ASHESI UNIVERSITY

THE DESIGN OF A LOW-COST PULSE SENSOR FOR REMOTE HEALTHCARE

CAPSTONE PROJECT

B.Sc. Electrical & Electronic Engineering

Abena Manubea Darko

2020

ASHESI UNIVERSITY

THE DESIGN OF A LOW-COST PULSE SENSOR FOR REMOTE HEALTHCARE

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Electrical & Electronic Engineering.

Abena Darko

2020

Declaration

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: ……………………………………………………………………………

Candidate's Name: Abena Manubea Darko ……………………………………………………

Date: 29/05/2020 ……………………………………………………………………………

I hereby declare that the preparation and presentation of this capstone were supervised in accordance with the guidelines on the supervision of capstone laid down by Ashesi University.

Supervisor's Signature: ……………………………………………………………………......

Supervisor's Name: ………………………………………………………………………….

Date: ……………………………………………………………………………………………

Acknowledgments

To my supervisor, Dr. Heather Beem, for providing guidance, access to resources, and helpful feedback that pushed me to dig for answers through the stages of this capstone.

Also, to Nicholas Tali for his help with the hardware.

Finally, I am grateful to my friends Isaac Owusu, Dzifa Hodey, Eric Gbekor, Samantha Kyei, Sharon Wiafe, Jeremiah Takyi and Obinna Ogbuagu for their constant support, insight, and motivation which helped me through my lowest points while working on this paper.

Abstract

Ghana has a low doctor to patient ratio, and to help ease pressure from the health care sector, interest in remote healthcare is growing. This paper explores design options for low-cost pulse monitors for use by patients who require access to healthcare but do not have the means to visit regularly for consultation. The product measures their heart rate and sends the data to a remote database using a mobile app implemented with flutter as a gateway. Patients can view their health trends in the app, and health professionals can monitor their patients using the app. The circuit for this pulse sensor was designed in Eagle and implemented on a breadboard to show that the non-invasive technique of acquiring heart rate using infrared light-emitting diodes (LEDs) could be used favorably on persons with darker skin tones. The research project designed an algorithm for the task and compared this with an existing algorithm using PPG and ECG datasets from medical databases as well as PPG signals generated from the pulse sensor. Heart rates detected ranged from 30-300 beats/minute. 5 datasets from the device were statistically compared to PPG signals from a medical database to ascertain their reliability.

Table of Contents

List of Figures

1 Introduction

1.1 Introduction

Ghana's health sector has laudable goals for health care in the country, which include bridging the gap in geographical access to health services, improving the quality of health care as well as increasing efficiency in health system management [1]. However, the inadequacy of the resources available, as is typical in many African countries make attaining those goals difficult. According to the Facts and Figures document put out by the Ghana Health Service (GHS), the total number of hospital beds available in private, public, and religious health facilities across the country as of the year 2016 was 7089, for a population of about 29.5 million people. The doctor to patient ratio, which was at one doctor to 8481 patients [1] is also below the World Health Organization (WHO) recommended ratio of 1:1000 [2], [3]. The same document shows that the hospital admission rate on a national level was 53.9% in 2016. This statistic means if two people went to a hospital, one of them got admitted. The small number of hospital beds available drives the need for more sustainable ways to provide healthcare to members of the general public without increasing pressure on already scarce resources.

Remote healthcare monitoring becomes relevant here. This technological innovation seems to be the future of medicine, where patient vitals can be measured with smart devices and sent across a network to their trusted healthcare professional [4]. This advancement could save patients hundreds of Ghanaian Cedis that would otherwise have been spent in transportation to the hospital or forfeiting some wages to be able to make regular in-person meetings with their doctor.

1.2 Background

Smartphones and watches, seemingly abundant in the Ghanaian technology market, are not accessible to much of the population. These gadgets provide near-accurate measurements of some health parameters such as heart rate, body temperature, which could facilitate the remote healthcare solution. However, since the people who need it most perhaps because of geographical and financial constraints to healthcare centers, a low-cost alternative with accurate sensing ability needs to be available to them to facilitate their healthcare experience. As information technology is improving rapidly in Ghana, incorporating this into healthcare will prove relevant in the long term, although it would take some getting used to.

1.3 Problem Definition

Government hospitals are usually overcrowded, and people spend long, arduous hours in lines, waiting to have their vitals taken before they can even see a doctor. For other people, bills for healthcare while they are admitted to the hospital are too high for them to pay, so they may resort to nursing themselves and compromising on quality healthcare for cheap and subpar solutions. This project is an attempt to bring convenience into the lives of ordinary Ghanaian citizens, while maintaining standards of quality healthcare, by implementing and giving people access to remote healthcare.

The proposed solution to the problem stated above is a lightweight and unobtrusive device that patients can have with them at home to collect their vitals and send it over to health professionals for analysis and consultation. Another facet of this solution is for this same device to be able to monitor sick patients for a duration of time in the comfort of their homes so they can save on in-patient bills.

1.4 The motivation for Project Topic

This capstone project was born out of the researcher's frustrations with the Ghanaian Healthcare system. Hospitals, especially those run by the government, tend to have uncomfortably long lines for taking vitals and for consultations. Many health administrators also pander to social class and ageism while directing patients to doctors. These things typically affect people with non-visible disabilities. Another motivation is the ease of access for the elderly with no caretaker(s) who cannot make the journey to hospitals regularly to have their vitals taken. This project could open the possibility of older people being monitored remotely by the health professionals, and any alarming irregularities picked up by the device and sent to the doctor while alerting any of the elderly patients' preferred family members. The last motivation for this project is to investigate optical sensor bias in their measurements on the skin and how skin tones factor into the results.

1.5 Scope of Work

The scope of this project would be to build a low-cost pulse sensing device for patients who are unable to make frequent visits to the hospital. The complete system would be able to:

- Measure the patient's pulse using non-invasive methods (infrared light)
- Filter and process the signal, paying attention to the impact of different skin tones
- Send and receive data with the combined Bluetooth and mobile application features
- Put time stamps on the data collected for record-keeping

Other features of the system include:

- Communication between the health professional and the user via a web application
- Manual entering of heart-rate data visible on the device's display into the web application by the user

1.5.1 Proposed Chapter Outline

For this research, the plan is to put the introduction, problem definition, motivation, scope, and objectives for the project in Chapter 1. The literature review of remote healthcare, pulse oximetry, signal processing, and wireless communication can be found in Chapter 2. Chapter 3 contains user and system requirements for the device, and the Pugh matrices that informed design decisions as needed. The design process and architecture of the wearable device, implementation of the hardware and software components, and the methodology are documented in Chapter 4. Chapter 5 provides experimental and simulation results of the prototype, as well as documentation about testing strategies. Finally, chapter 6 concludes the project with discussions about limitations and future works.

1.6 Objectives of the Project

The objectives of this research project are as follows:

- To remove the barriers that complicate access to healthcare by introducing a more convenient method.
- To give incapacitated people access to affordable and quality healthcare.
- To create a light, unobtrusive wearable device tailored for a specific patient that measures their temperature and heart rate.
- To show the reliability of the data generated from the pulse sensing device.
- To eliminate the effects of skin tone on the signals generated from the device and extract heart rate from that.

1.7 Expected Outcomes of the Project Work

An expected outcome of this project work is to fabricate a working wearable device that accurately measures the patient's vitals. Accuracy will be determined by comparing results from this device with standard medical data.

Two other outcomes will be smooth communication between the wearable device and a mobile app that the health professional and patient have access to, and adequate signal processing by the microprocessor and MATLAB to establish improved accuracy.

2 Related Work

2.1 Literature review

Considerable work has been done in remote healthcare, and much of the focus has been on reducing doctor-patient ratios across the world and accelerating technological advancement in the field of medicine. A group of researchers did work on improving the quality of life of older adults by equipping these patients with a wearable monitoring system suited to their health needs so that they do not have to spend time in the hospital for monitoring [5]. They developed the INCASA architecture, which is a way for them to help the elderly maintain their autonomy by fitting sensors around their houses to track their daily activities and send an alert to the older peoples' caretaker(s) or doctors when something went amiss. To achieve this, they used Hydra middleware to interface all the physical devices present in the patient's home, with a web service to which reports could be sent to and viewed by relevant parties.

Another group of researchers, hoping to make life easier for the elderly, used telecommunication technology to provide remote healthcare services for patients. They came up with the CARA architecture [6], which comprised systems to record vitals (temperature, mobility, and pulse), allowed the patient to communicate with their caregiver via a web camera and could apply medical rules to the data that is gathered, to provide the best care. One good thing about these projects that could quickly be taken for granted is the fact that the patients do not need to be technologically savvy to use these devices. The engineers and researchers did all the work. The patients just need a quick education when they buy the devices, and they are good to go. Both these projects also had features for alerting patients' caregivers when something was wrong. What the reports failed to mention, however, was how much power the devices used, and the product's exact cost.

Yet another research group did work on this topic using embedded systems and MATLAB [7], unlike the previous two, whose work was based mainly on the Internet of Things. Once the physiological data was gathered from the pulse and temperature sensors, they went through signal processing and then to an ARM processor. That processor was the interface that enabled the signals to be sent to the physician's computer. This research only involved simulations of data using MATLAB. They could not test it on actual people.

There have also been many research projects conducted around the science and engineering of heartbeat monitoring, one of which is the use of the IR sensor to measure the pulse. For this purpose, the sensor works on the principle that once the heart is pumping blood, blood concentration changes, and these changes can produce an electrical voltage or pulse [8]. The patient puts their fingertip on the IR LED, which transmits infrared light to it through the blood in the arteries. The researchers use the Arduino UNO microcontroller to read and process these signals. The average person can read the output from the Arduino on an LCD and know their pulse without consulting a doctor. However, being able to read is not quite the same as knowing what to do with it. The patient still needs a health professional to interpret this data, and this is where the integrated remote healthcare system comes in. Heart-rate data can be collected from patients, processed, and made accessible to the patient and their healthcare professional.

Signal processing algorithms were compared in a paper by Rezaei and Basu to improve the accuracy of the measurements. Since Photo-plethysmography (PPG) heart-rate monitoring is used by many fitness trackers, it needs to consume as little power as possible while providing very accurate data [9]. The algorithms were based on a derivative filter and a high-pass filter each, and in the end, the derivative filter-based algorithm was the more accurate one.

One of the non-intrusive methods that many wearables use to take measurements from the skin is photoplethysmography (PPG). It is a technique used in pulse oximetry, blood pressure measuring, to detect changes in blood volume in the smallest blood vessels [10]. This method has become popular because it is cheap, portable, requires small semiconductor components, and pulse-wave analysis with computers has grown; therefore, using it for clinical applications has become simpler [10]. The paper does not provide insight into the origins of PPG signals. It acknowledges, however, that melanin, in addition to blood volume and the orientation of red blood cells, can affect the amount of light that goes back to the photodetector. It also provides a possible workaround; red infrared light because that has a higher wavelength than other colors, hence it can penetrate the melanin.

While PPG is a convenient method of collecting signals for cardiovascular assessment, it is riddled with noise or interference, so the next paper highlights new ways to filter the PPG signal effectively. MATLAB was used to process the PPG signal obtained from a database. The researchers corrupted this signal themselves and then attempted to retrieve it through three stages of cascaded filtration. They evaluate their results using the Fast Fourier Transform (FFT) compared with root mean square error (RMSE) [11]. This method of filtration worked very well for the researchers.

Accuracy is crucial in wearable technology, especially for a project whose goal is for health professionals to have usable data. In this paper, the authors evaluate the accuracy of an already existing heart rate monitor called PulseOn that employs PPG signals and measures it against an Electrocardiogram (ECG). Their experiment, using ECG as a reference, showed that the PulseOn tracker using PPG detected 99.42% of heartbeats correctly, missed 0.58% of beats, and falsely recorded 1.93% extra beats before analysis and correction. After correction, however, the accuracy increased to 99.57%, missed beats fell to 0.48%, and falsely recorded heartbeats also fell to 0.72% [12]. They were able to obtain these improved levels of accuracy because they used the same correction algorithm that the reference ECG used.

Another matter of accuracy when using PPG is its ability to detect changes in blood volume at different levels of vessel concentration (vascular occlusion) [13]. The study analyzes how well the PPG technique can record heartbeat with its optical method, given the barriers it encounters through the skin, although skin tone was not one of the factors considered.

While measuring heartbeat or oxygen saturation at the wrist area has been the center of many of these articles about wearable technology, data obtained from that area is often distorted and cannot be used reliably in clinical applications. This is as a result of the noise introduced from motion artifacts [14] in the patient's daily life. In their paper Lee et al. test a myriad of spots in and around the wrist area to propose the most effective placement of the infrared lights for suitable PPG signals [15]. These tests were conducted on 5 male people, and the researchers discovered spots on the subjects' wrists that were most suitable for generating PPG signals using green, red, and infrared light. These spots varied slightly from the concentration area of commercial PPG signal detectors.

However, the paper failed to expand on the skin types and tones of their test subjects, especially since they mentioned that light absorption characteristics of tissues [13] could either increase or decrease the signal-to-noise ratio. The study cannot be exhaustive as tests were not conducted on women either, and it is unknown though highly likely that the positions for accurate PPG signal generation will change.

3 Requirements & Architecture

3.1 User Requirements

The user should be able to:

- Move the device around without displacing components
- Login to the web app on any device to enter details about the patient's state when a measurement was taken
- Manually enter heart rate information when wireless communication from device to application fails
- Access history of data
- Communicate with their health care provider through the app
- Afford the device

The healthcare professional should be able to:

- Login to the web application
- Add multiple patients
- View each patient's contact information
- View each patient's present and past health data
- Communicate with each patient via the app by leaving remarks

3.2 System Requirements

The device should:

- Be energy efficient
- Be more affordable than smartwatch alternatives

- Be user-friendly
- Be able to update the database with data gathered
- Accurately measure the heart rate of users regardless of skin tone
- Be able to process the data it collects and present it in a readable and visually pleasing form
- Should be secure and protect user’s health data

Many of these requirements were generated based on a paper by Lymberis, that talked about the impact of telemedicine, and research and development prospects about smart wearables in remote healthcare [16].

3.3 Design Decisions & Pugh Matrices

To ensure that the project achieved desired user and system requirements, the Pugh matrix tool, in addition to the manufacturer’s data sheets, was used. Pugh matrices allow manufacturers to evaluate their material choices or approach by using a selection criterion based on established requirements and research, using a scoring model. The Pugh matrix below gives each design criteria a weight out of 1. The designer then scores each tool or sensor out of 5 based on how well they fit the requirements, and a rating is generated by multiplying the weight with the score. Each tool’s rating is summed, and the one with the biggest total is chosen and used.

The Atmega328p is an 8-bit microcontroller produced by Atmel for use in embedded systems. It has 24 pins and includes timers, input and output pins, ROM, RAM, and CPU built into it.

The MKL25Z128VLK4 is a 32-bit microcontroller based on the ARM Cortex-MO+ architecture and manufactured by NXP. These two microcontrollers will be compared in the Pugh matrix below for a good design decision to be made.

Table 3.1 - Pugh Matrix for Microcontroller unit

Component		ATMEGA328p		MKL25Z128VLK4	
Design Criteria	**Weight**	**Score**	**Rating**	**Score**	**Rating**
1. Ease of use	0.2	3.5	0.7	3	0.6
2. Size	0.3	3	0.9	3.5	1.05
3. Cost	0.25	4	1	3.5	0.875
4. Power Usage	0.15	4	0.6	3	0.45
5. Speed	0.1	2.5	0.25	3.5	0.35
Total	1		3.45		3.325

The Atmega328p highlighted in yellow, satisfactorily fit the criteria because it outweighed the other in the areas with more weight (ease of use, cost, and power usage), so it was selected for the project.

Table 3.2 - Pugh Matrix for Sensors

Component		IR Transmitter and Receiver LED		SEN-11574	
Design Criteria	**Weight**	**Score**	**Rating**	**Score**	**Rating**
1. Accuracy	0.3	3	0.9	3	0.9
2. Size/ portability	0.2	5	1	4.5	0.9
3. Cost	0.35	5	1.75	3	1.05
4. Power Usage	0.15	3	0.45	4	0.6
Total	1	16	4.1	14.5	3.45

The same approach was used to choose the best sensor for the project, and the IR transmitter and receiver LED unit was selected for its affordability, small size, and its lack of a pre-assigned circuit. The last point was relevant to give this research free reign to explore its sensing abilities and manipulate them as necessary.

Table 3.3 - Pugh Matrix for Communication module

		WIFI Module		**Bluetooth Module**	
Design Criteria	**Weight**	**Score**	**Rating**	**Score**	**Rating**
1. Ease of use	0.3	3.5	1.05	3	0.9
2. Size	0.2	3	0.6	4	0.8
3. Cost	0.35	3	1.05	2.5	0.875
4. Power Usage	0.15	2.5	0.375	4	0.6
Total	1		3.075		3.175

The Bluetooth module is a good fit for this project because the pulse sensor would always be close to the patient, so a communication module with a wider range would have been overkill. The Bluetooth module is an access technology that falls under the category of a Personal Area Network (PAN), has a range of up to 10m [17], and works in the frequency band of 2.4GHz [18]. It also uses less power than a Wi-Fi module would use, making it suitable for a low-cost device.

Table 3.4 – Component Specifications and cost

Component	Type	Specification	Price (GHC)
Microcontroller chip	ATMEGA328P	1.8-5.5V, 23 programmable I/O lines, external and internal interrupt, 8-bit counter, 6 PWM channels, 2k bytes SRAM, 1k bytes EEPROM, 20 years data retention at 25º	15.10
Bluetooth Module		2.4GHz, 2-3Mbps modulation, Low power of 1.8V	35.30
Resistors	220, 360Ω, 470Ω, 10kΩ, 50kΩ, 62kΩ, 100kΩ 500kΩ,		4.00
Capacitors	1µF, 0.1µF, 4.7µ		4.03
Infrared LED	Transmitter	940nm wavelength, 1.2-1.3V, 7m transmitting & receiving distance, 30mA max forward current	0.20
	Receiver		0.20

Power Supply	Battery	9V	3/30
Amplifier	LM 358P	3-32V supply voltage, 0-70ºC temperature range, differential voltage amplification	2.00

The total cost of the pulse monitoring device going by the component list above is GHC 63.83, with a cheap battery and GHC 90.83 with the Energizer battery. The Apple Watch, which can monitor heart rate among other things, costs anywhere from $200 to $600 depending on the series, hardware, and communication technology. Approximately between GHC 1,140.00 to GHC 3420.00. Fitbit watches, which are also equipped for heart-rate monitoring and communications, cost between $130 and $350 (GHC 740.00 - GHC 2000.00). These two devices work well but cost more than the target population can afford. The product of this project is for those people and costing less than a GHC 100.00 for long term use; they can afford it.

4 Design & Implementation

4.1 Device Objectives

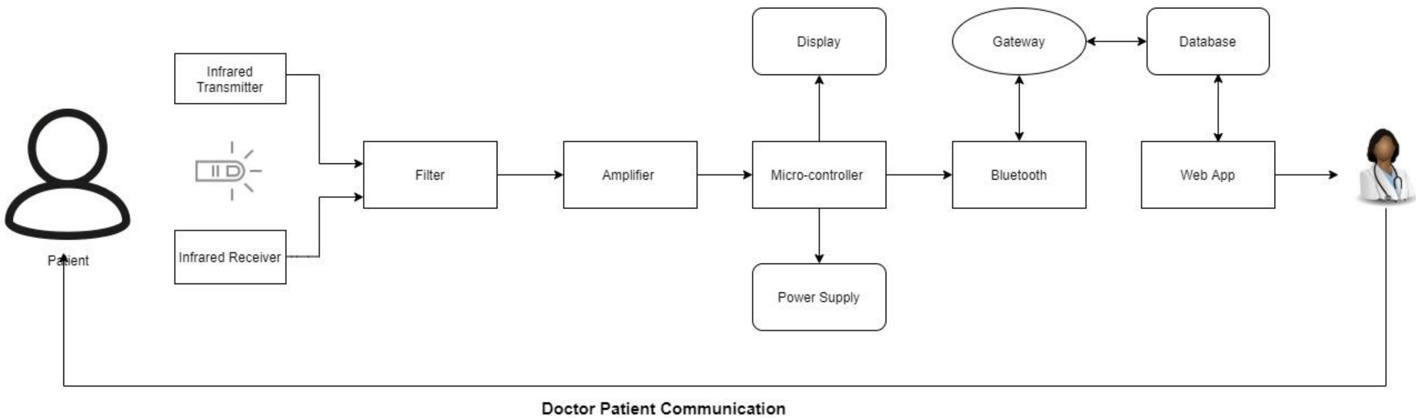

Figure 4-1. Functional diagram of the remote pulse monitoring system

The pulse monitoring device requires the user to put their finger over the infrared receiver and transmitter, for their heart rate to be measured.

Component Relationship

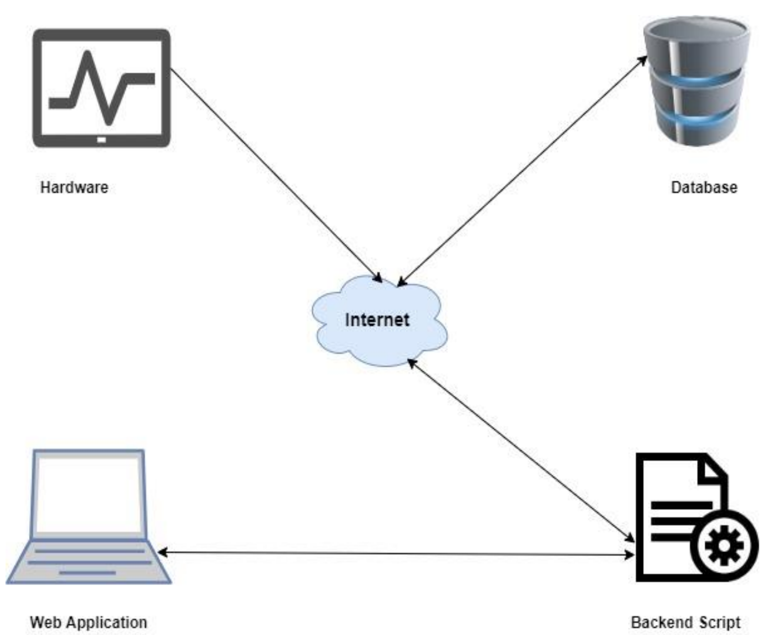

Figure 4-2. Diagram showing the relationship between components

This is a high-level diagram of what the project seeks to achieve. A sensor is built, paying attention to its accuracy and usability on darker skin. Signals are filtered, amplified, and an algorithm is used to extract the pulse rate. The pulse rate information is then transmitted via Bluetooth to a database through a mobile application which the health professional and patients have access to.

4.2 Device Implementation [Hardware]

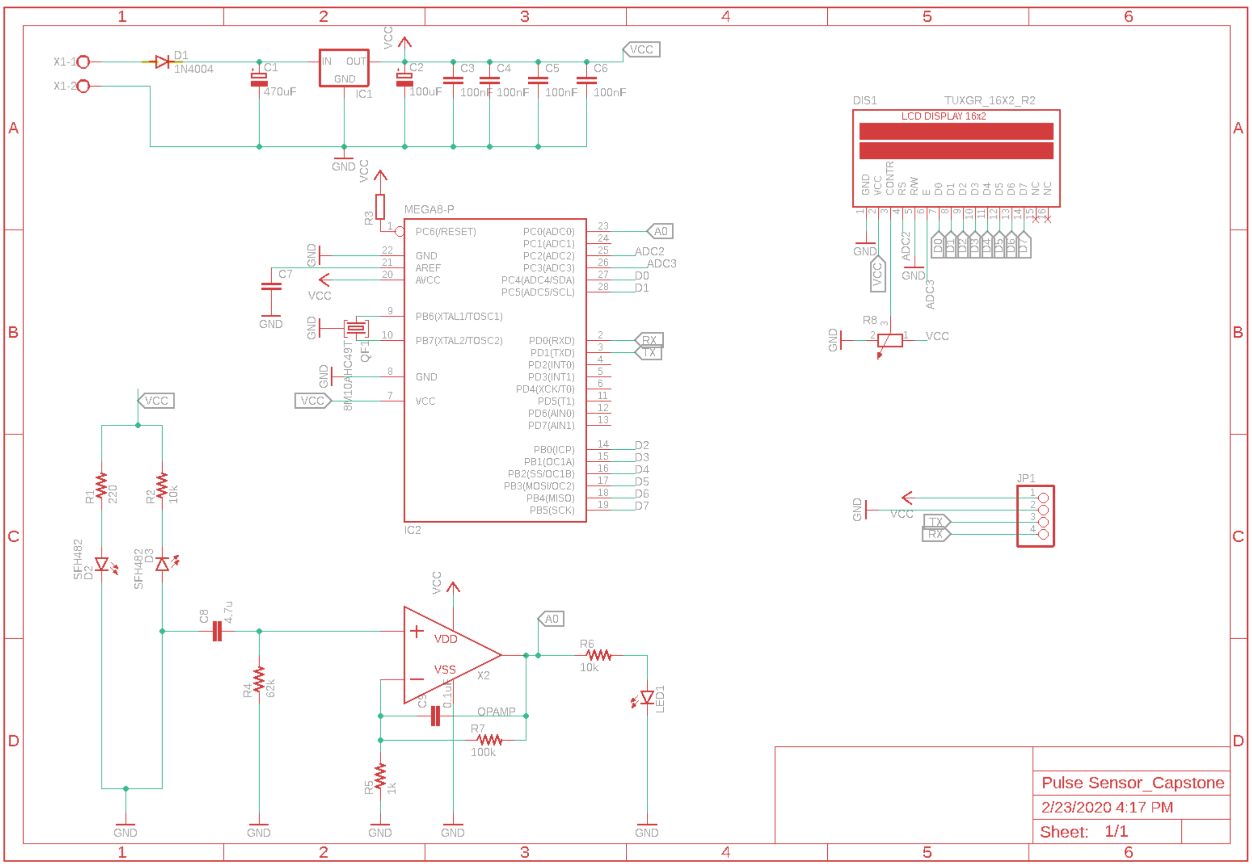

Figure 4-3. Circuit diagram for pulse sensing device

The pulse sensing device is the tool through which a patient's heart rate is determined and then sent to the database. This device includes an infrared led transmitter and receiver pair that operates on a maximum voltage of 5V and current of 30mA, an LM358 operational amplifier (op-amp), and a comparator integrated circuit (IC) with a working voltage of 3-32V. An ATMEGA

328P microcontroller is essential since it interfaces these sensors with the communication device and processing algorithm. This microcontroller chip operates on 5V and has enough memory to store the instructions the components ought to obey, to enable data to be transmitted.

Research shows that resting heart-rate for adults is between 60 – 100 beats per minute [19]. Heart rate, while engaged in physical activity, could go much higher. Using the formula, frequency, $F = \frac{count}{time}$, resting heart rate is between 1-1.6Hz in a minute.

The pulse sensing device being an optical measuring tool could be set up in two different ways to obtain a PPG signal. These modes are illustrated in Figure 4-4 below.

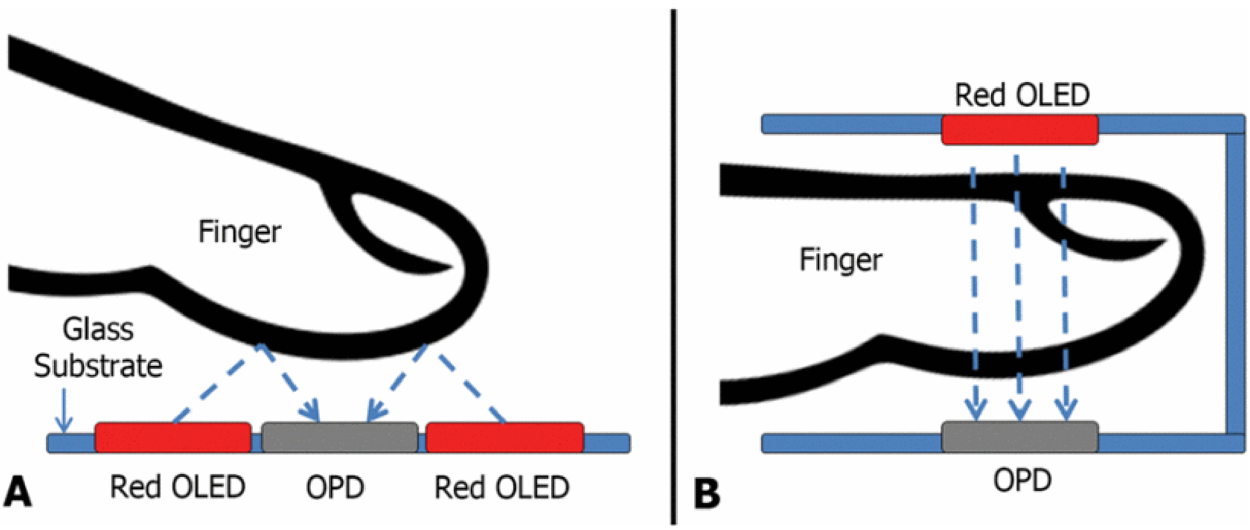

Figure 4-4. Set up to obtain a PPG signal from a pulse sensing device. (A) Reflective mode; (B) Transmissive mode [20].

For this project, the reflective mode of collecting signals was used because it is more stable.

4.3 Device Methodology

For this project, we could not accurately select only one frequency to be the outcome because of component limitations, so a broader range was used, i.e., 0.5-2.3Hz per minute. This range led to cascaded high-pass and low-pass filters in the circuit design. The stopband and passband frequencies were calculated in Hertz using the formula, $f_c = \frac{1}{2\pi RC}$.

Passband frequency, $f_c = \frac{1}{2*\pi*66*10^3*4.7*10^{-6}} = 0.51Hz$ (cutoff frequency for high-pass filter)

Stopband frequency, $f_c = \frac{1}{2*\pi*14.7*10^3*4.7*10^{-6}} = 2.3Hz$ (cutoff frequency for low-pass filter)

These filters, while capturing signals within the range of 0.5-2.3Hz and attenuating signals beyond the range, also allowed electrical noise into the circuitry.

The infrared transmitter (white LED) sends infrared light into the human tissue over it, and the receiver (black LED) collects the reflected infrared light in which the sensor measures by tracking changes in blood volume to give the heart rate of the user. The changes are shown as voltage variations. To measure their heart rate, the patient needs to put a clean finger over the sensors. The presence of the finger causes voltage drops across the receiver bases on blood volume changes. These drops form a signal that is filtered and fed to the non-inverting pin of the op-amp. This input voltage is compared with the reference voltage at the op-amp's non-inverting input. If the input voltage is higher than the reference voltage, the difference between the two voltages is amplified, and this comes to the op-amp's output pin as a near discrete quantity. The op-amp's output is fed to an analog input of the microcontroller and processed. The result, i.e., heart rate in beats per minute (BPM), is displayed on the LCD after it has been counted in the microcontroller.

The heartbeat was calculated in Arduino using a threshold algorithm whose value was determined experimentally and will likely vary from patient to patient but only slightly. The Arduino program works in the following way; when it detects a heartbeat (when the variable containing the PPG signal rises to or above the experimentally determined threshold), a Boolean variable changes its state to true. It sends that beat count to the serial monitor. The Boolean variable then re-sets itself and waits for another beat. If the finger is over the sensor and the analog values as a result of oscillating voltages keep changing to meet and exceed the threshold, a beat count is given. This code can be found in Appendix C.

The resistors used to achieve the desired gain of the amplifier were calculated using the formula, $A_o = 1 + \frac{R_2}{R_1}$.

Amplification gain of the circuit, $A_o = 1 + \frac{510k\Omega}{5.1k\Omega} = 101$

This gain was necessary because the voltage output from the sensor after filtering could go as low as 200mV.

4.4 Database Implementation

An essential facet of this project is storage so that decisions about the patients' health can be made in the future based on past data. For this reason, a database was implemented to transmit the heart rate data continuously. A MySQL database was used because of its ability to handle relational data. Once the data has been collected, the microcontroller sends it via a Bluetooth module to the database using a mobile application as a gateway. Once the data has reached the database, the health professional using the web or mobile app can make requests to retrieve it as and when needed. The user, apart from requesting data from the database, can write to it too via the app.

While the database is necessary for storage, its interactions with the mobile application are not autonomous, and a script needs to be written at the backend to facilitate this. This is known as the

Application Programming Interface (API) and was written using python and the flask framework. A web application that was also started in PHP did not achieve the expected results.

4.5 Application Implementation

Initially, a web application was chosen over a mobile app because data can be stored on the cloud, rather than the client's device, reducing the need for ample storage space. This part of the project was implemented using PHP, an open-source user interface software. However, the Bluetooth medium for transferring data between the device and the database for viewing could not be applied in a web application. A mobile app was created using flutter as a fix for the problem. Flutter is a software development kit for applications created by Google that allows the programmer to use one dart code for apps in multiple mobile platforms like android and iOS. The data is not stored on the phone either. Once the Bluetooth mode is activated, data is sent directly to the database.

The patient registers for the remote healthcare service on the app. Details entered into the form are stored in a database table, where subsequent login details will be matched with to ensure that only the person who registered can sign in and access data.

4.6 Simulation Methodology

Restrictions to movement before the project was complete necessitated a move away from a physical wearable device to a simulation. This simulation was done in MATLAB, using the Simulink tool and MATLAB's inbuilt signal analyzer app.

4.6.1 Digital Filtration

According to literature, the frequency range for heart rate signals is 0.5-5Hz, which allows heart-rate measurements of up to 300 bpm. A Filter Impulse Response (FIR) bandpass filter was designed in Simulink to allow frequencies in a specific range to pass but attenuate signals outside the chosen field. The FIR filter was used because it is often more stable than an Infinite Impulse Response (IIR). Parameters needed to be carefully selected for this filter so that the right frequencies could be analyzed. The passband frequency range was 0.5-5Hz. Since filters are unable to attenuate signals beyond the range of the passband perfectly, a stopband frequency range was required for this, and it was between 0.05-0.1. Therefore, the frequencies filtered were actually between 0.45 and 5.05Hz.

This bandpass filter is stable, has a corresponding linear phase, and uses the equiripple [21] algorithm for its design. The magnitude response for this bandpass filter can be found in figure 4-5 below and the algorithm in appendix

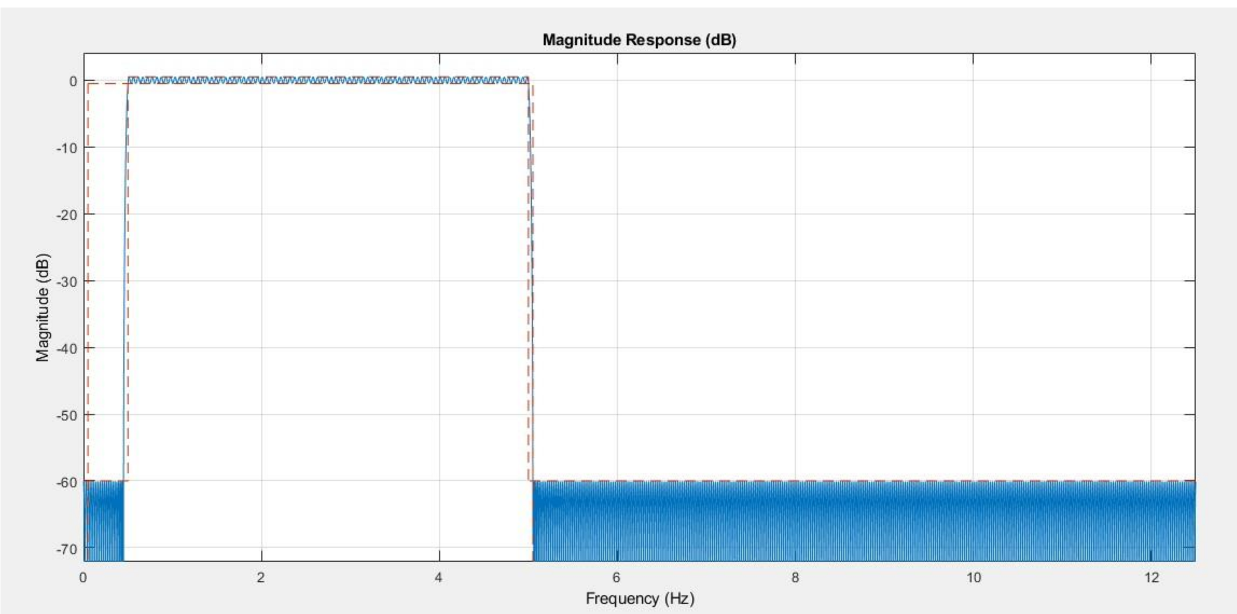

Figure 4-5. Magnitude response of the FIR bandpass filter

5 Testing & Results

This chapter is a discussion of the tests conducted on the device, the simulation, and results obtained from these. The aim is to prove the concepts of a non-invasive way to measure heart rate, ascertain if the objectives listed earlier have been achieved, and the system works as expected.

5.1 Results Obtained from Prototype

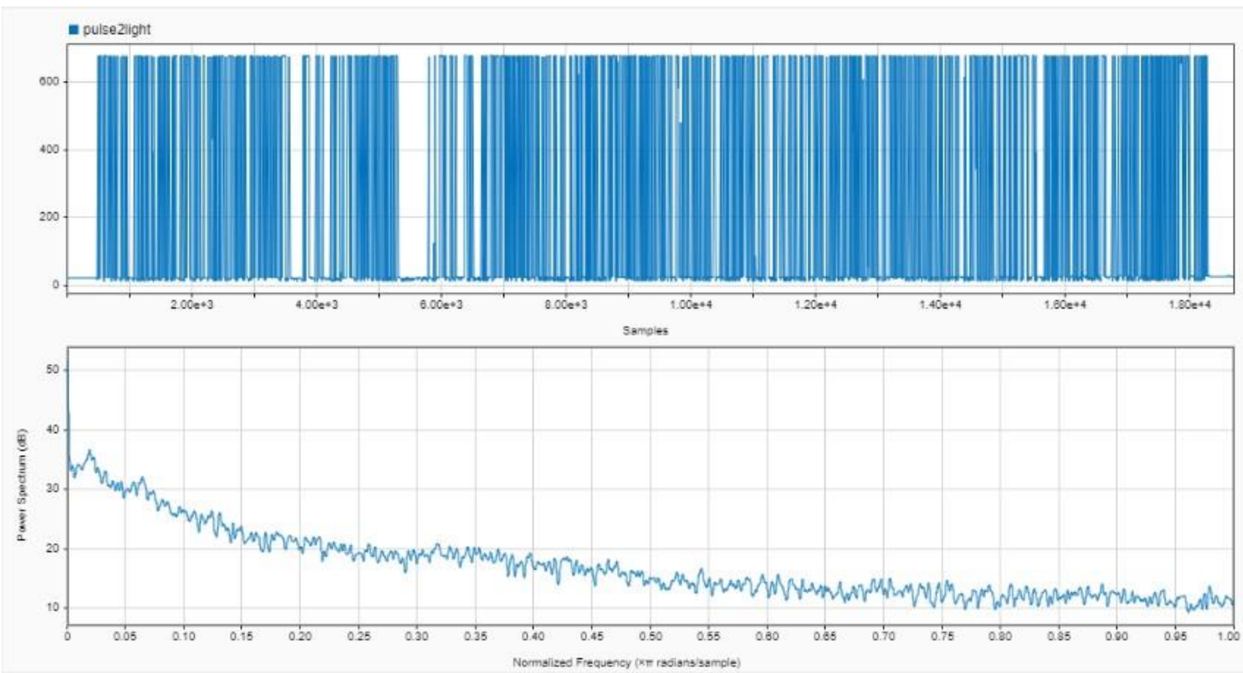

Figure 5-1. Signal output and power spectrum for heart rate taken from a finger

Figure 5-1 above shows the microcontroller output after taking sensor data. The image underneath is the power spectrum of the signal output, whose maximum attenuation is 60dB. The power spectrum was calculated using MATLAB’s inbuilt spectrum functionality in the signal analyzer app. These signal outputs were plotted using MATLAB’s signal analyzer tool and give quite a different power spectrum output than that which used established algorithms.

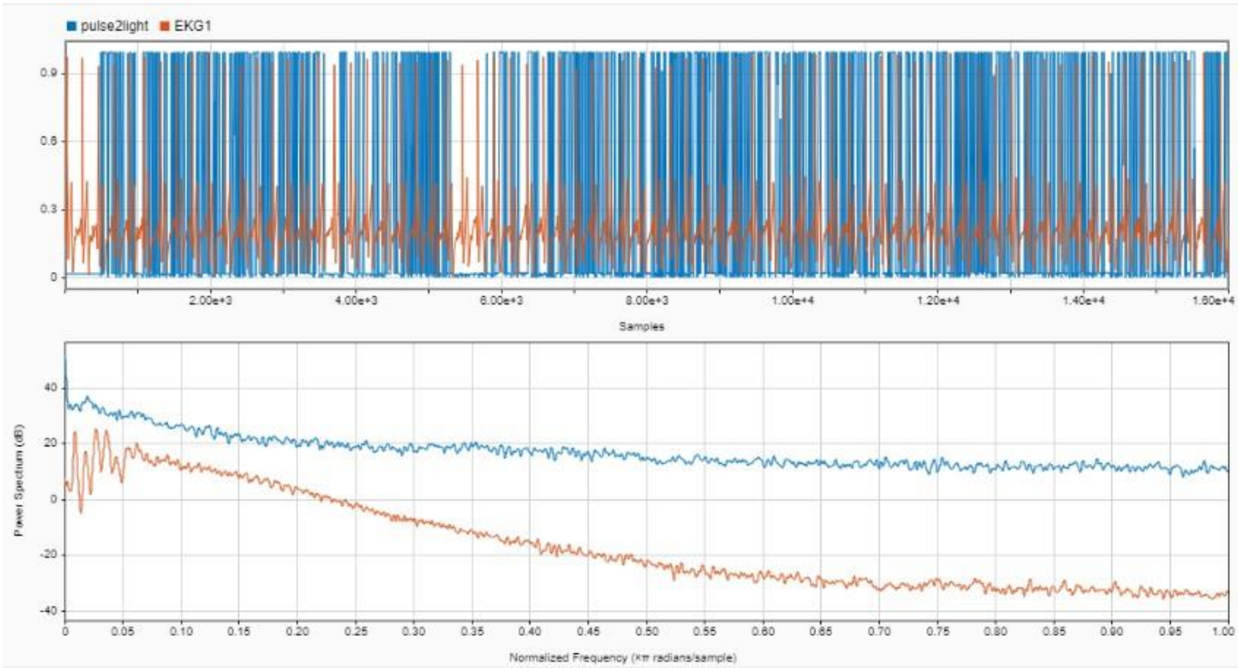

Figure 5-2. Signal output for finger versus ECG signal from the medical database

Figure 5-2 is a plot of a standard ECG signal (in orange) against data from the sensor that was built for this project. There is a distinct difference between the two. The maximum attenuation of the medical ECG is 25dB, whereas the pulse sensor-generated signal is 60dB.

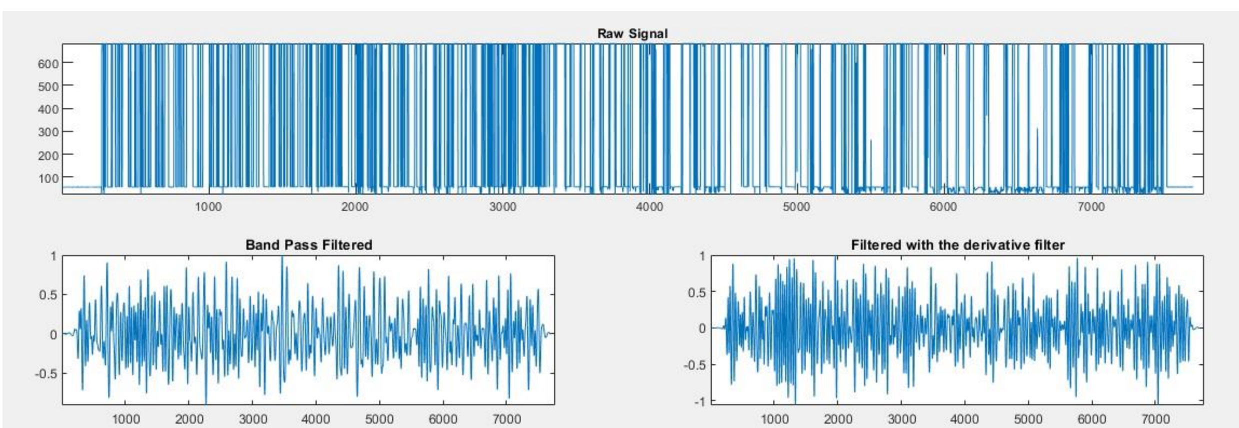

Figure 5-3. Pulse sensor-generated signals before and after each stage of filtration

The sensor data from the device was passed through the Pan-Tompkins algorithm [22], [23]. This is an algorithm used to detect QRS peaks in ECG signals to get pulses. The algorithm works by capping the ECG signal at 50Hz and converting the analog signal into discrete points using an analog to digital converter. The bandpass filter cleans up the noisy signal, and then a derivative for the signal is found [22].

Figure 5-3 shows the pulse sensor data set at the top. The signal at the bottom left came out of a bandpass filter at a sampling rate of 500Hz. The signal on the bottom right came out of a derivative filter.

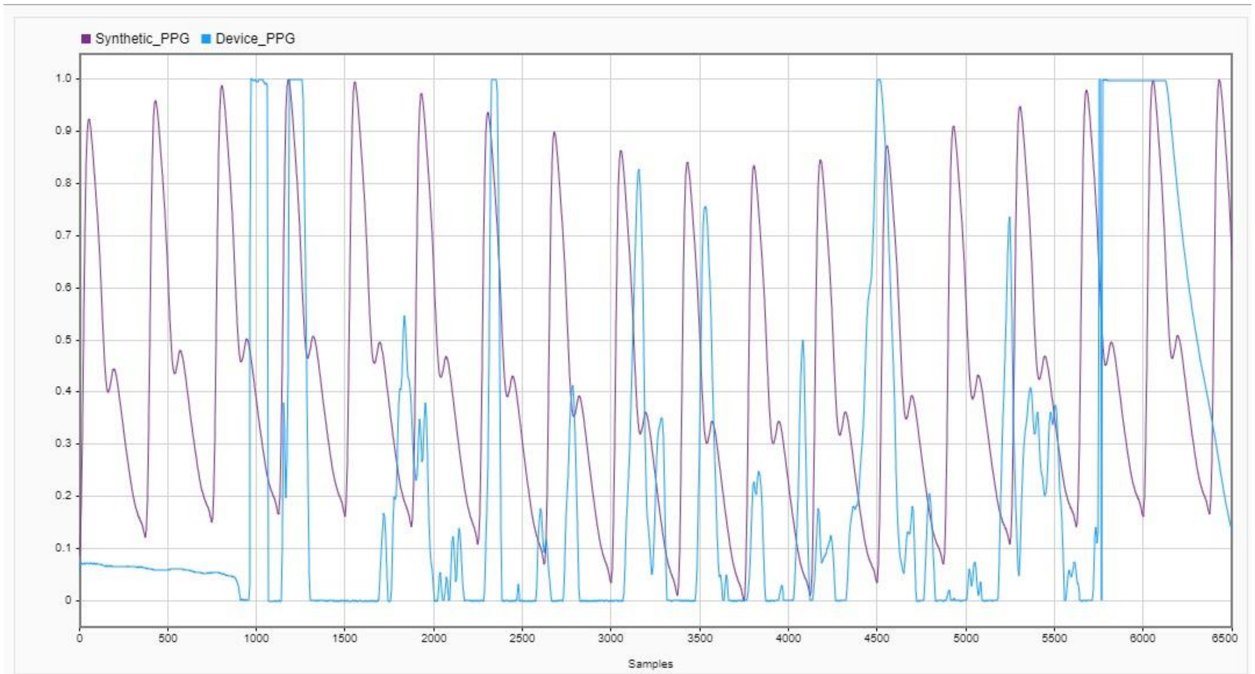

Figure 5-4. Plot of synthetic PPG signal and PPG signal generated from the pulse sensor

Figure 5-4 compares signals of a synthetic PPG to pulse sensor-generated PPG signals. The artificial PPG signals were obtained from an online dataset called Vortal [24], and an algorithm to generate them can be found in the same resource.

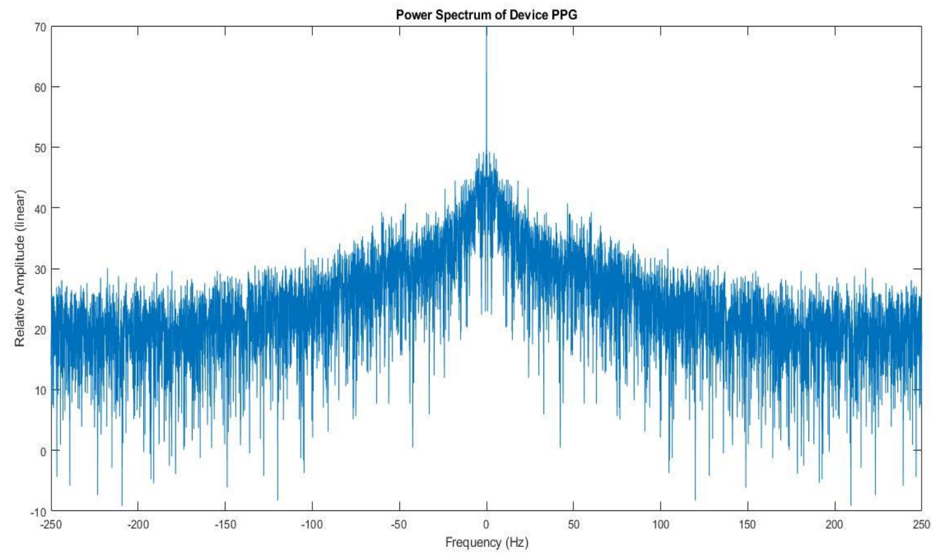

Figure 5-5. Power spectrum of pulse sensor-generated PPG signal

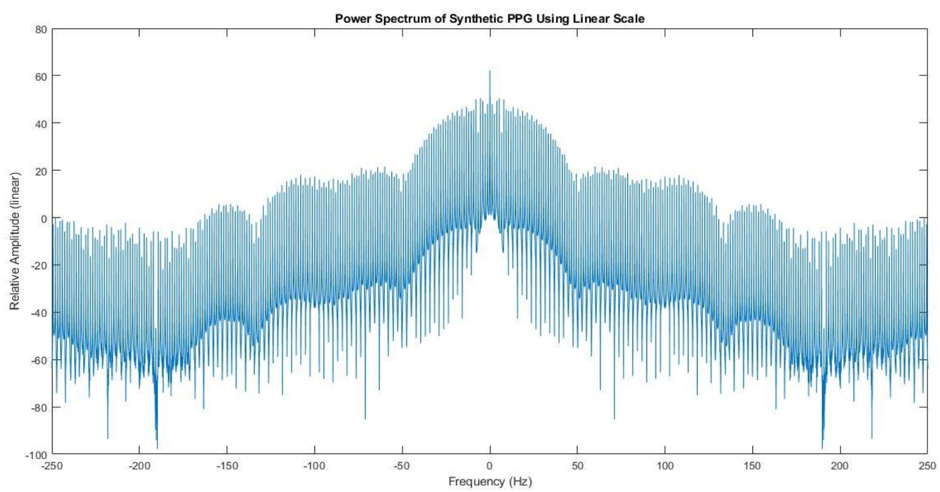

Figure 5-6. Power spectrum of an artificial PPG signal

Figures 5-5 and 5-6 show that the general outline of the power spectrums from the different sources are the same. This can be attributed to the fact that they are both PPG signals. The ECG signal in Figure 5-2 has a distinct outline for its power spectrum.

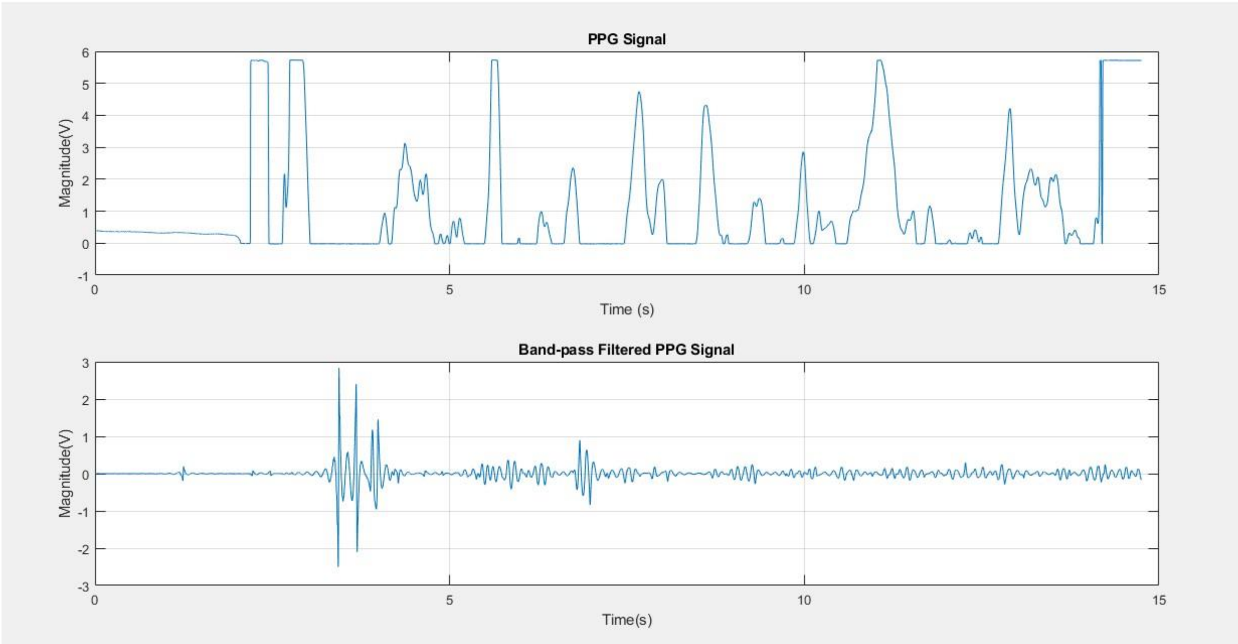

Figure 5-7. Time-domain plot of sensor PPG signal and the filtered resultant

The dataset obtained from the device was passed through the bandpass filter designed in Figure 4-5. The outcome of that can be seen in Figure 5-7 in the time domain.

5.2 Application results

Some relevant screenshots of the mobile app called Remoh can be found in Figures 5-8 to 5-10 below.

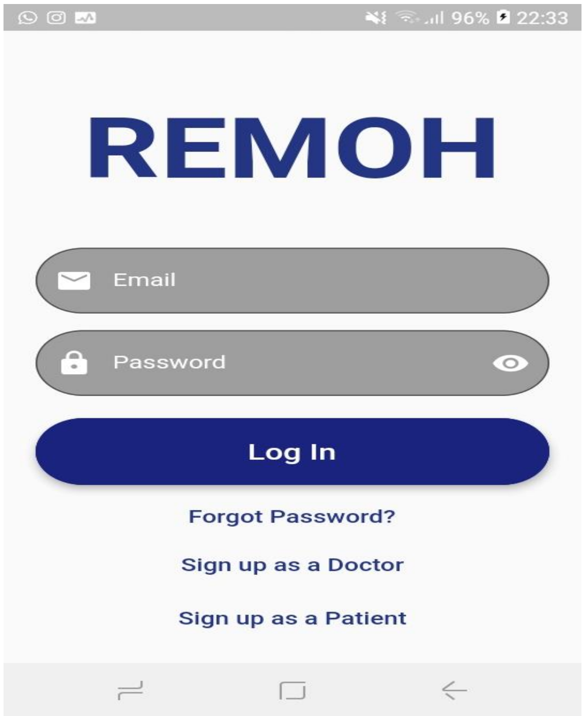

Figure 5-8. App login page for all users

Figure 5-9. App register page for doctors. Patients' register page is similar with extra information

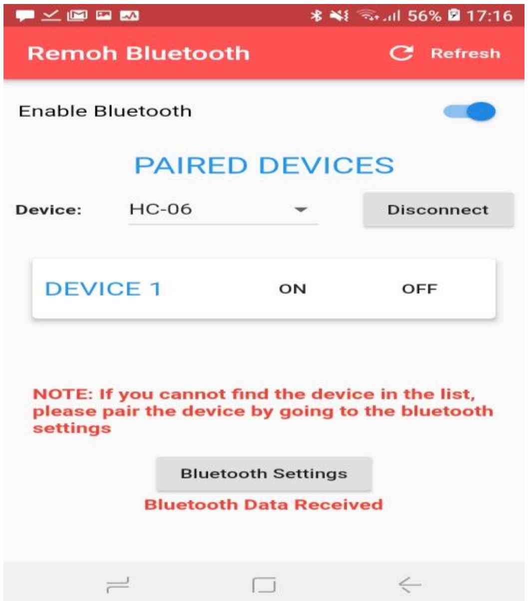

Figure 5-10. App Bluetooth page to pair devices and view BPM

The statement "Bluetooth data received" in Figure 5-9 happens after a connection has been made to the device, and it has received the heart rate data being transmitted as BPM from the microcontroller.

5.3 Statistical Analysis

Statistical analysis was performed on the data sets using an independent t-test. This test was to check the null hypothesis variance of the datasets obtained from the device against standard PPG signals; in this case, synthetic ones from the Vortal database. The algorithm for this can be found in appendix 5. Results were H=1 and P=0, meaning that the null hypothesis was rejected, and the data obtained from the difference bore statistically significant differences to standard PPG signals.

6 Conclusion

6.1 Discussions

The results from the tests documented above did not fulfill all the objectives listed in chapter 1 of this paper. Firstly, the desire for reliable data from the pulse sensor was not realized, as is evident from the t-test results. The null hypothesis that there was no variance between data from the sensor and synthetic PPG signals from databases was rejected because the p-value was less than 0.05. The algorithm used in Arduino needs to be optimized or accuracy.

Another expectation for this project was for doctors and patients to be able to sign up and access data recorded through a web app. This has been achieved, but with a mobile application after problems setting up Bluetooth communication through a web app were encountered. A database has been implemented to store the heart rate data. The patient and health professional can sign up and log in on the app and view instantaneously recorded heart-rate data.

Finally, this project was supposed to be a low-cost alternative to existing optical pulse sensing technology, and it was on track to achieve that based on the design and cost calculations.

6.2 Limitations

Working from home and social distancing significantly limited access to hardware that could have given a better output with the circuit (like an OP402 amplifier suggested during a presentation). It also hampered plans to make this device more compact and wearable with printed circuit boards

Another limitation of the simulation was Simulink's problems communicating directly with the microcontroller. This meant that data could not be processed in real-time as planned. The unavailability of real-time data for processing means

Filtering out melanin to clean the data was difficult, especially since none of the algorithms explicitly mentioned that. The results above do not reflect an extraction of signal elements introduced by melanin during testing.

6.3 Future works

This project ended as a prototype with a focus on analyzing PPG signals to extract the heart rate of patients to make medical decisions. The things in the objectives and requirements which could not be achieved are listed here and include:

- Ensuring seamless transfer of data from the microcontroller to web and mobile apps.
- Building both web and mobile apps for different kinds of users.
- Access to patients' historical medical data to determine patterns, predict and analyze health trends better.
- Using Wi-Fi rather than Bluetooth to optimize data transfer while maintaining minimum power usage.

Aside from these, considerable work will be done in integrating different instruments for measuring vital physiological data like temperature, blood pressure, and more to make more extensive medical decisions.

References

[1] "Publications | Ghana Health Service." https://www.ghanahealthservice.org/ghs-category.php?cid=5 (accessed May 05, 2020).

[2] R. Kumar and R. Pal, "India achieves WHO recommended doctor population ratio: A call for paradigm shift in public health discourse!," *J. Fam. Med. Prim. Care,* vol. 7, no. 5, pp. 841–844, 2018, doi: 10.4103/jfmpc.jfmpc_218_18.

[3] M. Kinyua, N. Boruett, and W. Kangara, "E-learning, transforming the health sector," *CIO East Africa*, vol. 4, no. 9, 2012.

[4] "Remote Health Care Is the Future of Medicine - ASME." https://www.asme.org/topics-resources/content/remote-health-care-future-medicine (accessed May 05, 2020).

[5] G. Lamprinakos, E. Kosmatos, D. Kaklamani, and I. Venieris, "An Integrated Architecture for Remote Healthcare Monitoring," in *2010 14th Panhellenic Conference on Informatics*, Sep. 2010, pp. 12–15, doi: 10.1109/PCI.2010.20.

[6] B. Yuan and J. Herbert, "Web-based real-time remote monitoring for pervasive healthcare," in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, Mar. 2011, pp. 625–629, doi: 10.1109/PERCOMW.2011.5766964.

[7] S. Kale and C. S. Khandelwal, "Design and implementation of real time embedded tele-health monitoring system," in *2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT)*, Mar. 2013, pp. 771–774, doi: 10.1109/ICCPCT.2013.6528842.

[8] T. S. Arulananth and B. Shilpa, "Fingertip based heart beat monitoring system using embedded systems," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Apr. 2017, vol. 2, pp. 227–230, doi: 10.1109/ICECA.2017.8212802.

[9] M. Rezaei and A. S. Basu, "Comparison of Two Low-Power Signal Processing Algorithms for Optical Heart Rate Monitoring," in *2018 IEEE SENSORS*, Oct. 2018, pp. 1–4, doi: 10.1109/ICSENS.2018.8589811.

[10] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiol. Meas.*, vol. 28, no. 3, pp. R1–R39, Feb. 2007, doi: 10.1088/0967-3334/28/3/R01.

[11] A. E. Awodeyi, S. R. Alty, and M. Ghavami, "On the Filtering of Photoplethysmography Signals," in *2014 IEEE International Conference on Bioinformatics and Bioengineering*, Nov. 2014, pp. 175–178, doi: 10.1109/BIBE.2014.76.

[12] J. Parak, A. Tarniceriu, P. Renevey, M. Bertschi, R. Delgado-Gonzalo, and I. Korhonen, "Evaluation of the beat-to-beat detection accuracy of PulseOn wearable optical heart rate monitor," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug. 2015, pp. 8099–8102, doi: 10.1109/EMBC.2015.7320273.

[13] T. Y. Abay and P. A. Kyriacou, "Accuracy of reflectance photoplethysmography on detecting cuff-induced vascular occlusions," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug. 2015, pp. 861–864, doi: 10.1109/EMBC.2015.7318498.

[14] H. Han and J. Kim, “Artifacts in wearable photoplethysmographs during daily life motions and their reduction with least mean square based active noise cancellation method,” *Comput. Biol. Med.*, vol. 42, no. 4, pp. 387–393, Apr. 2012, doi: 10.1016/j.compbiomed.2011.12.005.

[15] S. Lee, H. Shin, and C. Hahm, “Effective PPG sensor placement for reflected red and green light, and infrared wristband-type photoplethysmography,” in *2016 18th International Conference on Advanced Communication Technology (ICACT)*, Jan. 2016, pp. 556–558, doi: 10.1109/ICACT.2016.7423470.

[16] A. Lymberis, “Smart wearables for remote health monitoring, from prevention to rehabilitation: current R D, future challenges,” in *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003.*, Apr. 2003, pp. 272–275, doi: 10.1109/ITAB.2003.1222530.

[17] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, and J. Henry, “IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things,” p. 351.

[18] R. A. Rashid and R. Yusoff, “Bluetooth Performance Analysis in Personal Area Network (PAN),” in *2006 International RF and Microwave Conference*, Sep. 2006, pp. 393–397, doi: 10.1109/RFM.2006.331112.

[19] “Heart rate: What’s normal? - Mayo Clinic.” https://www.mayoclinic.org/healthy-lifestyle/fitness/expert-answers/heart-rate/faq-20057979 (accessed May 11, 2020).

[20] F. Elsamnah *et al.*, “Reflectance-based monolithic organic pulsemeter device for measuring photoplethysmogram signal,” in *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, May 2018, pp. 1–5, doi: 10.1109/I2MTC.2018.8409873.

[21] P. Zahradnik and M. Vlcek, “Analytical design method for optimal equiripple comb FIR filters,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 52, no. 2, pp. 112–115, Feb. 2005, doi: 10.1109/TCSII.2001.840117.

[22] J. Pan and W. J. Tompkins, “A Real-Time QRS Detection Algorithm,” *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985, doi: 10.1109/TBME.1985.325532.

[23] H. Sedghamiz, “Matlab Implementation of Pan Tompkins ECG QRS detector,” 2014. https://www.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector (accessed May 10, 2020).

[24] P. H. Charlton, T. Bonnici, L. Tarassenko, D. A. Clifton, R. Beale, and P. J. Watkinson, “An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram,” *Physiol. Meas.*, vol. 37, no. 4, pp. 610–626, Mar. 2016, doi: 10.1088/0967-3334/37/4/610.

Appendix

Appendix A

MATLAB code to generate a power spectrum

```
% system impedance (ohms)
R=10;

% sampling frequency (Hz)
fs=500;

% read data from file
%s=xlsread('testdataMe.xlsx','B12:B6722'); %reading .xls files
s=csvread('rrest-syn002_data.csv',2810,1);  %  Reading  .csv  files
%csvread('FILENAME',Row,Column)

% number of time-domain samples
L = length(s); %length(variable containing data)

% time vector for time-domain signal (s)
t=(1/fs)*(1:L);

% frequency vector for frequency-domain signal (Hz)
nfft=L;
f=fs/2*(-1:2/nfft:1-2/nfft);

% normalized FFT of signal
S=(fftshift(fft(s,nfft))/(L));

% power spectrum
Sp=10*log10((abs(S).^2)/R*1000);

% plot power spectrum
figure(2)
clf
plot(f,Sp)
%plot(f/1e3,Sp)
```

```
title('Power Spectrum of Synthetic PPG Using Linear Scale') %('Power
Spectrum of Synthetic PPG Using Linear Scale')
xlabel('Frequency (Hz)')
ylabel('Relative Amplitude (linear)')
```

Appendix B

The Pan-Tompkins algorithm to detect QRS complexes in an ECG signal can be found here [23].

The MATLAB algorithm is too lengthy for this paper.

Appendix C

Snippets of Arduino code that calculates BPM from the signal.

```
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
#include <SoftwareSerial.h>  //Library for bluetooth
SoftwareSerial BTserial(10, 11);
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);

//  Variables
int pulsePin = 0;                   // Pulse Sensor purple wire
connected to analog pin 0
int blinkPin = 8;

// Volatile Variables, used in the interrupt service routine
volatile int BPM;                   // int that holds raw Analog
in 0. updated every 2mS
volatile int Signal;                // holds the incoming raw data
volatile int IBI = 600;//600;                 // int that holds the
time interval between beats
//normal IBI (R-R inteval is 0.6-1.2seconds)
volatile boolean Pulse = false;      // "True" when User's live
heartbeat is detected. "False" when not a "live beat".
volatile boolean QS = false;        // becomes true when Arduoino
finds a beat.

// Regards Serial OutPut  -- Set This Up to your needs
static boolean serialVisual = true;   // Set to 'false' by Default.
Re-set to 'true' to see Arduino Serial Monitor ASCII Visual Pulse

volatile int rate[10];                      // array to hold last
ten IBI values
volatile unsigned long sampleCounter = 0;            // used to
determine pulse timing
```

```
volatile unsigned long lastBeatTime = 0;           // used to find
IBI
volatile int P = 512;                            // used to find peak in
pulse wave
volatile int T = 512;                          // used to find trough in
pulse wave
volatile int thresh = 525;                       // used to find instant
moment of heart beat
volatile int amp = 100;                         // used to hold amplitude
of pulse waveform, seeded, 100
volatile boolean firstBeat = true;          // used to seed rate
array so we startup with reasonable BPM
volatile boolean secondBeat = false;        // used to seed rate
array so we startup with reasonable BPM

void setup()
{  lcd.init();
  lcd.backlight();
  pinMode(blinkPin,OUTPUT);        // pin that blinks to  heartbeat
  Serial.begin(9600);            // baudrate use 115200 to be fast
  BTserial.begin(9600);
  Serial.println("Date&Time, Beat Alert, BPM");
  delay(1000);
  interruptSetup();                    // ReadsPulse Sensor signal
every 2mS
  lcd.print("Place finger");    //Message printed to the LCD when
the device starts up to prompt user
}

//  Part where measurement happens
void loop()
{
   serialOutput();

  if (QS == true) // A Heartbeat Was Found
    {
      sOutonBeat();   // A Beat Happened show in serial
      QS = false; // reset the Quantified Self flag for next time
    }
  delay(50);  // break
}
void interruptSetup()
{
  // Initializes Timer2 to throw an interrupt every 2mS.
  TCCR2A = 0x10;     // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND
GO INTO CTC MODE
  TCCR2B = 0x06;     // 256 PRESCALER WITHOUT COMPARING
  OCR2A = 0X7C;      // SETS TOP OF COUNT TO 124 FOR 500Hz SAMPLE
RATE
  TIMSK2 = 0x02;      // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2
AND OCR2A
  sei();             // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
```

```
}

void serialOutput()
{   // Decide How To Output Serial.
 if (serialVisual == true)
  {
     //arduinoSerialMonitorVisual('-',  Signal);      //  goes  to
function that makes Serial Monitor Visualizer
     arduinoSerialMonitorVisual(Signal);
  }
 else
  {
      //sendDataToSerial(Signal);     // goes to sendDataToSerial
function
      sendDataToSerial('S', Signal);
   }
}

void sOutonBeat()
{
 if (serialVisual == true) //  Code to Make the Serial Monitor
Visualizer Work
   {
     Serial.print(",");
     Serial.print("Beat detected");    //Output on serial monitor
when a heart beat is detected
     Serial.print(",");
     Serial.println(BPM);
     BTserial.write(BPM);
     BTserial.print(BPM);
     for  (byte  i=0;i<2;i++)  {  //converts  BPM  to  binary  for
transmission through bluetooth
      BTserial.write(static_cast<byte>(BPM));
     // this is decoded in the flutter app and the BPM in decimal
is printed
     }
     lcd.clear();
     lcd.print("BPM: ");
     lcd.print(BPM);
   }
 else
   {
     sendDataToSerial(BPM);      // send heart rate
     //sendDataToSerial('B',BPM);   // send heart rate with a 'B'
prefix

     sendDataToSerial('Q',IBI);   // send time between beats with
a 'Q' prefix
     //sendDataToSerial(IBI);
   }
}
   Serial.print(",");
```

```
   Serial.print(symbol);
   Serial.print(",");
   Serial.println(data);
}

ISR(TIMER2_COMPA_vect)   //triggered when Timer2 counts to 124
{
  cli();                                      // disable interrupts
for a bit
  Signal = analogRead(pulsePin);                 // read the Pulse
Sensor
  sampleCounter += 2;                          // keep track of the
time in mS
  int N = sampleCounter - lastBeatTime;        // monitor the time
since the last beat to avoid noise
                                               //   find  the  peak
and trough of the pulse wave
  if(Signal < thresh && N > (IBI/5)*3)         // avoid dichrotic
noise by waiting 3/5 of last IBI
    {
      if (Signal < T) // T is the trough. Keeps track of lowest
point in pulse wave
      {                        T = Signal; }
    }
  if(Signal > thresh && Signal > P)          // thresh condition
helps avoid noise
    {
      P = Signal;                                // P is the peak.
Keeps track of highest point in pulse wave
    }

  //  LOOKING FOR THE HEART BEAT
  // signal surges in value every time there is a pulse
  if (N > 250)
  {                                  // avoid high frequency noise
    if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3)
)
      {
        Pulse = true;                                    // set the
Pulse flag when pulse is found
        digitalWrite(blinkPin,HIGH);                // turn on pin
13 LED
        IBI = sampleCounter - lastBeatTime;         // measure time
between beats in mS
        lastBeatTime = sampleCounter;                // keep track
of time for next pulse
        if(secondBeat)
        {                        // if this is the second beat, if
secondBeat == TRUE
          secondBeat = false;                  // clear secondBeat
flag
```

```
          for(int i=0; i<=9; i++) // seed the running total to get
a realisitic BPM at startup
          {              rate[i] = IBI;                      } }
        if(firstBeat) // if it's the first time we found a beat,
if firstBeat == TRUE
        {
          firstBeat = false;                   // clear firstBeat
flag
          secondBeat = true;                   // set the second
beat flag
          sei();                               // enable interrupts
again
          return;                              // IBI value is
unreliable so discard it
        }
      // keep a running total of the last 10 IBI values
      word runningTotal = 0;                   // clear the
runningTotal variable
      for(int i=0; i<=8; i++)
        {                // shift data in the rate array
          rate[i] = rate[i+1];                 // and drop the
oldest IBI value
          runningTotal += rate[i];             // add up the 9
oldest IBI values
        }
      rate[9] = IBI;                           // add the latest
IBI to the rate array
      runningTotal += rate[9];                 // add the latest
IBI to runningTotal
      runningTotal /= 10;                      // average the last
10 IBI values
      BPM = 60000/runningTotal;                // beats per minute
      QS = true;                               // set Quantified
Self flag
      // QS FLAG IS NOT CLEARED INSIDE THIS ISR
    }                        }
  if (Signal < thresh && Pulse == true)
    {   // when the values are going down, the beat is over
      digitalWrite(blinkPin,LOW);              // turn off pin 13
LED
      Pulse = false;                           // reset the Pulse
flag for repetition
      amp = P - T;                             // get amplitude of
the pulse wave
      thresh = amp/2 + T;                      // set thresh at 50%
of the amplitude
      P = thresh;                              // reset these for
next time
      T = thresh;
    }
  if (N > 2500)
```

```
    {                                 // if 2.5 seconds go by without a
beat
      thresh = 512;                           // set thresh default
      P = 512;                                // set P default
      T = 512;                                // set T default
      lastBeatTime = sampleCounter;                  // bring the
lastBeatTime up to date
      firstBeat = true;                       // set these to avoid
noise
      secondBeat = false;                        // when we get the
heartbeat back
    }
  sei();                                      // enable interrupts
when done
}
```

Appendix D

MATLAB code for student's t-test to measure the variability of the pulse sensor-generated data from synthetic PPG data.

```
% Performing a t-test on the unpaired data sets.
% Data from synthetic PPG as well as data from pulse sensor

psppg = xlsread('testdataMe.xlsx','B3:B6722'); %pulse sensor
sppg = csvread('rrest-syn002_data.csv',1,1); %synthetic ppg
%psp = psppg(:,1); turning data into arrays
%sp = sppg(:,1);
[H,P] = ttest2(psppg,sppg)
%result: [H,P] = [1,0]
```

Appendix E

The dart code for the Remoh (what I call the entire project) app can be found here. This link takes the reader to a GitHub repository since the code is far too lengthy to be put in his document.

Appendix F

MATLAB algorithm for the implementation of the FIR bandpass filter

```
%%BAND-PASS FILTER DESIGN

Fstop1 = 0.05; % First Stopband Frequency
Fpass1 = 0.5; % First Passband Frequency
Fpass2 = 5; % Second Passband Frequency
Fstop2 = 5.05; % Second Stopband Frequency
Astop1 = 60; % First Stopband Attenuation (dB)
Apass = 1; % Passband Ripple (dB)
Astop2 = 60; % Second Stopband Attenuation (dB)
Fs = 25; % Sampling Frequency
Fn = Fs/2; %Nyquist Frequency

%Implementing the filter
sig = xlsread('testdataMe.xlsx','B100:B6000');
h = fdesign.bandpass('fst1,fp1,fp2,fst2,ast1,ap,ast2', Fstop1,
Fpass1,Fpass2, Fstop2, Astop1, Apass, Astop2, Fs);
Hd = design(h, 'equiripple');

fvtool(Hd)

fs=25;  % sampling frequency (Hz)

% number of time-domain samples
L = length(sig);


% time vector for time-domain signal (sig)
tvect=(1/400)*(1:L);


%%PASSING PPG SIGNAL THROUGH BANDPASS FILTER
```

```
S = filter(Hd,sig);


figure(1)
subplot(2,1,1)
plot(tvect,sig)
grid on
title('PPG Signal')
xlabel('Time (s)')
ylabel('Magnitude(V)')

subplot(2,1,2)
plot(tvect,S)
grid on
title('Band-pass Filtered PPG Signal')
xlabel('Time(s)')
ylabel('Magnitude(V)')
```