><td colspan="4">-20ºC to + 60ºC</td><td colspan="2">Storage Temperature</td><td colspan="2">-40ºC to + 75ºC</td></tr>
<tr><td>Loading Resistance (RL)</td><td colspan="4">External, 680 Ω at Vs = 1.5 V, Max. 2,200 Ω</td><td colspan="2">Built in Capacitors</td><td colspan="2">None</td></tr>
<tr><td>Termination</td><td colspan="8">PC Pins, 4.5 mm Long, 0.6 mm ø, 2.54 mm Spacing</td></tr>
<tr><td>Dimensions</td><td>Length / Diameter</td><td>9.7 mm ø</td><td>Height</td><td>4.5 mm</td><td>Housing Material</td><td>Al-Mg Alloy.</td><td>Color</td><td></td></tr>
<tr><td>Approximate Weight</td><td>0.7 grams</td><td>Options</td><td colspan="3"></td><td>Compliance</td><td colspan="2">RoHS, Lead Free</td></tr>
</table>

Table 1 Electret microphone specification

5.4.2 Amplifier circuit

Pre-amplifier circuit is used to amplify the amplitude of the input signal from microphone(5-10mV) to higher level. Figure 35 is a demo for preamplifier circuit used in the project.

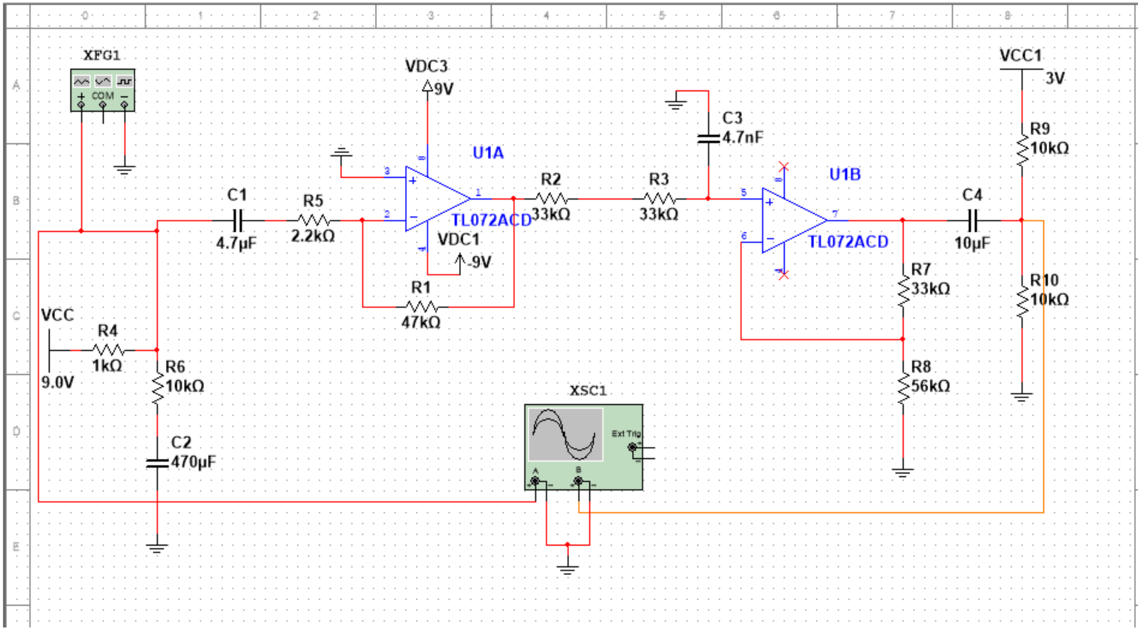
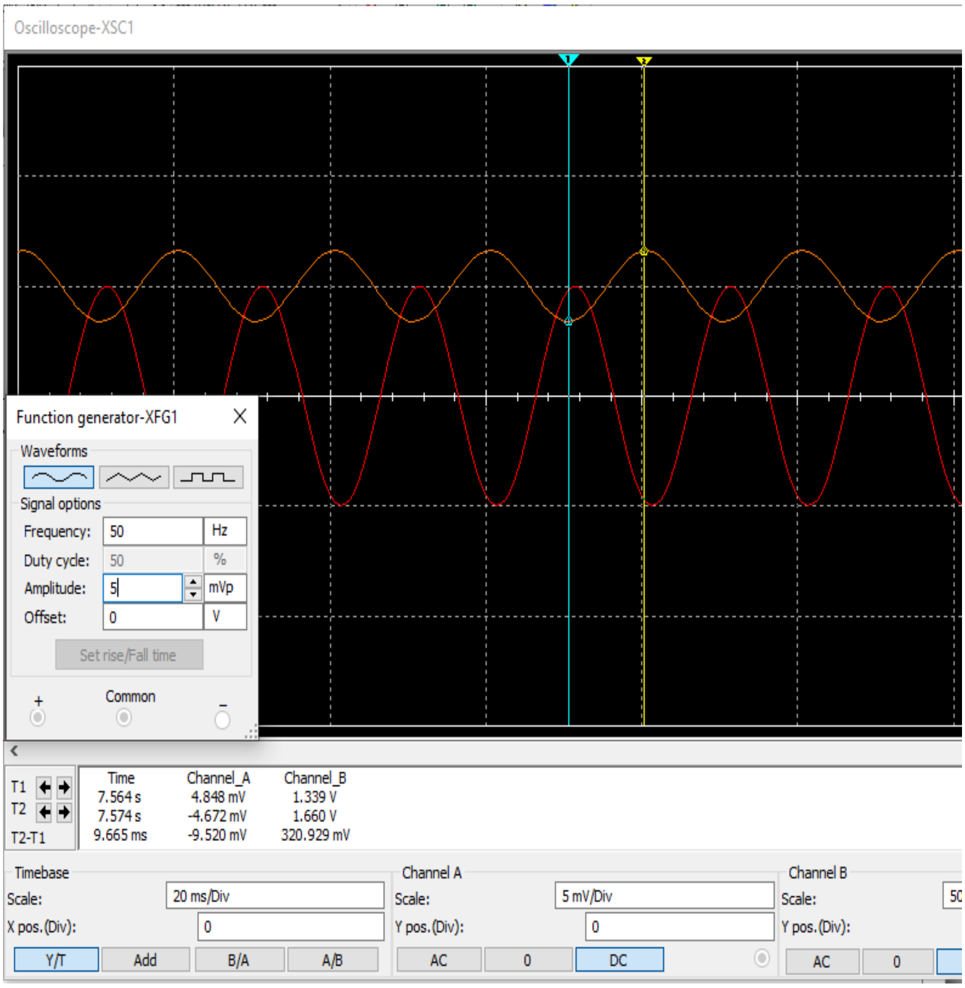


Figure 35 Amplifier circuit schematic

Op-amps used in the schematic can be any type that has low power operation and low noise. XFG1 is a signal generator which was used as the sound source to create small amplitude signal to running in this simulation. Figure 36 shows the original and amplified signal in the simulation through oscilloscope(Red line represents the input signal and orange line represents the output signal).



Frequency: 50 Hz
Duty cycle: 50 %
Amplitude: 5 mVp
Offset: 0 V

	Time	Channel_A	Channel_B
T1	7.564 s	4.848 mV	1.339 V
T2	7.574 s	-4.672 mV	1.660 V
T2-T1	9.665 ms	-9.520 mV	320.929 mV

Timebase
Scale: 20 ms/Div
X pos.(Div): 0

Channel A
Scale: 5 mV/Div
Y pos.(Div): 0

Channel B
Scale: 50
Y pos.(Div):

Figure 36 Amplifier circuit result

As can be seen from the oscilloscope, from input amplitude of 5 mVp the amplifier circuit can produce output of about 160 mVp. There is a small phase shift in the output signal however for project’s goal it is acceptable.

5.4.3 Lowpass filter circuit

As the desire frequency is fixed, building a lowpass filter circuit is a good choice to remove interferences and noise from the input signal. This circuit had cutoff frequency at about 4 Hz. There was a small amplifier circuit to make signal visible. Figure 37 presents the demo LPF circuit schematic used in the project.

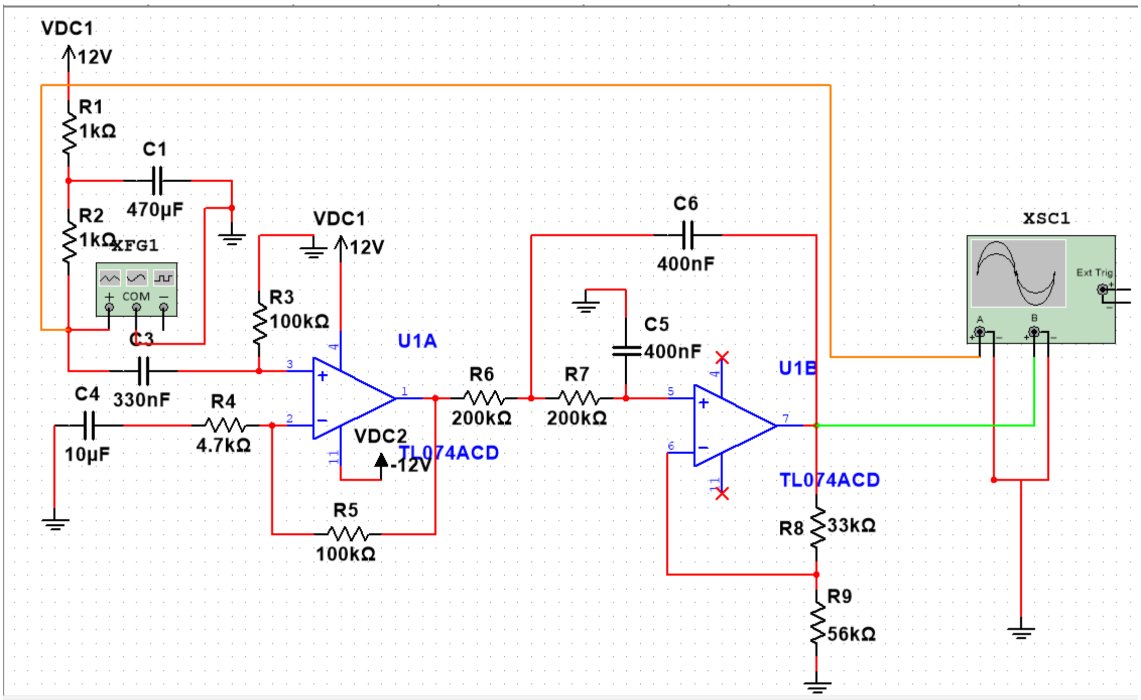
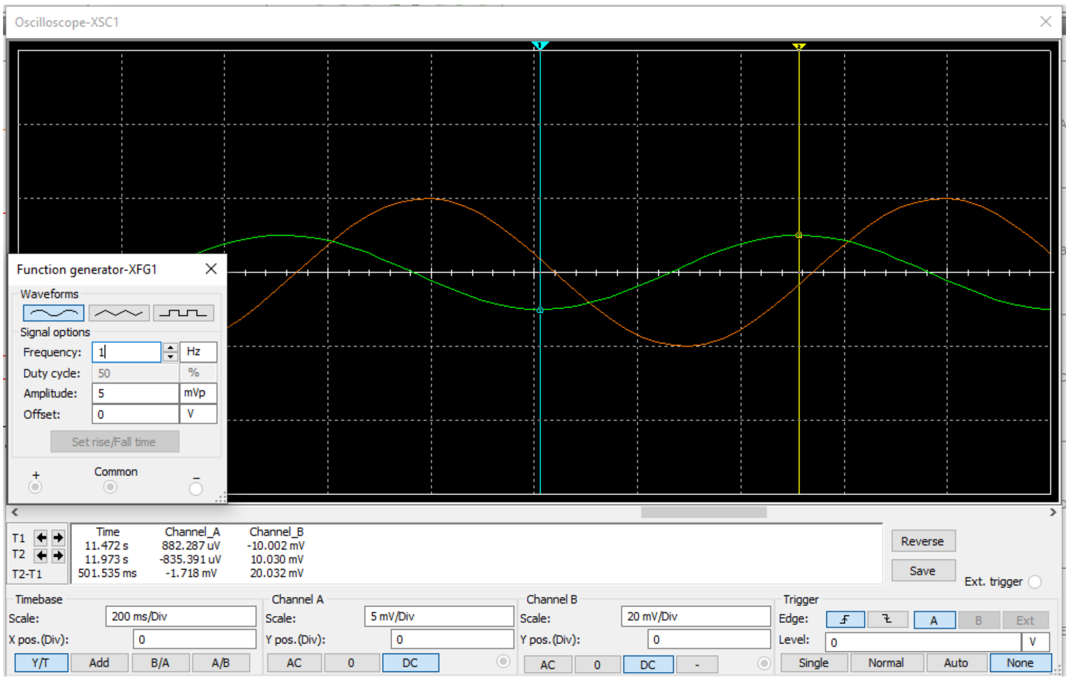


Figure 37 Lowpass filter schematic

This circuit was used to remove high frequency from output signal and amplify the result properly. XFG1 is a frequency-adjustable signal generator which is quite convenient to examine the behavior of output voltage corresponding to the input frequency. XSC1 was an oscilloscope with 2 channels: orange for signal before filtering and green for after filtering process. The next 3 figures present the behavior of output signal corresponding to input frequency.



Frequency: 1 Hz
Duty cycle: 50 %
Amplitude: 5 mVp
Offset: 0 V

	Time	Channel_A	Channel_B
T1	11.472 s	882.287 uV	-10.002 mV
T2	11.973 s	-835.391 uV	10.030 mV
T2-T1	501.535 ms	-1.718 mV	20.032 mV

Timebase
Scale: 200 ms/Div
X pos.(Div): 0

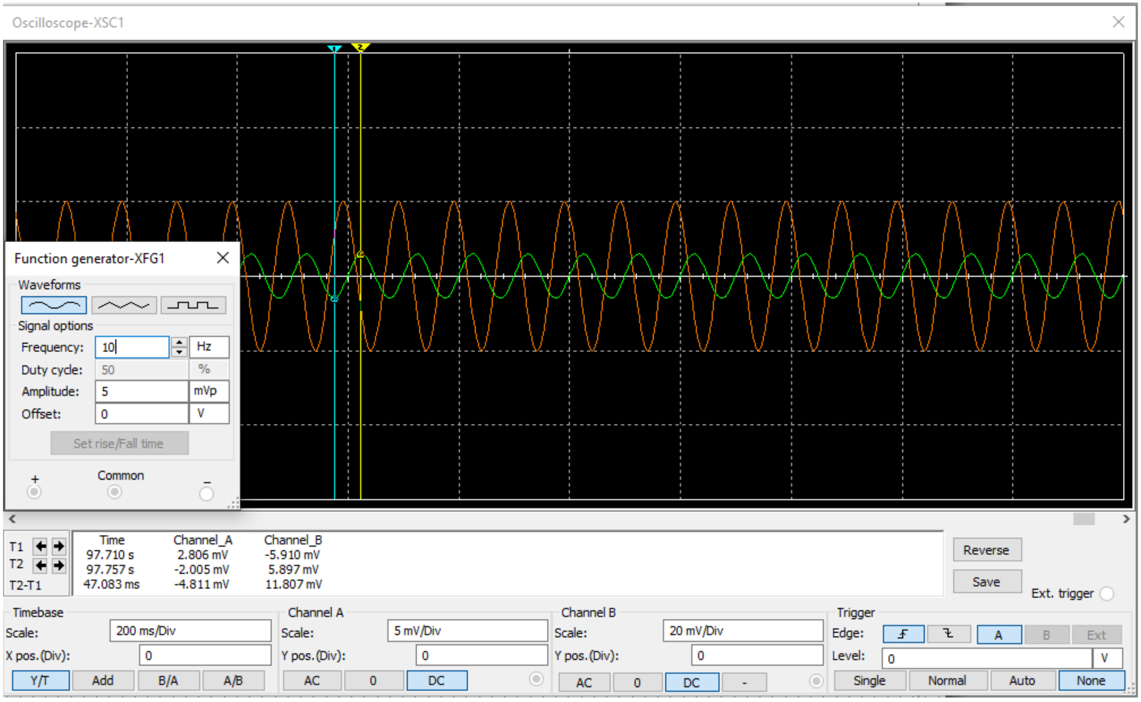
Channel A
Scale: 5 mV/Div
Y pos.(Div): 0

Channel B
Scale: 20 mV/Div
Y pos.(Div): 0

Trigger
Edge: A
Level: 0 V

Figure 38 Filtered signal at 1 Hz input frequency

According to figure 38, there is no signal distortion in the output and the amplification gain were about 2. Signal with frequency from 1 to 2 Hz were desired target for filter as it is 60-120 BPM in respect.



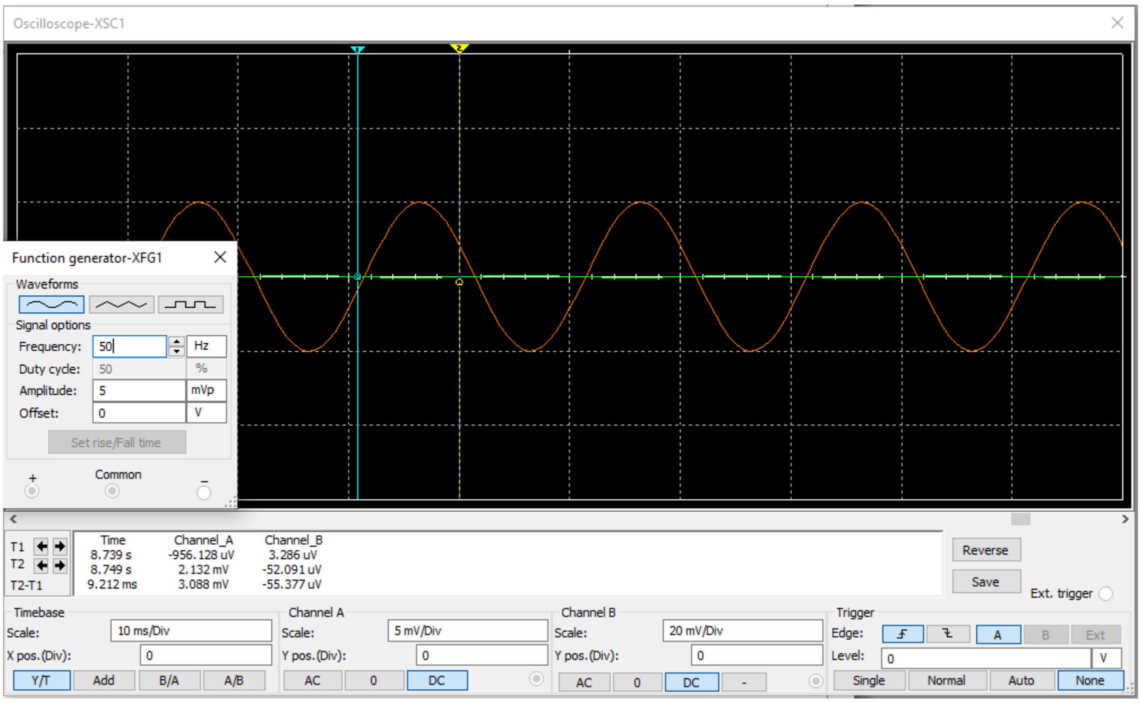
Frequency: 10 Hz
Duty cycle: 50 %
Amplitude: 5 mVp
Offset: 0 V

	Time	Channel_A	Channel_B
T1	97.710 s	2.806 mV	-5.910 mV
T2	97.757 s	-2.005 mV	5.897 mV
T2-T1	47.083 ms	-4.811 mV	11.807 mV

Timebase Scale: 200 ms/Div
X pos. (Div): 0
Channel A Scale: 5 mV/Div
Y pos. (Div): 0
Channel B Scale: 20 mV/Div
Y pos. (Div): 0
Trigger Level: 0 V

Figure 39 Filter signal at 10 Hz input frequency

As can be seen from figure 39, output signal decreased significantly to amplitude at about 5mPv. The cutoff frequency was at about 4 Hz, however in practice, a filter should be installed with an amplifier, so the output signal was still visible. The most popular noises were at 50Hz and above 100 Hz will be examined in the next figure(40).



Frequency: 50 Hz
Duty cycle: 50 %
Amplitude: 5 mVp
Offset: 0 V

	Time	Channel_A	Channel_B
T1	8.739 s	-956.128 uV	3.286 uV
T2	8.749 s	2.132 mV	-52.091 uV
T2-T1	9.212 ms	3.088 mV	-55.377 uV

Timebase Scale: 10 ms/Div
X pos. (Div): 0
Channel A Scale: 5 mV/Div
Y pos. (Div): 0
Channel B Scale: 20 mV/Div
Y pos. (Div): 0
Trigger Level: 0 V

Figure 40 Filtered signal at 50 Hz input frequency

In figure 40, the output signal was totally cancelled. Setting higher input frequency, oscilloscope showed the same result. The filter circuit can be considered as well designed.

5.4.4 Final circuit

After testing with components circuits in simulation, a final circuit was achieved as the combination of smaller circuits. The duties of this circuit were: pre-amplifying signal, filtering high frequency signal(above 4Hz), amplifying voltage to run a speaker, stabilizing input and output signal. Figure 41 shows the schematic of final circuit designed in this project.

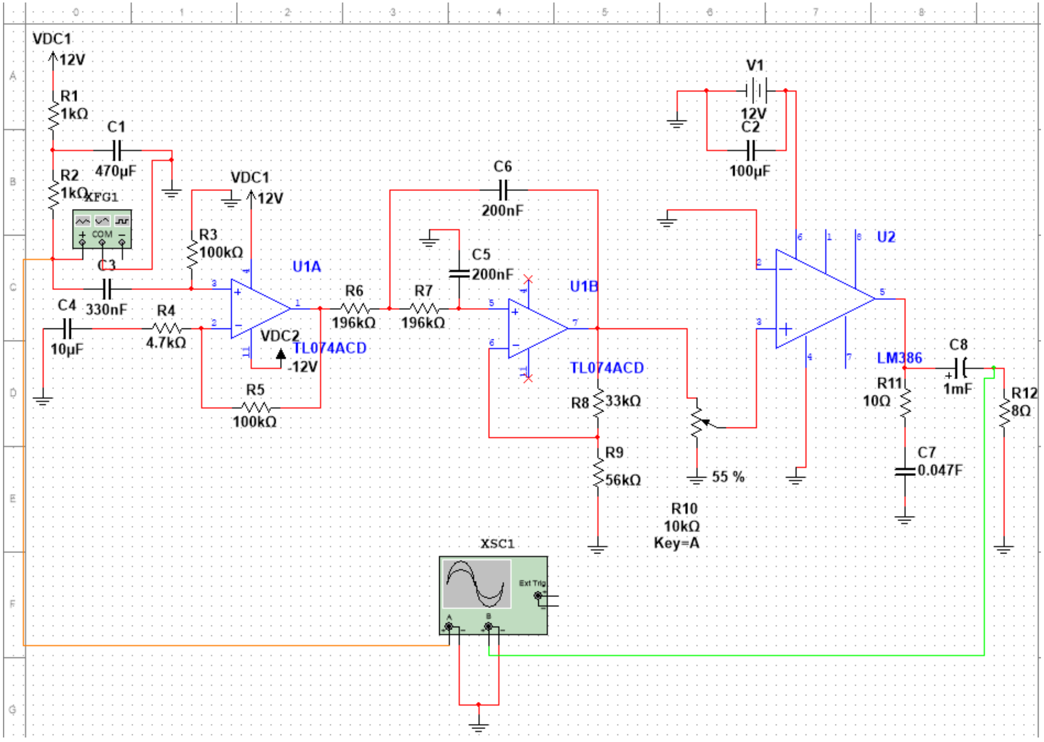
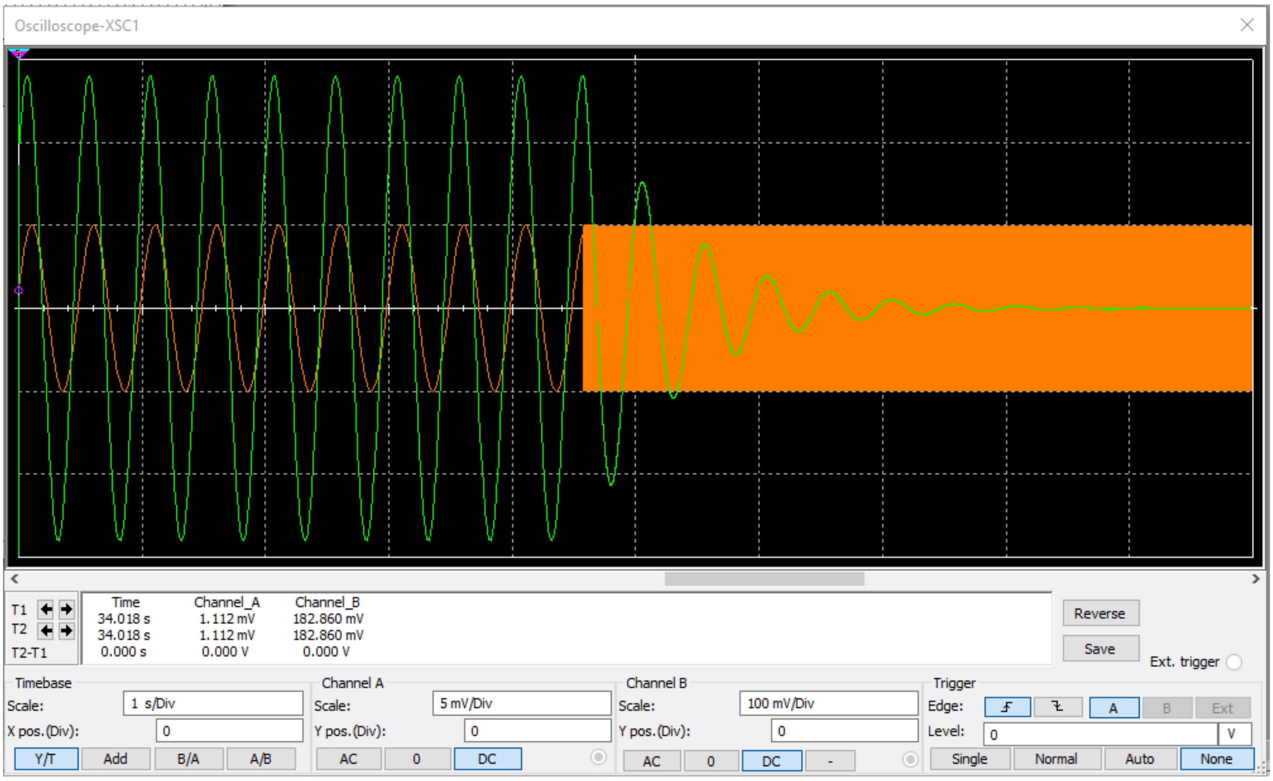


Figure 41 Final circuit schematic

In the schematic, real speaker was replaced by resistor R12. R10 of 10k Ohms was added to enable volume controlling. Electret microphone was replaced by XFG1 signal generator. The microphone and op-amps were power by 12V voltage sources which in practice can be replaced with series of batteries. Figure 42 demonstrate the output signal of the circuit coinciding to the frequency of the input signal.



	Time	Channel_A	Channel_B
T1	34.018 s	1.112 mV	182.860 mV
T2	34.018 s	1.112 mV	182.860 mV
T2-T1	0.000 s	0.000 V	0.000 V

Timebase
Scale: 1 s/Div
X pos.(Div): 0

Channel A
Scale: 5 mV/Div
Y pos.(Div): 0

Channel B
Scale: 100 mV/Div
Y pos.(Div): 0

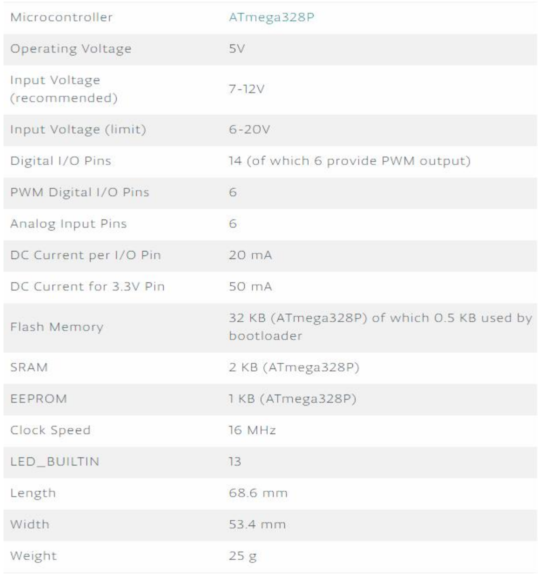
Trigger
Edge: A
Level: 0 V
Ext. trigger ○

Figure 42 Filtered signal at 2Hz and 100 Hz input frequency from final circuit

In figure 42, green line was output signal and orange line was the input signal. On the left side of the screen, signal was fed with frequency of 2Hz. Move to the right side of the oscilloscope screen, signal frequency was changed to 100 Hz. The change is significant, at 2 Hz, signal was amplified and remains the original form even there was small phase shift in the output. When frequency raised to 100Hz , output signal was totally cancelled which was the target of this circuit.

5.4.5 Arduino Uno

Arduino is microcontroller board including simple hardware and software. Arduino platform has become popular for small to medium project run by both professional to electrical amateur. User can load code to Arduino by a USB cable and Arduino IDE, unlike other programmable circuit board. Uno is one of the most popular boards in the Arduino family and suitable for beginners.(Arduino intro, n.d)
Figure 43 shows the appearance and specification of Arduino Uno.



Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

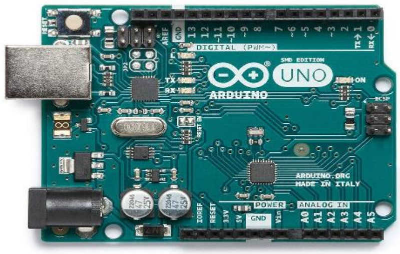


Figure 43 Arduino specification(Arduino intro, n.d)

5.5 **Programming**

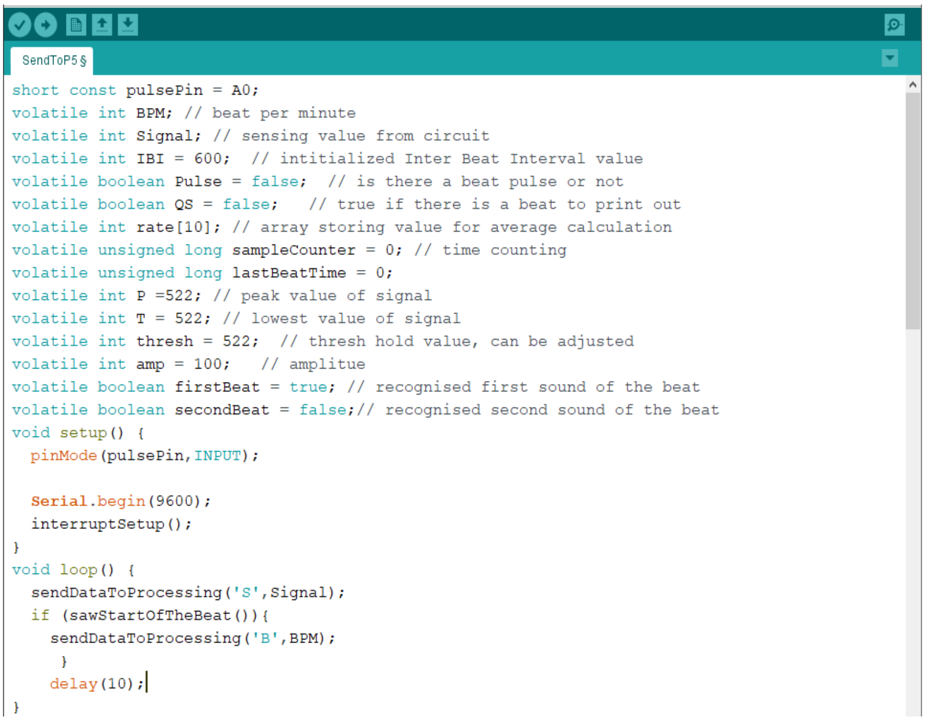
Programming is an essential part for this project. Data from sensing circuit must me collected, processed and sent to PC. Consequently, data presentation requires also an applicable program.

5.5.1 Heartbeat Arduino program

This program is used to read the electric voltage transferred from the circuit and calculate the heartbeat from the time interval between 2 consecutive P-peaks (Inter Beat InteEcobladerrval or IBI). The program was named “SignalProcessing.uno”, it also sends data continuously to the server.

Figure 44 shows the initialization, setup and main function of Arduino program. Details about the code was commented out in the figure.

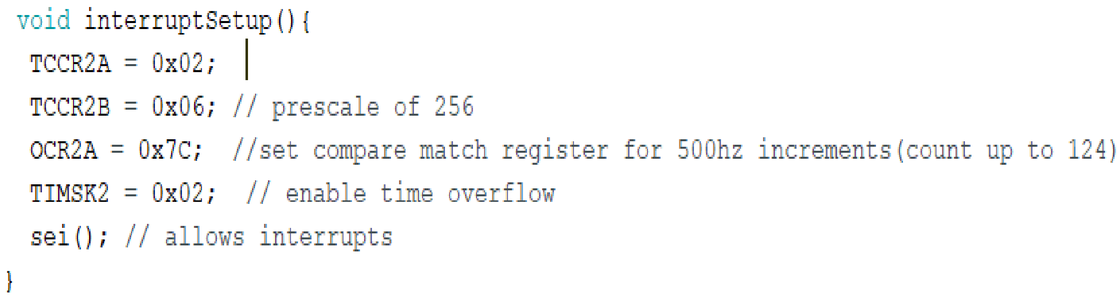
As can be seen from figure 44, the program sends pulse value data (marked with ’S’) constantly to Processing every 10 ms and check if there is start of a beat to send BPM value(marked with ‘B’).



```
short const pulsePin = A0;
volatile int BPM; // beat per minute
volatile int Signal; // sensing value from circuit
volatile int IBI = 600;  // intitialized Inter Beat Interval value
volatile boolean Pulse = false;  // is there a beat pulse or not
volatile boolean QS = false;   // true if there is a beat to print out
volatile int rate[10]; // array storing value for average calculation
volatile unsigned long sampleCounter = 0; // time counting
volatile unsigned long lastBeatTime = 0;
volatile int P =522; // peak value of signal
volatile int T = 522; // lowest value of signal
volatile int thresh = 522;  // thresh hold value, can be adjusted
volatile int amp = 100;   // amplitue
volatile boolean firstBeat = true; // recognised first sound of the beat
volatile boolean secondBeat = false;// recognised second sound of the beat
void setup() {
  pinMode(pulsePin,INPUT);

  Serial.begin(9600);
  interruptSetup();
}
void loop() {
  sendDataToProcessing('S',Signal);
  if (sawStartOfTheBeat()){
    sendDataToProcessing('B',BPM);
     }
    delay(10);
}
```

Figure 44 Arduino initializing code

In setup function, there was interruptSetup () function which was used to interruption for Arduino and Serial.begin(9600) to set the baud rate for Arduino. In figure 45, interruptSetup () function was presented and explained by comments.

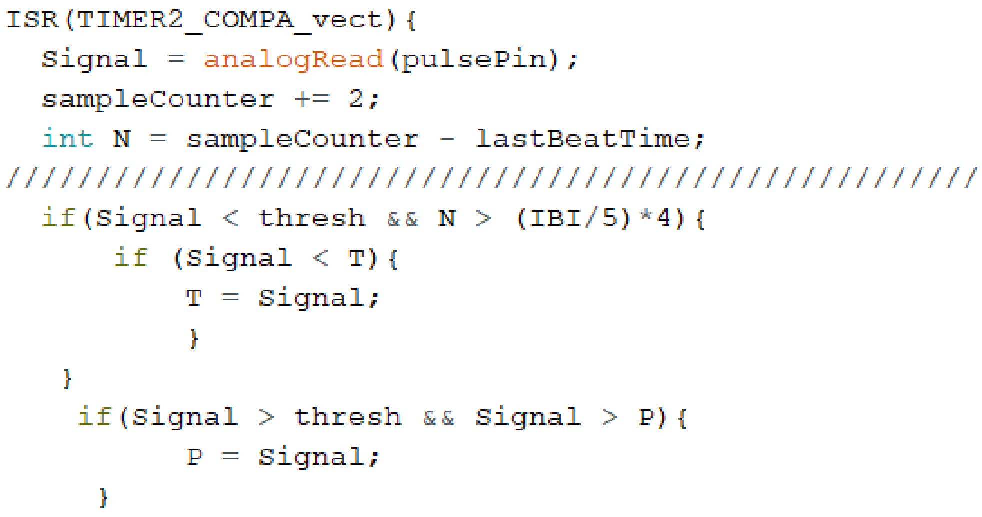


```
 void interruptSetup(){
  TCCR2A = 0x02;
  TCCR2B = 0x06; // prescale of 256
  OCR2A = 0x7C;  //set compare match register for 500hz increments(count up to 124)
  TIMSK2 = 0x02;  // enable time overflow
  sei(); // allows interrupts
}
```

Figure 45 interruptSetup ();

There are three Timers in Arduino Uno (Timer0, Timer1, Timer2) with different specification and can be activated as well as adjusted to change the counting down time. In this project, Timer2 was chosen to enable data updating and calculation. Timer2 was set to CTC mode and took 2 ms to count to 124 as pre-scale of 256 was chosen. Every time the internal clock of Timer2 reached 124, a specific function called Interrupt Service Routine or ISR which will be presented later will implemented immediately. ISR function will bypass any function running at that moment thanks to sei() functions enabled global interrupts. Interrupt Service Routine function of this program is demonstrated in figures 46,47 and 48.

Figure 46 presents the code for data and peak value updating.



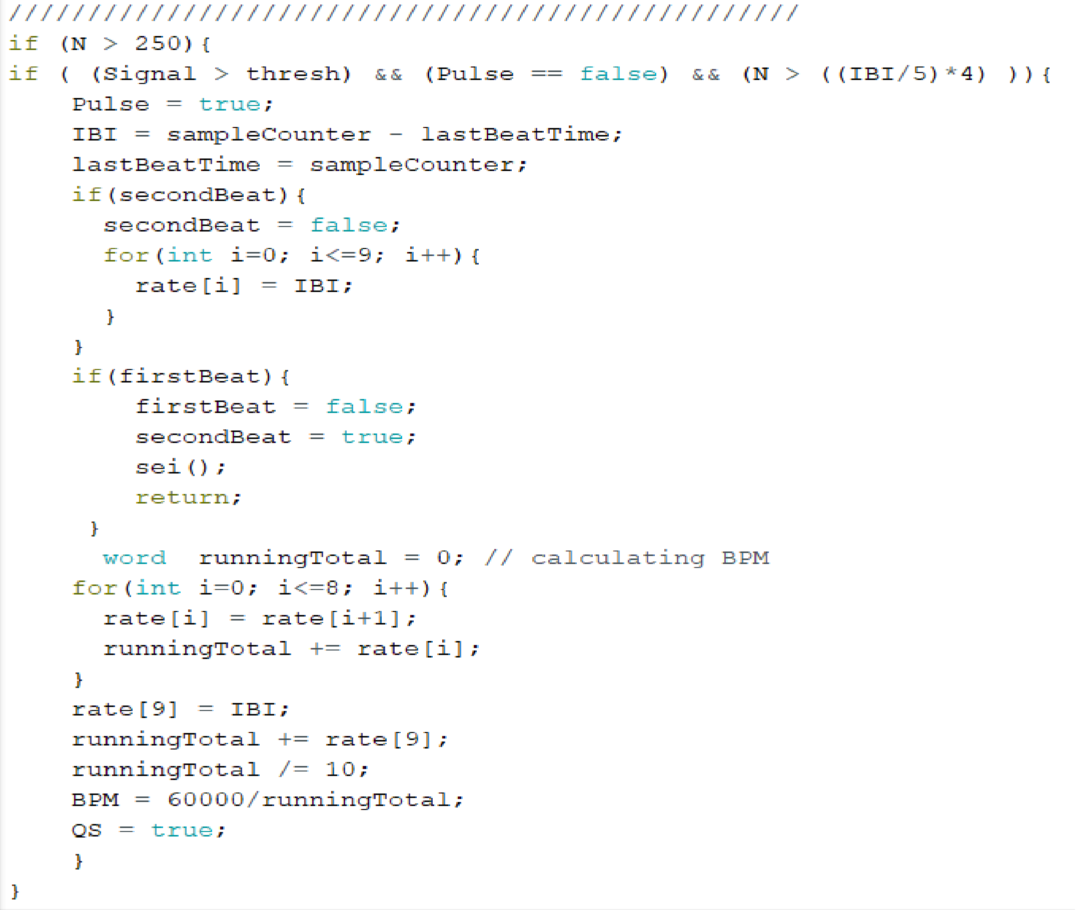
```
ISR(TIMER2_COMPA_vect){
  Signal = analogRead(pulsePin);
  sampleCounter += 2;
  int N = sampleCounter - lastBeatTime;
//////////////////////////////////////////////////////
  if(Signal < thresh && N > (IBI/5)*4){
      if (Signal < T){
          T = Signal;
          }
   }
    if(Signal > thresh && Signal > P){
          P = Signal;
     }
```

Figure 46 ISR function (part 1)

ISR(TIMER_COMPA_Vect) was the function called every 2 ms by the Timer2. The first thing it did was receiving analog signal from pulsePin. SampleCounter added up by 2 to update the time and N was used to store the time interval for comparing with the previous IBI. The profound purpose of N is to avoid noise of signal.

Thresh variable was initialized at 541 and updated at 50% of the amplitude later. The trough value was only update N was larger than 80% of IBI to reduce the noise.

Figure 47 illustrate how program determine a beat.



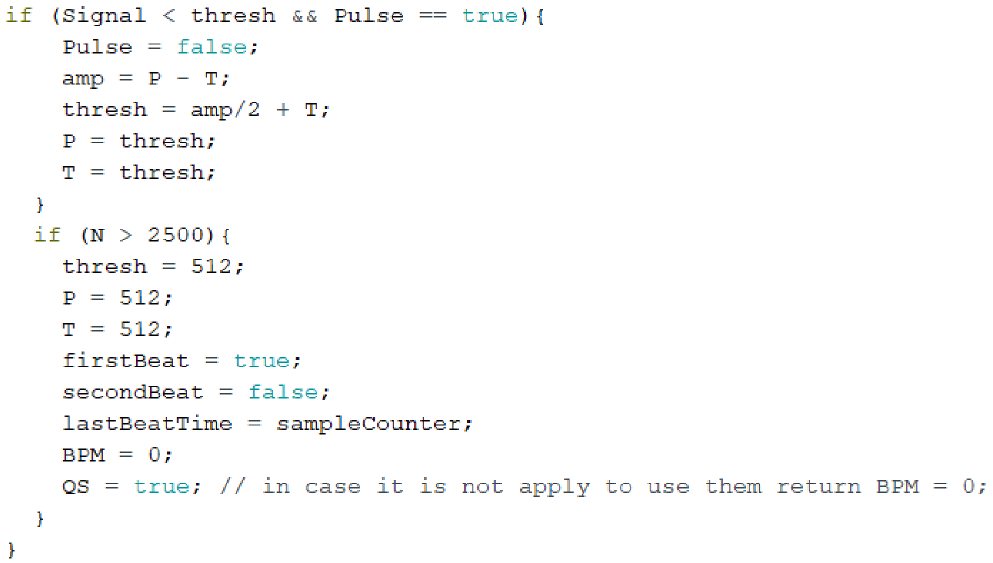
```
////////////////////////////////////////////////////
if (N > 250){
if ( (Signal > thresh) && (Pulse == false) && (N > ((IBI/5)*4) )){
    Pulse = true;
    IBI = sampleCounter - lastBeatTime;
    lastBeatTime = sampleCounter;
    if(secondBeat){
      secondBeat = false;
      for(int i=0; i<=9; i++){
        rate[i] = IBI;
      }
    }
    if(firstBeat){
        firstBeat = false;
        secondBeat = true;
        sei();
        return;
     }
      word  runningTotal = 0; // calculating BPM
    for(int i=0; i<=8; i++){
      rate[i] = rate[i+1];
      runningTotal += rate[i];
    }
    rate[9] = IBI;
    runningTotal += rate[9];
    runningTotal /= 10;
    BPM = 60000/runningTotal;
    QS = true;
    }
}
```

Figure 47 ISR function (part 2)

If N was higher than a specific value, in this case was 250ms, the program run next steps of beat detection. 250 ms was coinciding with 240 BPM which was quite high for normal the heart rate. If there was no pulse at that moment and signal was greater than thresh value, program received a heart pulse.

Two heartbeat sounds were involved for a heartbeat. If the pulse was at the first sound then program marked firstBeat to false, secondBeat to true and return. Otherwise, secondBeat was unflagged and program started feeding value to rate[] array, skipped to beat calculation part which was marked in figure 47. QS value was set to true, set a flag for program to pop up value of BPM after computing. The calculation process was mentioned in 5.2 Heartbeat calculation. The runningTotal is sum of all current BPM value and number of BPM samples in this project was 9.

Figure 48 shows implementation after sending back a BPM value.



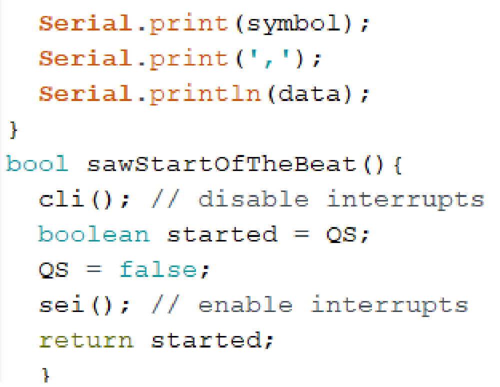
```
if (Signal < thresh && Pulse == true){
    Pulse = false;
    amp = P - T;
    thresh = amp/2 + T;
    P = thresh;
    T = thresh;
  }
  if (N > 2500){
    thresh = 512;
    P = 512;
    T = 512;
    firstBeat = true;
    secondBeat = false;
    lastBeatTime = sampleCounter;
    BPM = 0;
    QS = true; // in case it is not apply to use them return BPM = 0;
  }
}
```

Figure 48 ISR function (part 3)

When the Pulse status was still true and Signal value started falling, it was a sign for the end of a pulse. Pulse status was set to false, other values such as amp, thresh, P , T were updated.

In case the sensor did not work or after long period without pulse(2500ms), values were initialized again.

Supporting functions are shown in figure 49.



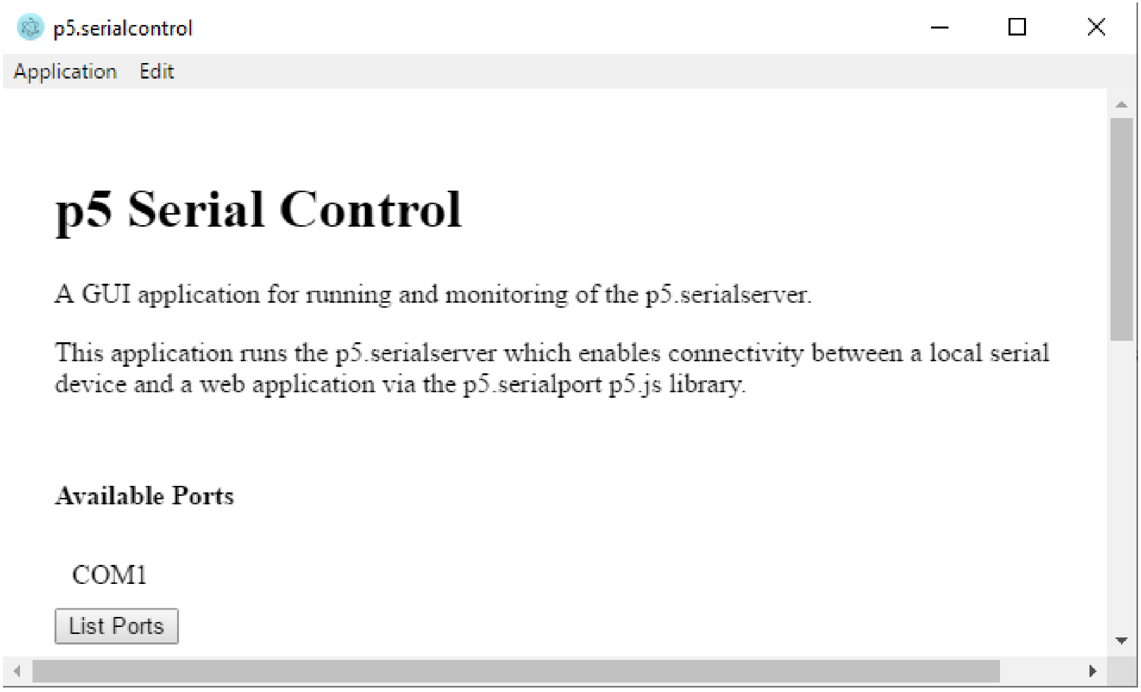
```
void sendDataToProcessing(char symbol, int data ){
  Serial.print(symbol);
  Serial.print(',');
  Serial.println(data);
}
bool sawStartOfTheBeat(){
  cli(); // disable interrupts
  boolean started = QS;
  QS = false;
  sei(); // enable interrupts
  return started;
  }
```

Figure 49 Supporting function

Data sent to server must distinguish between Signal value and BPM value hence sendDataToProcessing() function was used to form data package as: [S,'signal value'] or [B,'BPM value']. These values will be split into 2 parts: data type and data value in server to continue processing.
Function sawStartOfTheBeat() was prepared to make a tiny time gap to disable interrupt, set a flag started to send BPM and then activate interrupts again.

5.5.2 Server

One of the purposes was to visualize the data on screen and allow other device to access the view. Arduino supported only simple plotter and monitor which were not enough for project's needs. Data must be sent to server although Arduino was not able to connect to server by default. In this case, a bridge program was needed to set communication between Arduino and website.
There were several solutions available however to simplify the process, p5.serialport library was chosen as it was a part of p5.js which will be discussed later in data visualizing. The p5.serialport comes in two flavors: a simple application for all level users or Node.js based WebSocket server for advanced user who needs deep customization. In this project, the application was enough.
Figure 50 illustrate the GUI of the bridge program.



p5 Serial Control

A GUI application for running and monitoring of the p5.serialserver.

This application runs the p5.serialserver which enables connectivity between a local serial device and a web application via the p5.serialport p5.js library.

Available Ports

COM1

Figure 50 p5.serialcontrol application (p5.serialport, n.d)

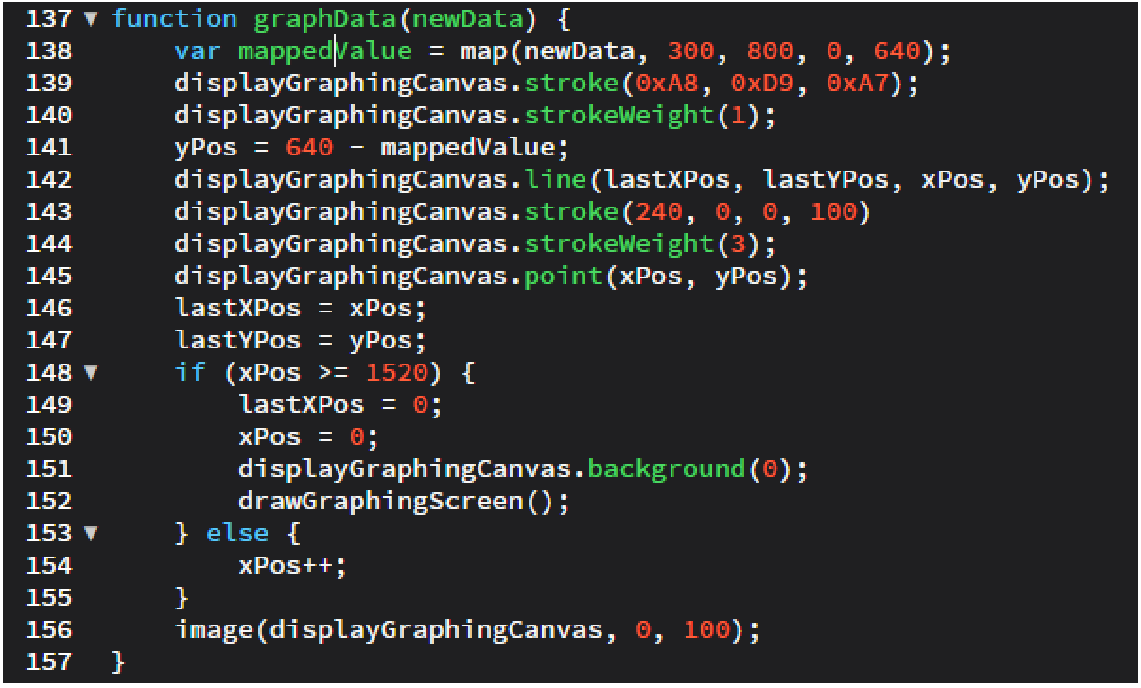
Since the program was activated, Arduino and website were able to communicate using function in p5.serialport library.

5.5.3 Data visualizing

For data visualizing, p5.js was the chosen solution. P5.js is a JavaScript library started with the original goal of Processing to make coding accessible for artists, designer, educator and beginners. This library has a full set of drawing functionality.(p5.js introduction, n.d)

Data was demonstrated on a webpage so essential files for data visualization must be JavaScript files(.js), HTML file(.html), CSS file(.css) and images if needed.

In general, the web interface was used as a canvas to draw data continuously sent from Arduino to server. One of the most important factors of the visualizing process was the the heart signal line which should obtained the form of normal phonocardiograph. Figure 51 shows the function used to graph phono cardio-line.

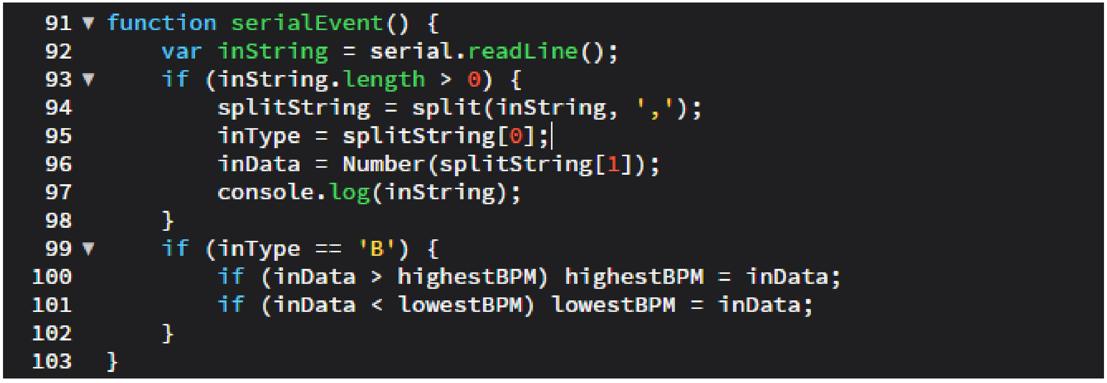


```
function graphData(newData) {
    var mappedValue = map(newData, 300, 800, 0, 640);
    displayGraphingCanvas.stroke(0xA8, 0xD9, 0xA7);
    displayGraphingCanvas.strokeWeight(1);
    yPos = 640 - mappedValue;
    displayGraphingCanvas.line(lastXPos, lastYPos, xPos, yPos);
    displayGraphingCanvas.stroke(240, 0, 0, 100)
    displayGraphingCanvas.strokeWeight(3);
    displayGraphingCanvas.point(xPos, yPos);
    lastXPos = xPos;
    lastYPos = yPos;
    if (xPos >= 1520) {
        lastXPos = 0;
        xPos = 0;
        displayGraphingCanvas.background(0);
        drawGraphingScreen();
    } else {
        xPos++;
    }
    image(displayGraphingCanvas, 0, 100);
}
```

Figure 51 Phonocardio-line graphing

According to figure 51, fed newData was mapped to the size of canvas and connected to the previous mapped value to achieve a continuous line. When xPos reached the border of the canvas, function reset background to draw a new page of graph.

Figure 52 explains where the newData came from.

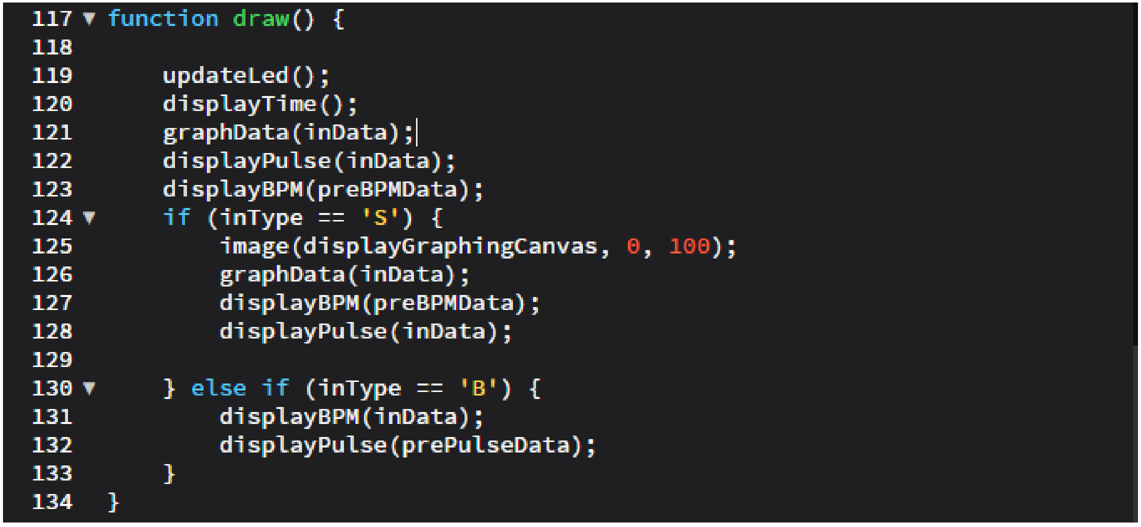


```
function serialEvent() {
    var inString = serial.readLine();
    if (inString.length > 0) {
        splitString = split(inString, ',');
        inType = splitString[0];
        inData = Number(splitString[1]);
        console.log(inString);
    }
    if (inType == 'B') {
        if (inData > highestBPM) highestBPM = inData;
        if (inData < lowestBPM) lowestBPM = inData;
    }
}
```

Figure 52 Data parsing

Recall the helping function in Arduino code, incoming data from Arduino was marked with ‘B,’ or ‘S,’ at the begin of data package. Every time there was data from Arduino, function serialEvent() will be called. As can be seen from figure 52, data was split with character ’,’ as the delim. Therefore, expected result string were letter ’B’ or letter ’S’ and a number. These values were stored in splitString[0] and splitString[1] then in inType and inData.

Figure 53 shows the main function in p5.js : draw().

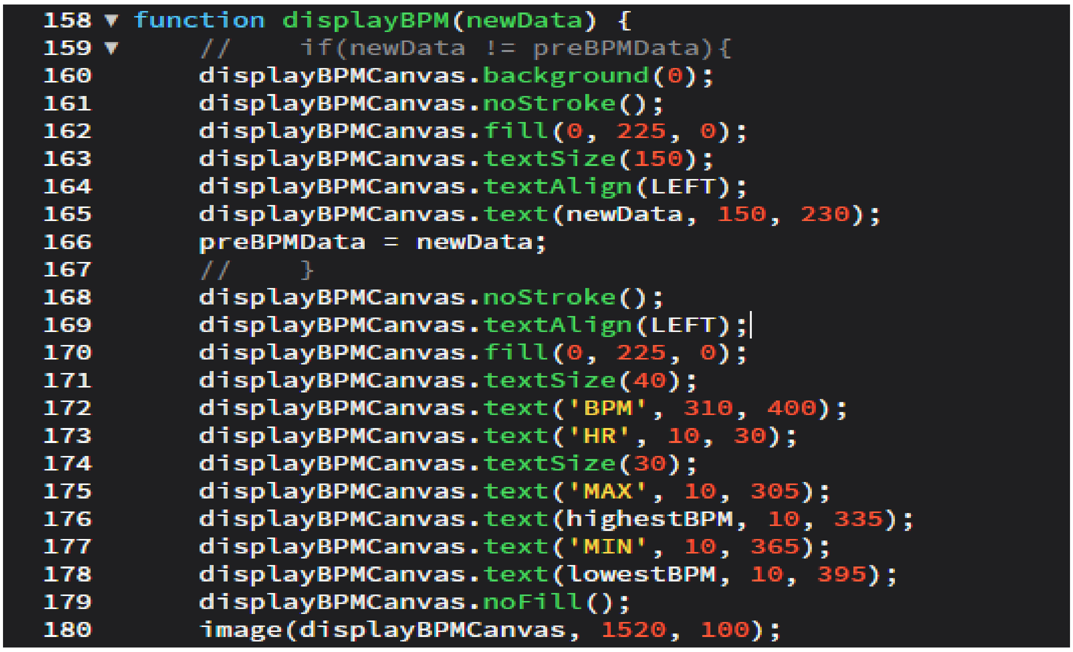


```
function draw() {

    updateLed();
    displayTime();
    graphData(inData);
    displayPulse(inData);
    displayBPM(preBPMData);
    if (inType == 'S') {
        image(displayGraphingCanvas, 0, 100);
        graphData(inData);
        displayBPM(preBPMData);
        displayPulse(inData);

    } else if (inType == 'B') {
        displayBPM(inData);
        displayPulse(prePulseData);
    }
}
```

Figure 53 Draw function

This function was called repeatedly if the web was still on. From figure 53, it was sensible that each element of the visualizing data process was wrapped in a function to be updated in the draw() function for example: time, pulse, BPM, … inType variable introduced before was used to distinguish the type of incoming data for draw function to update new values.

The complete sketches will be attached in appendix section. Figure 53 below presents one of the functions used in draw(): displayBPM().

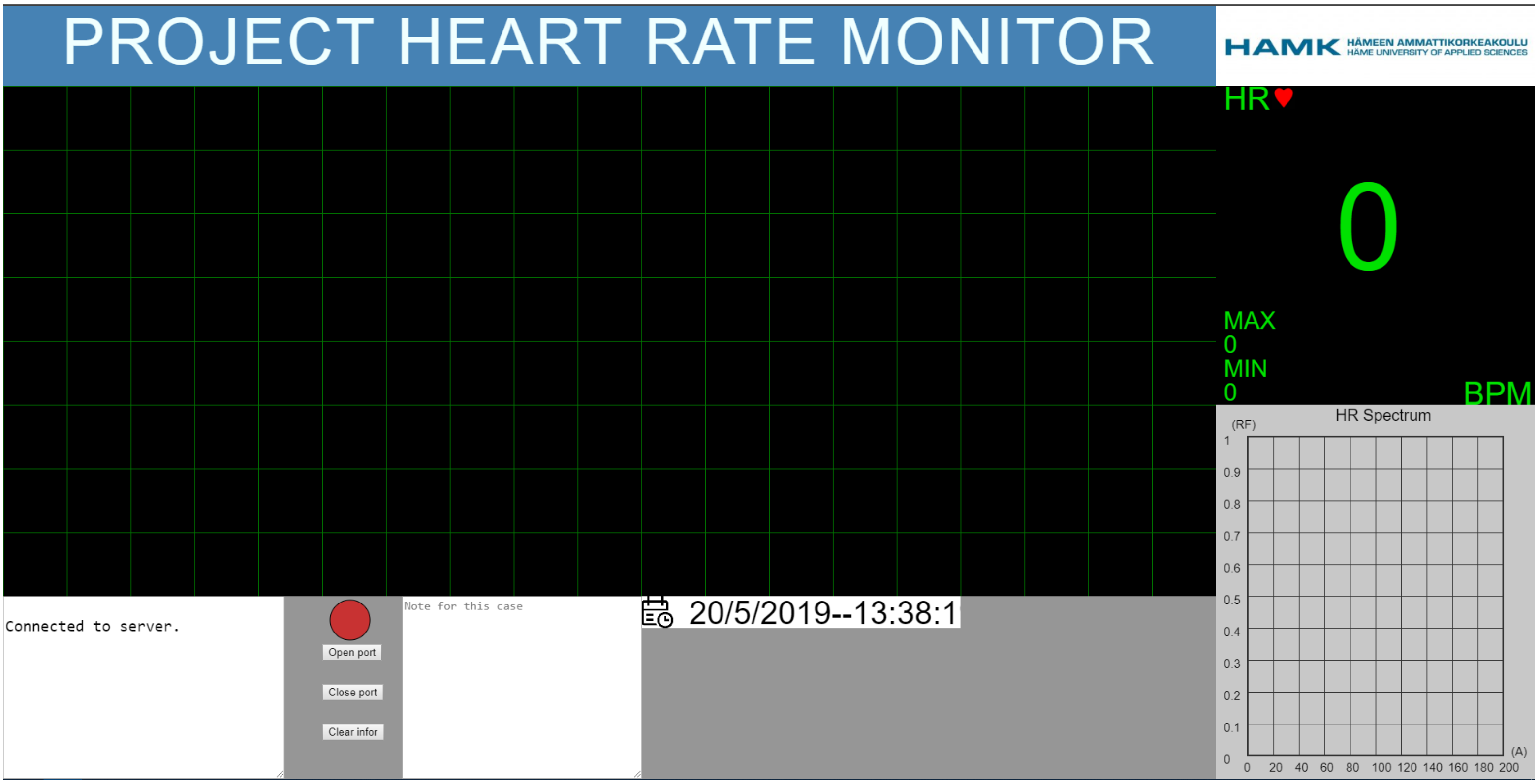


```
function displayBPM(newData) {
//    if(newData != preBPMData){
    displayBPMCanvas.background(0);
    displayBPMCanvas.noStroke();
    displayBPMCanvas.fill(0, 225, 0);
    displayBPMCanvas.textSize(150);
    displayBPMCanvas.textAlign(LEFT);
    displayBPMCanvas.text(newData, 150, 230);
    preBPMData = newData;
//    }
    displayBPMCanvas.noStroke();
    displayBPMCanvas.textAlign(LEFT);
    displayBPMCanvas.fill(0, 225, 0);
    displayBPMCanvas.textSize(40);
    displayBPMCanvas.text('BPM', 310, 400);
    displayBPMCanvas.text('HR', 10, 30);
    displayBPMCanvas.textSize(30);
    displayBPMCanvas.text('MAX', 10, 305);
    displayBPMCanvas.text(highestBPM, 10, 335);
    displayBPMCanvas.text('MIN', 10, 365);
    displayBPMCanvas.text(lowestBPM, 10, 395);
    displayBPMCanvas.noFill();
    image(displayBPMCanvas, 1520, 100);
```

Figure 54 displayBPM function

6 RESULTS

For the electrical design, the result was demonstrated in am oscilloscope as can be seen from figures 30, 40 and 42. The circuit was able to filter and amplify the Visuheart signal for the speaker as well as Arduino. The data visualizing result is demonstrated in this section.



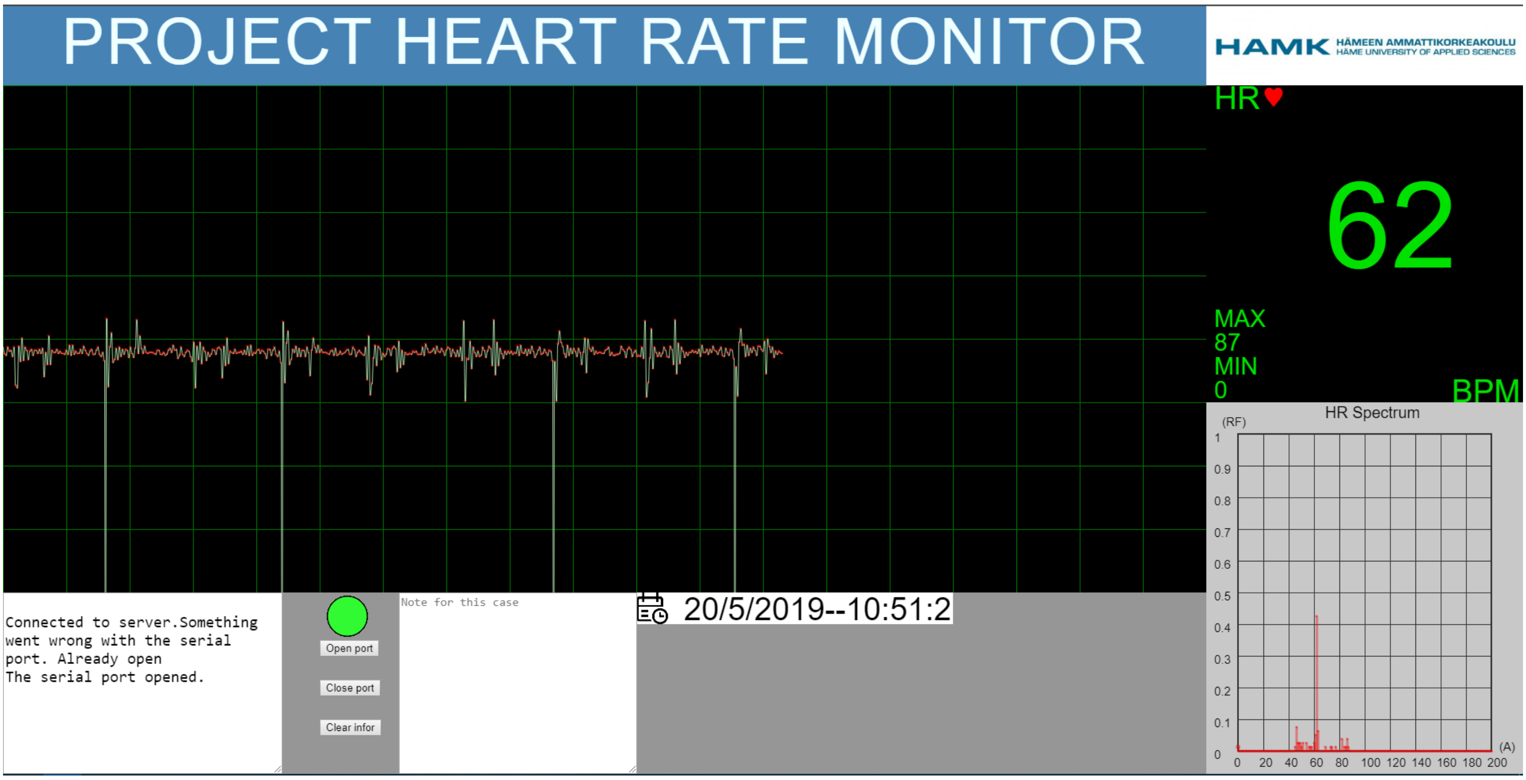
```
Connected to server.
```

Figure 55 Web interface no signal

Figure 55 demonstrates the de-activated website. The components of this user interface included:

- Information box at the left bottom corner showing a message from the system.
- Series of buttons: Open port/close port to enable/disable data receiving process of the website, Clear Infor to clean the information box.
- Led status turns green in graphing process and red otherwise.
- Grid screen for a phonocardiograph.
- Hear rate area on the right side of the screen with: BPM value, min and max values.
- Pulse area showing the amplitude of the signal.
- Time displaying area
- Note area at the right bottom corner for quick note for specific cases.
- Relative frequency (HR spectrum) area.

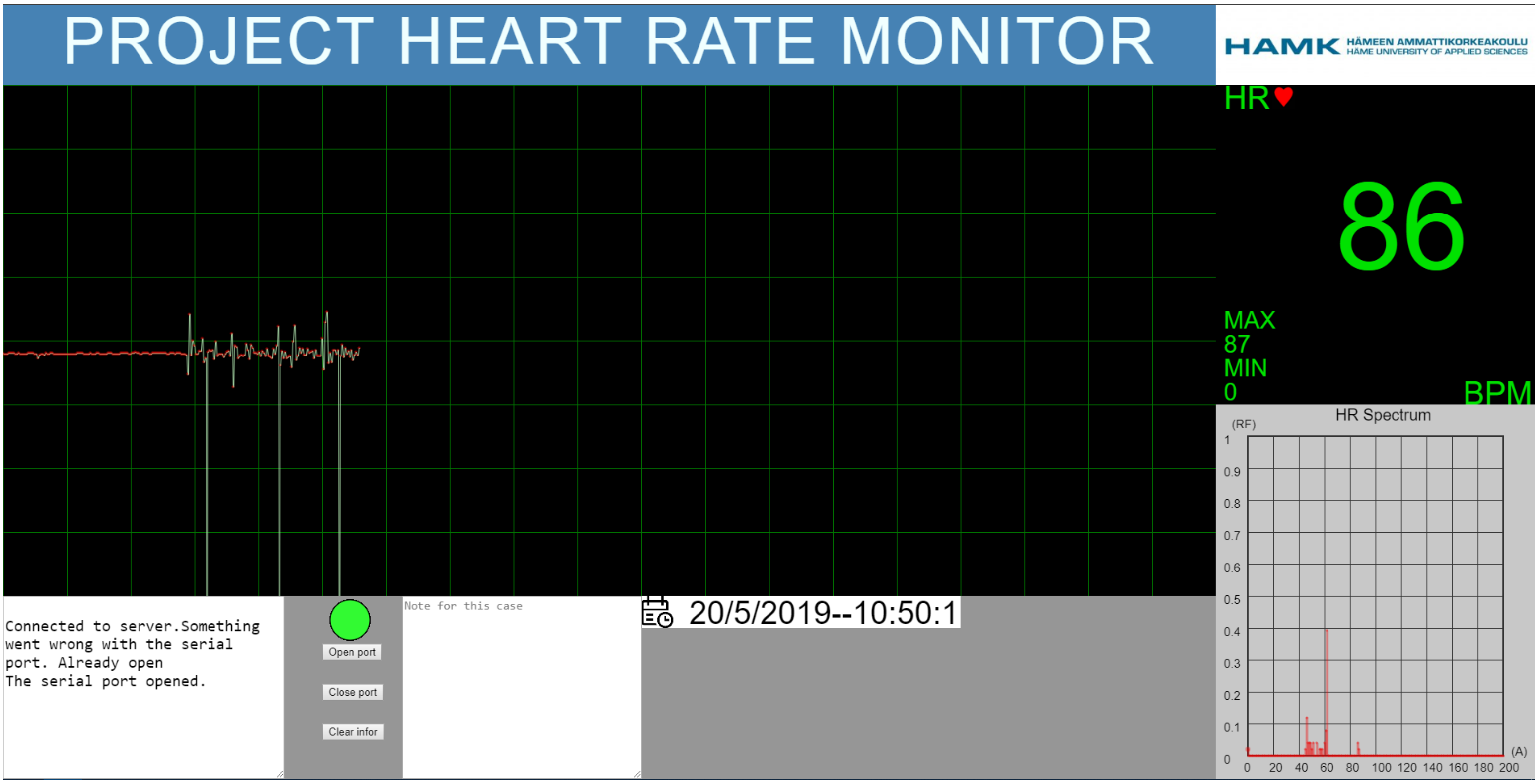
Figures 56 and 57 show the data visualization in different cases.



```
Connected to server.Something
went wrong with the serial
port. Already open
The serial port opened.
```

20/5/2019--10:51:2

Figure 56 Data visualization on web app (1)



```
Connected to server.Something
went wrong with the serial
port. Already open
The serial port opened.
```

20/5/2019--10:50:1

Figure 57 Data visualization on web app (2)

As can be seen from these figures, the program reacted sensitively with the change of the heartbeat sound. In Figure 57, the BPM value was higher corresponding to more heartbeat sounds can be detected from the graph in comparation to figure 56. The actual BPMs were 60 and 86 while the calculated values were 62 and 88. The information box and the note area and the buttons were implemented perfectly in these attempts. The HR spectrum area demonstrated the distribution of the heart rate during the measurement process.

7 LIMITATIONS AND EXPANSION POSSIBILITIES

7.1 Limitations

The limitations of the project included the following issues:

- The electric design only worked perfectly for simulation.
- The algorithm to smooth the signal was enough for proper visualization but not for advance requirements.
- The sensing components were not optimized. Figure 58 presents the current sensor design.

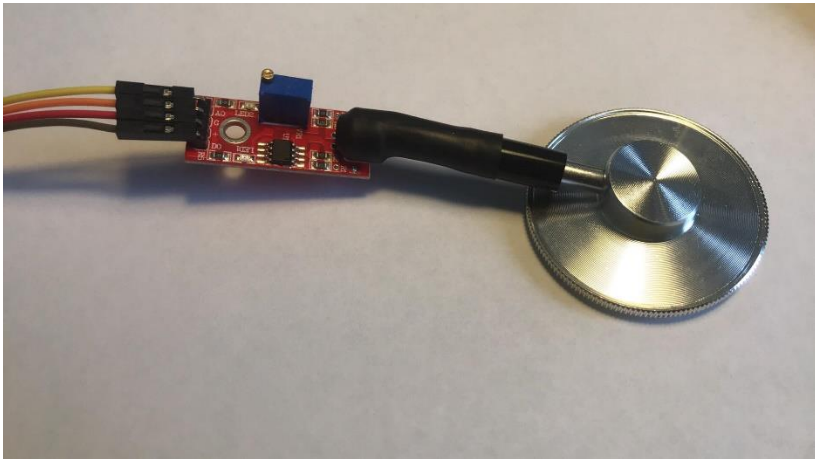


Figure 56 Current sensing components

7.2 Expansion possibilities

The project can be updated with more useful improvements to achieve the best performance. Future work can be divided into two categories: hardware and performance optimizing.

For the hardware, the circuit design could beat implemented with a higher efficiency and precision if there were more professional research and testing. Electronic components can be rearranged neatly in a small size printed circuit board to minimize the size and weight of the device which was the key factor in evaluating a portable device. There should be an elastic belt to locate the sensor firmly on the target’s body and a 3D printed housing component to fix the structure of the microcontroller and the circuit together.

Figure 57 Elastic belt to locate the sensor

For performance optimization, a more effective algorithm can be applied to achieve smoother signal, one of which was wave-let transformation. For the website, a function to record all the values on the screen should be useful compared to the current manual screen shoot method. The website can be more responsive and equipped with customized back-end service. Finally, data can be stored in a cloud service for further retrieval or processing.

8 CONCLUSION

In this thesis, some related concepts were defined such as cardiovascular system, the heart rate measurement methods, amplifier circuit and signal filters. The author also implemented a protype to experiment the design. During the period for this thesis project, professional skills such as circuit designing, signal processing and data visualizing were gained. In addition, both theoretical and empirical work played a big role in building deeper acknowledgment of author in this field.

REFERENCES

Mahmoodabadi S. Z (MSc), Ahmadian, Abolhasani (2005),
ECG feature extraction using Daubechies wavelets.

Retrieved May 28, 2019, from:
https://www.researchgate.net/publication/6482990_Photoplethysmography_and_it_application_in_clinical_physiological_measurement

Dog cardiovascular(n.d). Retrieved May 28, 2019, from vetmed.wsu.edu:
https://www.vetmed.wsu.edu/outreach/Pet-Health-Topics/categories/cat-and-dog-anatomy/cardiovascular-system-of-the-dog

Walker, HK .1990. Clinical Methods: The History, Physical, and Laboratory Examinations. 3rd ed. Boston: Butterworths. Chapter 30.

Cunningham, SM. (2018), Introduction to Heart and Blood Vessel Disorders in Dogs. Retrieved May 28,2019, from: https://www.msdvetmanual.com/dog-owners/heart-and-blood-vessel-disorders-of-dogs/introduction-to-heart-and-blood-vessel-disorders-in-dogs

Amiri, MA. (2013), Heart Sound Analysis for Diagnosis of Heart Diseases in Newborns. Retrieved May 28, 2019, from:
https://www.researchgate.net/publication/210290203_Phonocardiography_Signal_Processing

Allen(2007), Photoplethysmography and its application in clinical physiological measurement. Retrieved May 29, 2019, from:
https://www.researchgate.net/publication/210290203_Phonocardiography_Signal_Processing

Abbas, Bassam.(2009), Phonocardiography Signal Processing. Retrieved May 29, 2019, from:

https://www.researchgate.net/publication/210290203_Phonocardiography_Signal_Processing

Ahlstrom, Höglund,Hult (Jan 2008), Distinguishing Innocent Murmurs from Murmurs caused by Aortic Stenosis by Recurrence Quantification Analysis. Retrieved May 29, 2019, from researchgate.net:

https://www.researchgate.net/publication/242416470_Distinguishing_Innocent_Murmurs_from_Murmurs_caused_by_Aortic_Stenosis_by_Recurrence_Quantification_Analysis

Measurement system (n.d). Kumar, S. Medical Application. Retrieved June 1,2019, from eu.mouser.com:

https://eu.mouser.com/applications/medical_efficient_monitoring/

Stethoscope(n.d). Retrieved June 1,2019, from nurselk.com:

https://www.nurselk.com/2016/09/parts-of-stethoscope.html

Nelson,G (Oct,2015), Stethoscope Design for Auscultation in High Noise Environments. Retrieved May 26,2019 from:

https://conservancy.umn.edu/handle/11299/183345

ECG (n.d). Retrieved June 1,2019, from amazon:

https://www.amazon.co.uk/Denshine-Portable-6-channel-Electrocardiograph-Software/dp/B00RB80YFA

ECG signal (n.d). Retrieved June 1,2019, from rohde-schwarz.com:

https://www.rohde-schwarz.com/in/applications/capturing-small-ecg-signals-in-medical-applications-application-card_56279-152385.html

Amplifier (n.d). Amplifier using op-amp. Retrieved June 1,2019, from electronic-tutorials.ws:

https://www.electronics-tutorials.ws/opamp/opamp_3.html

Filter types (n.d). An Introduction to Filters. Retrieved June 1,2019, from allaboutcircuit.com:

https://www.allaboutcircuits.com/technical-articles/an-introduction-to-filters/

Passive LPF (n.d). Passive Low Pass Filter. Retrieved June 1,2019, from electronic-tutorials.com:

https://www.electronics-tutorials.ws/filter/filter_2.html

Active LPF (n.d). Active Low Pass Filter. Retrieved June 1,2019, from electronic-tutorials.com:

https://www.electronics-tutorials.ws/filter/filter_5.html

Texas Instruments(September 2002). Analysis of the Sallen-Key Architecture

Vysakh (n.d). Retrieve June 1,2019, from circuittodays:

http://www.circuitstoday.com/basics-of-microcontrollers

Ross Shannon(2012,August 21). Retrieved June 1,2019, from yourhtmlsource:

https://www.yourhtmlsource.com/starthere/whatishtml.html

MDN web docs CSS(Mar 23, 2019) How CSS works? Retrieved June 1,2019, from developer.mozilla.org:

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/How_CSS_works

MDN web docs Javascript(Apr 23, 2019) What is JavaScript? Retrieved June 1,2019, from developer.mozilla.org:

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Sparkfun microphone(n.d). Retrieved June 2,2019, from sparkfun:

https://www.sparkfun.com/products/8635

Arduino intro(n.d). Retrieved June 2,2019, from store.arduio.cc:

https://store.arduino.cc/arduino-uno-rev3

p5.js introduction (n.d). Retrieved May 25,2019, from p5.org

p5.serialport (n.d). Retrieved May 25,2019, from github:

https://github.com/vanevery/p5.serialport