An IoT-based real-time telemedicine system from the perspective of Bangladesh

Project & Thesis-II

CSE 4250

A thesis

Submitted in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Submitted By

Muhammad Khaleduzzaman	**18.01.04.102**
Md. Tamzidul Islam	**18.01.04.123**
Md. Mubayer Rahman	**18.01.04.144**
Tanzela Nasrin Aohana	**18.02.04.012**

Supervised by,

Dr. Md. Raqibul Hasan

Ms. Nawshin Tabassum Tanny

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

May 31, 2023

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Dr. Md. Raqibul Hasan and Ms. Nawshin Tabassum Tanny, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, following the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part of has been submitted anywhere else for the award of any degree, diploma, or other qualifications.

Muhammad Khaleduzzaman
18.01.04.102

Md. Tamzidul Islam
18.01.04.123

Md. Mubayer Rahman
18.01.04.144

Tanzela Nasrin Aohana
18.02.04.012

CERTIFICATION

This thesis titled, "An IoT-based real-time telemedicine system from the perspective of Bangladesh", submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering on May 31, 2023.

Group Members:

Muhammad Khaleduzzaman	**18.01.04.102**
Md. Tamzidul Islam	**18.01.04.123**
Md. Mubayer Rahman	**18.01.04.144**
Tanzela Nasrin Aohana	**18.02.04.012**

Ms. Nawshin Tabassum Tanny
Lecturer
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Prof. Dr. Md. Shahriar Mahbub
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENTS

The authors would like to express with due respect their deepest gratitude to their supervisors Dr. Md. Raqibul Hasan, former assistant professor, and Ms. Nawshin Tabassum Tanny, Lecturer, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, whose guidance and valuable directives were encouraging at all stages of this research.

The authors are also grateful to the Library of Ahsanullah University of Science and Technology, for providing the resources required to successfully complete this work.

Finally, they thank all the individuals who shared their knowledge through the World Wide Web.

Dhaka
May 31, 2023

Muhammad Khaleduzzaman

Md. Tamzidul Islam

Md. Mubayer Rahman

Tanzela Nasrin Aohana

ABSTRACT

Telemedicine is a rapidly growing field that has the potential to transform healthcare delivery. It uses a variety of technologies, such as video conferencing, remote monitoring, and mobile health applications, to connect patients and healthcare providers across distances. This can be especially beneficial for patients who live in rural or remote areas, as it can provide access to healthcare services that would otherwise be difficult to obtain. Telemedicine can be used for a wide range of medical services, including consultations with specialists, remote monitoring of chronic conditions, and mental health services. It can also be used to provide medical education and training to healthcare providers in remote areas. Telemedicine is the delivery of health care and the exchange of healthcare information across distances. Telemedicine is basically about two things. (1) the interaction between the patient and the doctor in real-time and (2) the type of information being transmitted (e.g., heartbeat, signals, texts, audio, video). Much of the telemedicine which is now practiced is performed in industrialized countries, but there is increasing interest in the use of telemedicine in developing countries, mostly in rural areas or villages. Telemedicine is considered under two things: (1) when there is no alternative (e.g., in emergencies in remote environments), and (2) when it is better than existing conventional services (e.g., teleradiology for rural hospitals). Telemedicine can be expected to improve equity of access to health care, the quality of that care, and the efficiency with which it is delivered.

Keywords: Telemedicine, healthcare delivery, technology, video conferencing, remote monitoring, mobile health applications, rural areas, remote areas, medical services, specialists, chronic conditions, mental health, medical education, healthcare providers, industrialized countries, developing countries, emergencies, conventional services, teleradiology, equity of access, quality of care, efficiency.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

Introduction

1.1 General

Telemedicine is one of the most vital programs in the present scenario. Technology is advancing day by day with lots of new terminologies and methodologies. Telemedicine brought into the focus the development of the health care system. Through electronic mediums, the resources are transmitted to every family. The way of interpretation of the modules is done to meet certain criteria of the health facility. Various kinds of medicine and insulins are needed for the survival of human beings. The whole implication technique is set to standardize in Bangladesh. There is a huge amount of area in Bangladesh where there is a lack of health care management. To regulate the process Telemedicine is required to mitigate the issues. This article covers some vital propaganda related to the standardization of telemedicine in Bangladesh. The supply chain is also one of the biggest models enrolled in this program.

Body sensors are a type of wearable technology that can be used in telemedicine to monitor a patient's health remotely. These sensors can collect data on a patient's vital signs, such as heart rate, blood pressure, and oxygen levels, as well as other health metrics, such as activity levels, sleep patterns, and even stress levels. Body sensors typically use wireless technology to transmit the data they collect to a healthcare provider or monitoring system. This allows healthcare providers to monitor a patient's health in real time and detect any changes or anomalies that may require intervention. Body sensors can also be used for remote patient monitoring, allowing healthcare providers to monitor patients who have been discharged from the hospital or who are recovering from surgery. This can help to prevent complications and ensure that patients receive the appropriate care they need. They are a valuable tool in telemedicine that can help to improve patient outcomes and increase access to healthcare services. By allowing healthcare providers to monitor patients remotely, body sensors can help to ensure that patients receive timely and appropriate care, regardless of their location or mobility.

In this work, we used IoT and biomedical sensors to collect, analyze, and transmit a patient’s health data/signal in real-time to a remote doctor. The patient and doctor could be far away from each other. This telemedicine platform will bring health care benefits within reach of rural people where it is difficult to find a specialist doctor for consultation. It will alleviate the long travel of a patient from village to city where specialist doctors are usually located. Hence the proposed system will save our time, energy, and healthcare costs.

1.2 Motivation

One of the primary motivations for working in telemedicine is the desire to improve access to healthcare services for underserved populations. Telemedicine has the potential to provide healthcare services to people who live in rural or remote areas, as well as those who have mobility issues or disabilities that make it difficult to travel to a healthcare facility. By working in telemedicine, healthcare providers can help to ensure that all patients have access to the care they need, regardless of their location or circumstances. Developing countries like Bangladesh suffer from inadequate healthcare and medical services. Lack of healthcare professionals and infrastructure contribute to this problem making it more and more difficult to deliver healthcare to people in rural and remote communities of the developing world. Particularly, there is a severe shortage of specialist doctors in the villages of Bangladesh. The doctors prefer to live in the cities because of better living, and better earnings.

Another motivation for working in telemedicine is the opportunity to use innovative technologies to improve healthcare delivery. Telemedicine involves the use of a variety of technologies, such as video conferencing, remote monitoring, and mobile health applications, to connect patients and healthcare providers across distances. By working in this field, healthcare providers can use their knowledge and expertise to help develop and implement new technologies that can improve patient outcomes and reduce healthcare costs. Over the past few years, there has been a growing interest in the ubiquitous healthcare system. Quality healthcare service availability is a huge challenge facing developing countries of the world like Bangladesh. The qualities of healthcare services in urban and rural areas are not evenly distributed. Wireless communication advancements and the creation of Telemedicine have paved the path for more convenient and high-quality healthcare services. Thus, in an emergency case, a real-time healthcare system is needed to remove the geographical barriers and improve the doctor-patient ratio virtually so that healthcare facilities are easily available in remote areas of Bangladesh. This technique also significantly lowers the cost of healthcare solutions.

Availability, accessibility, and efficient sharing of continuously growing medical data of a large number of people are the biggest challenge in the medical field. In the proposed system, the patient’s medical data will be stored on the cloud. Thus, the patient’s medical history will be available to doctors from anywhere at any time.

1.3 Objective of this study

The main objective of telemedicine is to improve access to healthcare services and provide quality healthcare to patients who may not have access to in-person healthcare services due to distance, mobility issues, or other barriers. Telemedicine aims to use technology to bridge the gap between patients and healthcare providers, allowing them to communicate and exchange medical information in real-time.

The objectives of telemedicine can be broken down into several key areas:

Access: Telemedicine aims to provide access to healthcare services for patients who may not have access to traditional in-person healthcare services, such as those who live in rural or remote areas.

Quality: Telemedicine aims to provide high-quality healthcare services to patients by allowing healthcare providers to monitor patients remotely and provide timely interventions when necessary.

Efficiency: Telemedicine aims to improve the efficiency of healthcare delivery by reducing the need for in-person visits and hospitalizations, as well as reducing wait times and improving patient flow.

Cost-effectiveness: Telemedicine aims to reduce healthcare costs by reducing the need for expensive in-person visits and hospitalizations, as well as reducing the cost of transportation and other related expenses.

Patient satisfaction: Telemedicine aims to improve patient satisfaction by providing convenient and accessible healthcare services that meet their needs and preferences.

Timeliness: Telemedicine aims to provide timely healthcare services to patients by allowing healthcare providers to monitor patients remotely and provide interventions in real-time.

Continuity of care: Telemedicine aims to improve continuity of care by allowing healthcare providers to monitor patients over time and provide ongoing care and support.

Education and training: Telemedicine aims to provide medical education and training to healthcare providers, particularly those in remote or underserved areas, to improve the quality of healthcare services.

Public health: Telemedicine aims to improve public health outcomes by providing population-level healthcare services, such as disease surveillance and monitoring, to help prevent the spread of infectious diseases and other health concerns.

Research: Telemedicine aims to support medical research by providing access to patient data and enabling remote monitoring of patients in clinical trials.

The objectives of telemedicine are focused on improving access to healthcare services, enhancing the quality and efficiency of healthcare delivery, and improving patient outcomes and satisfaction. Telemedicine also has the potential to support medical education, public health initiatives, and research, making it a valuable tool for improving healthcare delivery and outcomes.

1.3 Thesis Organization

In our research, our main aim is to establish a telemedicine system that will be beneficial to both doctors and patients.

CHAPTER ONE (Introduction): In this chapter, we have discussed problem definition and basic concepts, and we aim to create a telemedicine system using sensors and software.

CHAPTER TWO (Literature Review): In this chapter, we have discussed a short description of related papers which we have studied. We have also explained our overall telemedicine system here.

CHAPTER THREE (Methodology): In this chapter, we have discussed our prototype development for the hardware we built. We have also explained our work ideas and features.

CHAPTER FOUR (Experimental Result): In this chapter, we have discussed our noise suppression system and deployment of the software.

CHAPTER FIVE (Conclusion and Future work): Here we have finished up our paper by featuring our topic. Probable plans and ideas are also described here.

CHAPTER TWO

Literature Review

2.1 Related Work

- Work in [1] implemented a patient monitor system, which aims to gather data for continuous administration of patient’s health. The proposed work wants to implement a real time communication system for the doctors to administer medicine or other treatments.

- In [2] the researchers implemented a system, which monitors the body parameters such as pulse rate, ECG. ARM7LPC 2138 processor is used as a main interface and the data displayed by using graphical user interface. We showed that data through an android app to the doctor in real time of the patient’s health in their own home with the help of wireless sensor technology.

- Work in [3] focused on monitoring ECG signals in real time using IoT. The acquired signals are processed to extract certain required features (such as wavelet coefficients) and to present the ECG signal’s characteristics accurately before utilization.
- Many researchers have attempted before to solve it in many ways, but the earlier methods in several cases either SMS will be sent using GSM or an RF module will be used to send the patient’s data from the sender device to the receiver device. So, the purpose of this thesis is to maintain a record of patients’ data and to give an emergency alert if required, using a different technology which is the Internet of Things (IoT).
- In [1] and [2] the researchers wanted to implement the sensor results through the LCD of the device. They made a very cost-effective system for that process.

2.2 Overall system

- We have developed an affordable and accessible health care system for the rural people of Bangladesh.

- We are using ECG sensors to monitor and extract real-time heart rate information from patients and transmission of the acquired data to mobile phones using electronic stethoscope. We have worked on removing unwanted noise using noise cancelling algorithms.

- There is an app that stores patient's data on the cloud. Thus, the historical data of the patient will be available for doctors to access at any time from everywhere.

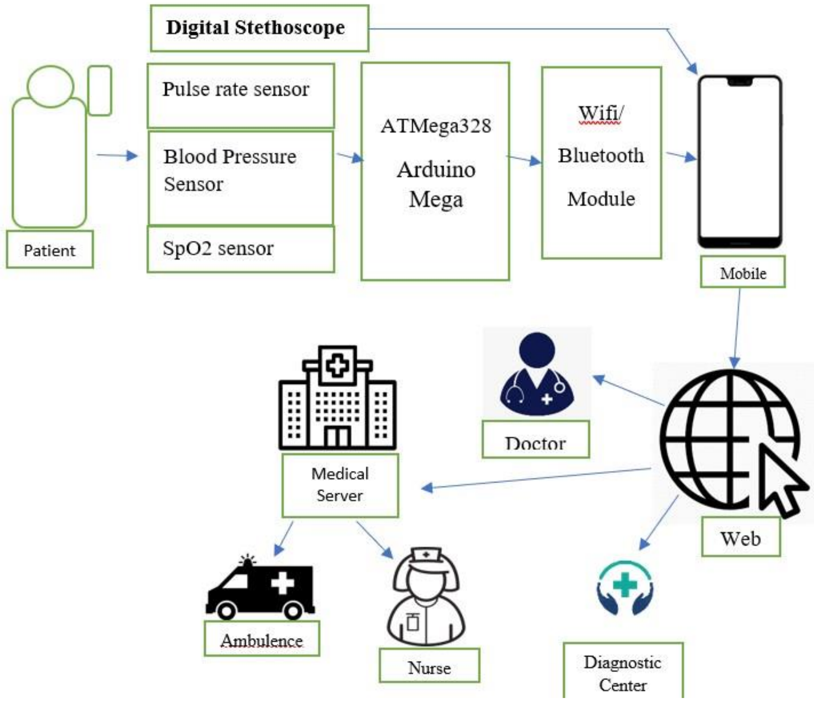

Figure 1. Overall system diagram of proposed telemedicine model

As illustrated in Figure 1, the heart rhythm is collected from the digital stethoscope and medical sensors on the patient's body to extract the body parameters. The digital stethoscope can be inserted directly into the mobile's 3.5mm headphone jack to send the heartbeat directly to the doctor in real-time via the developing app. The audio signal will undergo through low pass filter algorithm to avoid unwanted noise. All the other data will be transmitted via Bluetooth/wifi to the app. The app will store the data in the firebase database system and be shown on doctor's end in real-time using the IoT system.

CHAPTER THREE

Methodology

3.1 Prototype Development

An electronic stethoscope is at the core of the proposed system. A doctor mostly relies on a stethoscope signal for his/her initial examination of a patient. We will capture the patient's heart and lung sounds using the electric stethoscope and transmit the signal to the remote doctor's end in real-time. We will also use biomedical sensors such as pulse rate sensor, temperature sensor (LM35), SpO2 sensor for oxygen level measurement, blood pressure sensor and Heartbeat sensor (ECG). These sensor signals will be sent to microcontroller, Arduino Mega which is interfaced with a Wi-Fi module to transmit data to an android device or PC.

While using the stethoscope, the signals make some noise and to cancel the noise we are using low pass filter on Mat Lab, which allows signal below cutoff frequency and attenuates signal below cutoff frequency.

The data we are collecting from the app using the stethoscope will be stored in a Database, example: Firebase. This project has executed an Arduino based project which measures the necessary heart rate, bpm, spo2 etc. by detecting fingerprints of a human. Then all the data transfers in the mobile app and display the data in mobile screen.

3.2 Prototype Hardware Design

This is our Arduino-based project which measures the necessary heart rate, bpm, spo2, etc. by detecting the fingerprints of a human. And then all the data transfers in the mobile app and display the data on the mobile screen. The data is stored in a database.

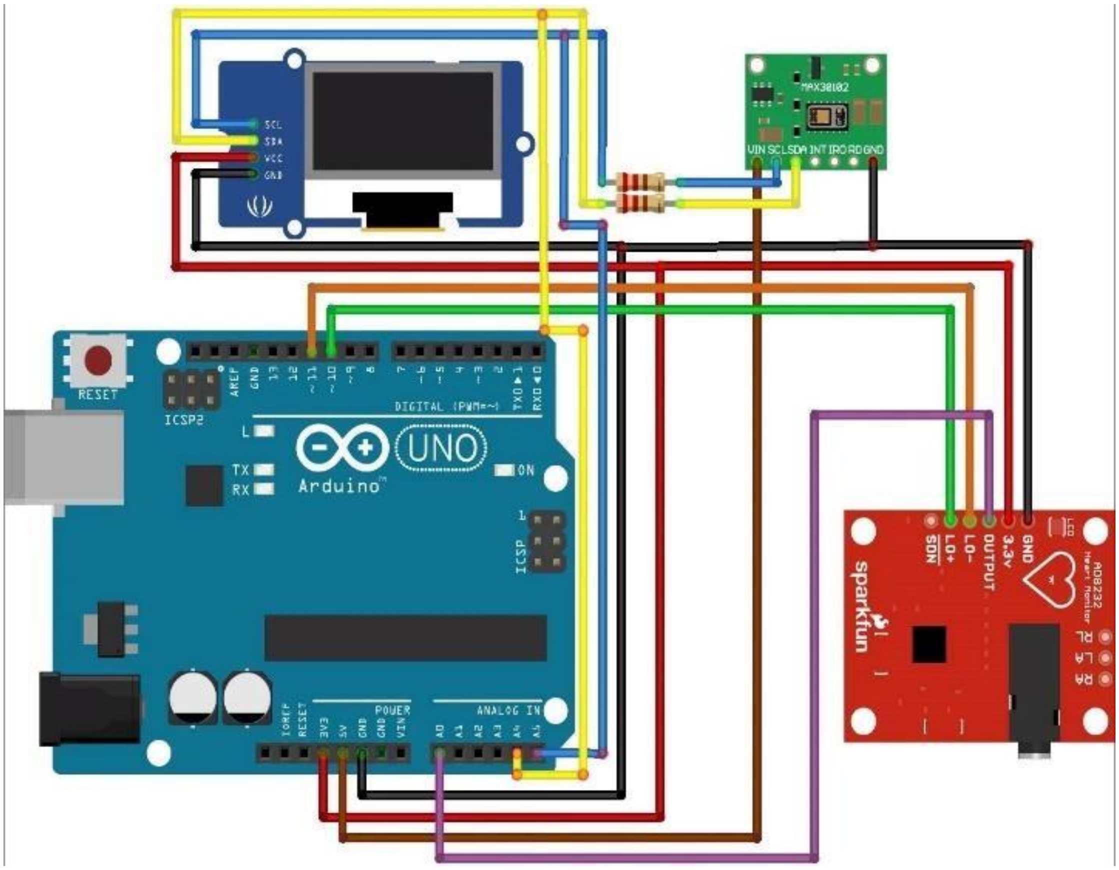

Figure 2: Figure 2: Prototype Hardware Diagram

Table 1: Hardware Components

S.No.	Components	Use
01	Arduino-Nano	Brain of the system
02	MAX30102	Detect Pulse Rate, SpO_2
03	AD8232	Detect ECG signal
04	Li-Po Battery	Power the system

In this device, Arduino is used. Arduino is the brain of this hardware. Here MAX30102 is used. It is used to detect pulse rate and SpO2. AD8232 is used to detect ECG signals. The Li-Po battery is the power system of this device.

Arduino Nano: Arduino Nano is a small, versatile, and widely used development board based on the ATmega microcontroller series. It is part of the Arduino family of microcontroller boards and shares many similarities with other Arduino boards, such as the Uno, but with a smaller form factor. The Arduino Nano board is designed to be compact and easily integrated into various projects. It features a microcontroller chip, input/output pins, digital and analog pins, a power regulator, a USB interface, and other essential components necessary for building and programming electronic circuits. The board can be powered via USB or an external power supply.

The ATmega microcontroller on the Arduino Nano provides a wide range of computational capabilities, making it suitable for a variety of applications. It can be programmed using the Arduino software, which is based on the C++ programming language. The software offers a user-friendly interface and a vast library of prewritten functions to simplify programming tasks. The Arduino Nano board is commonly used in projects such as robotics, home automation, prototyping, IoT (Internet of Things) devices, and educational purposes. Its small size makes it ideal for projects with space constraints, and its compatibility with a wide range of sensors, actuators, and modules makes it highly versatile. Overall, the Arduino Nano is a popular choice among electronics enthusiasts and hobbyists due to its compact size, ease of use, and extensive community support.

MAX30102: The MAX30102 is an integrated optical biosensor module that combines two key functions: a photodetector and an LED light source. It is specifically designed for non-invasive heart rate and oxygen saturation (SpO2) monitoring.

The MAX30102 is commonly used in wearable fitness trackers, healthcare devices, and medical applications. Here are the main functions and features of the MAX30102:

1. Photodetector: The module contains a high-sensitivity photodetector that can detect changes in light absorption caused by variations in blood volume. It utilizes red and infrared light to measure the amount of oxygenated and deoxygenated hemoglobin in the blood.

2. LED Light Source: The MAX30102 integrates an efficient LED light source that emits red and infrared light. These lights penetrate the skin and are reflected to the photodetector, allowing the sensor to capture the desired physiological data.

3. Heart Rate Monitoring: By analyzing the changes in light absorption, the MAX30102 can accurately calculate the heart rate of an individual. This is achieved by detecting the pulsatile nature of blood flow caused by each heartbeat.

4. Oxygen Saturation (SpO2) Monitoring: Oxygen saturation refers to the percentage of oxygen-saturated hemoglobin in the blood. The MAX30102 uses the ratio of red and infrared light absorption to estimate the SpO2 level. This measurement is important for monitoring respiratory health and detecting conditions such as hypoxemia (low blood oxygen levels).

5. Integrated Signal Processing: The MAX30102 includes built-in signal processing algorithms that help filter out the noise and enhance the quality of the acquired physiological data. These algorithms contribute to reliable and accurate measurements.

6. I2C Interface: The MAX30102 communicates with microcontrollers or development boards using the I2C (Inter-Integrated Circuit) protocol, allowing for easy integration into various electronic systems.

AD8232: The AD8232 is an integrated analog front-end (AFE) module designed for bioimpedance measurement applications, particularly in the field of electrocardiography (ECG). It is commonly used in wearable health monitoring devices, fitness trackers, and medical equipment. Here are the main functions and features of the AD8232:

1. ECG Signal Conditioning: The AD8232 is designed to process and amplify weak bioelectric signals obtained from the human body, specifically the electrical activity of the heart. It provides the necessary signal conditioning to extract and amplify the ECG signal while rejecting common-mode interference and noise.

2. Lead-Off Detection: The module incorporates lead-off detection functionality, allowing it to detect if the ECG electrodes are not properly connected or if they become disconnected during measurement. This feature helps ensure the reliability and accuracy of the ECG readings.

3. Integrated Right Leg Drive (RLD): The AD8232 includes a built-in right leg drive amplifier that aids in the reduction of common-mode interference. By driving the right leg electrode, it helps to minimize noise and improve the quality of the ECG signal.

4. Low Power Consumption: The AD8232 is designed to operate efficiently with low power consumption, making it suitable for battery-powered and wearable applications.

5. Easy Integration: The module provides a convenient interface for connecting ECG electrodes and communicates with microcontrollers or development boards using standard analog or digital interfaces, such as SPI or I2C, depending on the specific version of the module.

6. Noise Filtering and Signal Processing: The AD8232 integrates various filtering techniques, such as AC and muscle artifact filtering, to remove unwanted noise and interference from the ECG signal. It also supports additional signal processing steps to improve the accuracy and reliability of the acquired data.

Li-Po Battery: Li-Po (Lithium Polymer) battery, also known as Li-Poly or LiPo battery, is a type of rechargeable battery commonly used in various electronic devices. It is a lightweight and high-energy-density battery technology that has gained popularity due to its favorable characteristics. Here are the main features and benefits of Li-Po batteries:

1. High Energy Density: Li-Po batteries offer a high energy density, meaning they can store a large amount of electrical energy in a compact size. This makes them suitable for portable electronic devices where space and weight are critical factors.

2. Lightweight: Li-Po batteries are known for their lightweight construction. They are particularly advantageous in applications where weight reduction is important, such as drones, RC models, and portable electronics.

3. Flexibility in Shape and Size: Unlike traditional cylindrical or prismatic battery cells, Li-Po batteries can be manufactured in various shapes and sizes. This flexibility in design allows them to be adapted to fit specific device requirements and can enable the creation of slim and custom-shaped battery packs.

4. High Discharge Rates: Li-Po batteries can deliver high discharge rates, which means they can provide a large amount of power in a short period. This feature makes them suitable for applications that require bursts of high energy, such as remote-controlled vehicles or high-performance gadgets.

5. Low Self-Discharge: Li-Po batteries have a relatively low self-discharge rate compared to other rechargeable battery types. They can retain their charge for a longer period when not in use, reducing the need for frequent recharging.

6. Fast Charging: Li-Po batteries generally support fast charging capabilities, allowing them to be recharged in a shorter time compared to other battery chemistries. However, it is important to follow proper charging procedures and use compatible charging equipment to ensure safety and maximize the battery's lifespan.

7. Voltage Stability: Li-Po batteries maintain a relatively stable voltage throughout their discharge cycle. This characteristic ensures a consistent power supply to the connected device, providing more predictable performance and preventing voltage drops that could affect the device's functionality.

Li-Po batteries must require careful handling and proper charging to ensure safety and prolong their lifespan. Overcharging, over-discharging, or subjecting them to extreme temperatures can lead to performance degradation, reduced capacity, or even safety hazards. Therefore, it is crucial to follow the manufacturer's guidelines and use appropriate charging and storage practices when using Li-Po batteries.

3.3 Hardware Demonstration

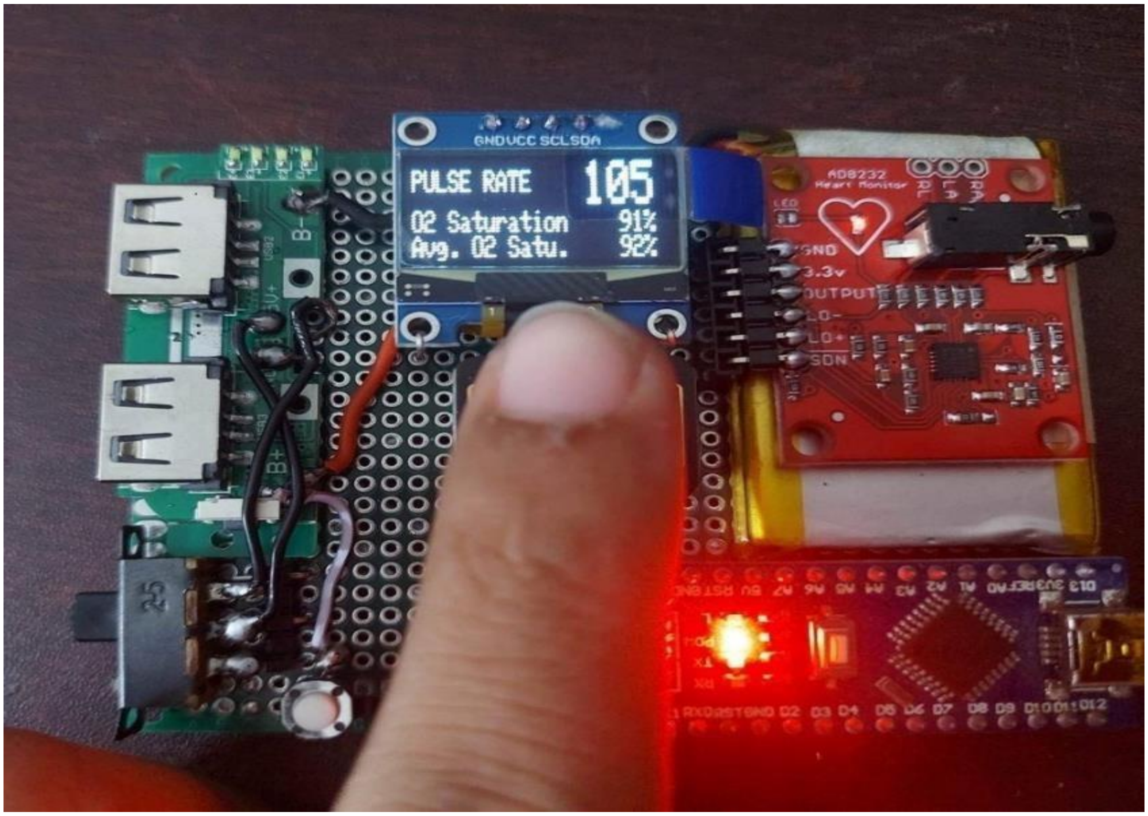

Figure 3: Prototype Hardware Implementation

3.4 Input Device

We used the LTINN Multifunctional Electronic Single Listening Stethoscope as our input device. The is a cutting-edge device designed for remote auscultation. The stethoscope features a 1-meter TPE cable that is both soft and bendable, allowing for comfortable and easy use. The stethoscope is designed for cardiopulmonary breathing, making it an ideal tool for monitoring respiratory and cardiovascular health. The device comes equipped with a 3.5mm headphone jack, making it easy to connect to a variety of headphones or other audio devices. This allows healthcare providers to listen to patients' heart and lung sounds remotely, making it an ideal tool for telemedicine applications. The LTINN Multifunctional Electronic Single Listening Stethoscope is a versatile device that can be used in a variety of settings, including hospitals, clinics, and remote healthcare facilities. It is designed to provide high-quality audio and reliable performance, making it a valuable tool for healthcare providers who need to monitor patients remotely.

This input device can take the necessary measures of heart rate by plugging this device into the phone and connecting it to the app. With this input device, healthcare professionals to remotely listen to a patient's heart, lung, and other body sounds. It combines traditional auscultation with digital technology, allowing the capture, amplification, and transmission of audio signals to a remote location for interpretation and diagnosis.

Figure 4: Primary Input Device for heart rate

3.5 Prototype App Blueprint

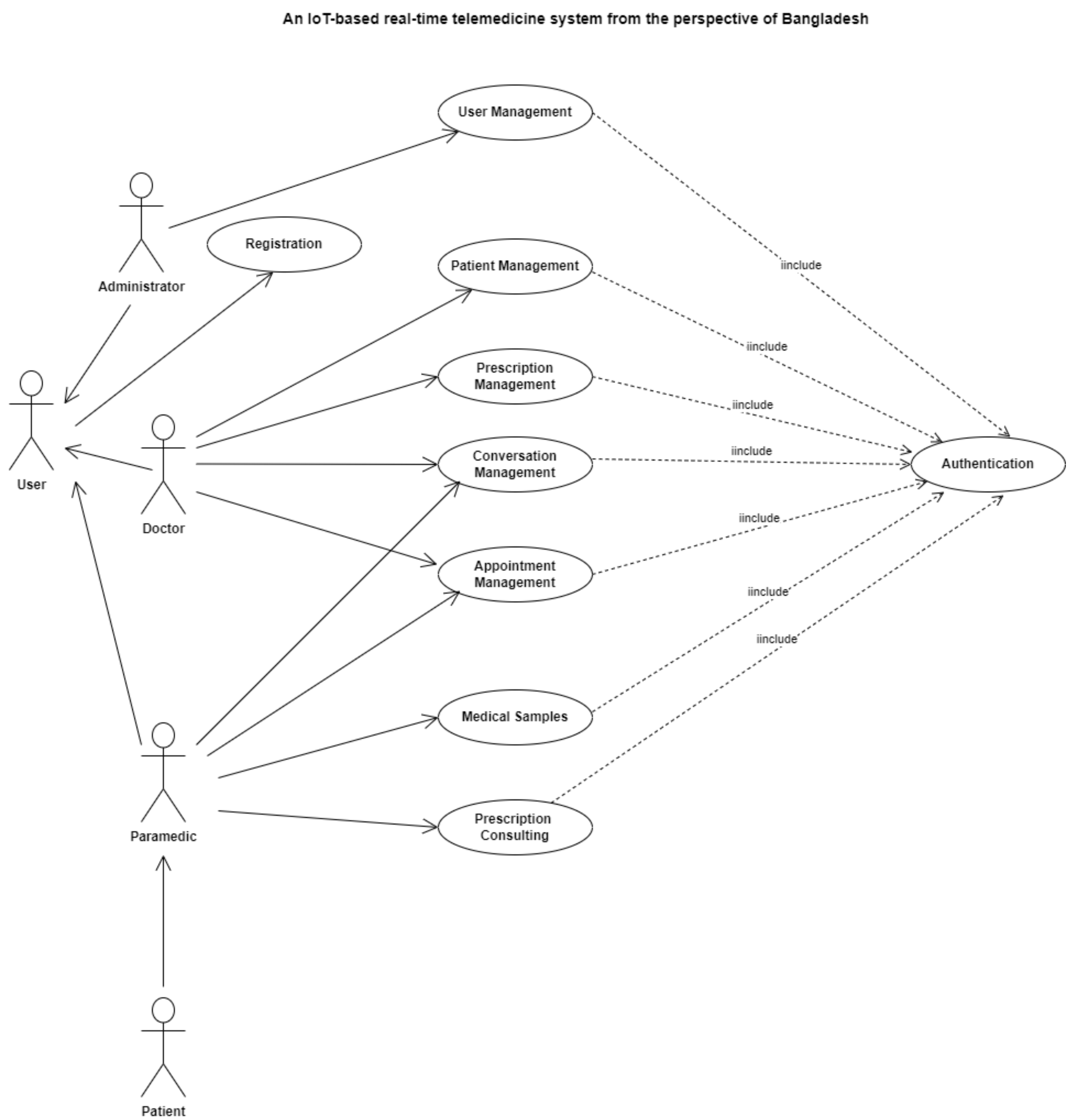

Figure 5: UML Diagram of the App

3.5 App Demonstration:

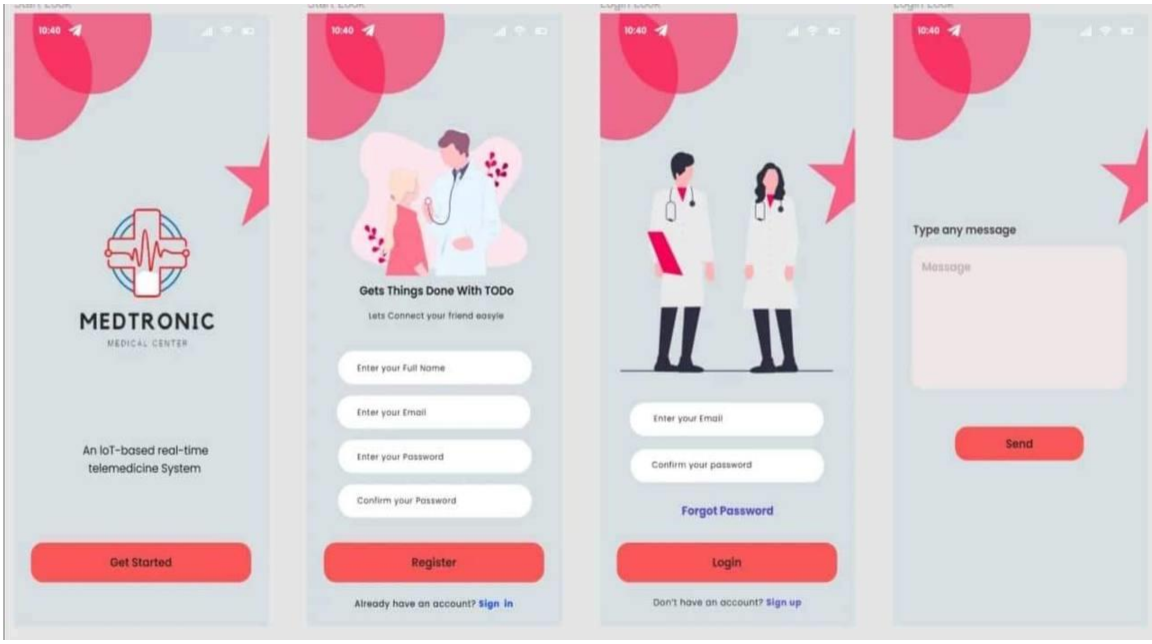

Figure 6: Prototype App Demonstration

Here are some key points about the app:

1. **Remote patient monitoring:** With this app, it allows healthcare providers to remotely monitor a patient’s vital signs and assess their respiratory and cardiovascular health without being physically present. This app is particularly useful for patients in remote locations or those who cannot easily access healthcare facilities.

2. **Real-Time Teleconsultation:** During a teleconsultation, a healthcare provider can use a digital stethoscope and this app to listen to a patient’s heart and lung sounds in real time. The captured audio can be transmitted over secure communication channels to the healthcare professional for immediate analysis and diagnosis.

3. **Store and Forward:** This app records and stores data for later review. This feature allows physicians to listen to the recorded sounds at their convenience, aiding in accurate diagnosis and facilitating second opinions from specialists if needed.

CHAPTER FOUR

Experimental Results

4.1 Results

A low-pass filter is an electronic circuit or digital signal processing technique that allows low-frequency signals to pass through while attenuating or blocking higher-frequency signals. It is commonly used in various applications to remove or reduce unwanted high-frequency components from a signal. Here are some key points about low-pass filters:

Function: The primary function of a low-pass filter is to pass signals with frequencies below a certain cutoff frequency (Fc) while attenuating or blocking signals above that frequency. It allows the passage of lower-frequency components while filtering out higher-frequency components.

Cutoff Frequency: The cutoff frequency (Fc) of a low-pass filter is the frequency at which the filter starts attenuating the signal. It is typically defined as the point at which the output signal is reduced by 3 dB (half the power) compared to the low-frequency region. Frequencies below the cutoff frequency are referred to as the passband, while frequencies above it are in the stopband.

Filter Characteristics: Low-pass filters can have different characteristics depending on their design and application. Common characteristics include the filter's slope or roll-off rate, which describes how quickly the filter attenuates frequencies above the cutoff point. Filter types such as Butterworth, Chebyshev, and Bessel have different trade-offs between sharp roll-off and passband ripple.

Analog and Digital Filters: Low-pass filters can be implemented using analog circuitry or digital signal processing algorithms. Analog filters typically use passive components (resistors, capacitors, and inductors) or active components (such as operational amplifiers) to shape the frequency response. Digital filters are implemented in software or digital signal processors and operate on discrete-time signals.

MATLAB is an ideal tool for plotting and filtering data before implementing it in telemedicine systems. Its powerful data analysis, signal processing, and visualization tools make it easier to analyze and interpret biological signals, improving the accuracy and reliability of the data. Its customization options also make it an ideal tool for developing custom telemedicine systems that meet the unique needs of different healthcare providers and patients. In Matlab, a low pass filter can be implemented using the "filter" function and the "designfilt" function. The "designfilt" function is used to design the filter coefficients based on the desired filter characteristics, such as the cut-off frequency and the filter order. The "filter" function is then used to apply the filter to the input signal.

4.2 Low Pass Filter:

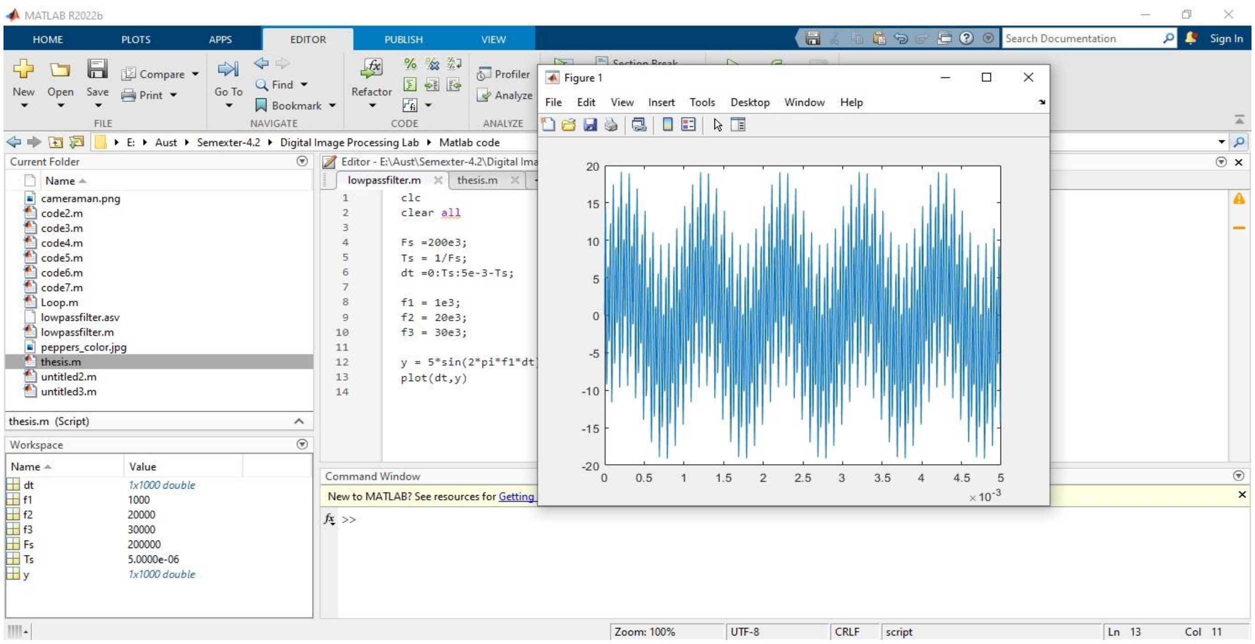

Figure 7: Input Signal

The low pass filter works by attenuating or blocking high-frequency components of the input signal, while allowing low-frequency components to pass through. The cut-off frequency determines the frequency at which the filter begins to attenuate the signal. In this example, the low pass filter will attenuate signals above 50 Hz, while allowing signals below 50 Hz to pass through.

We achieved this signal from the LTINN Multifunctional Electronic Single Listening Stethoscope that we used as our input device.

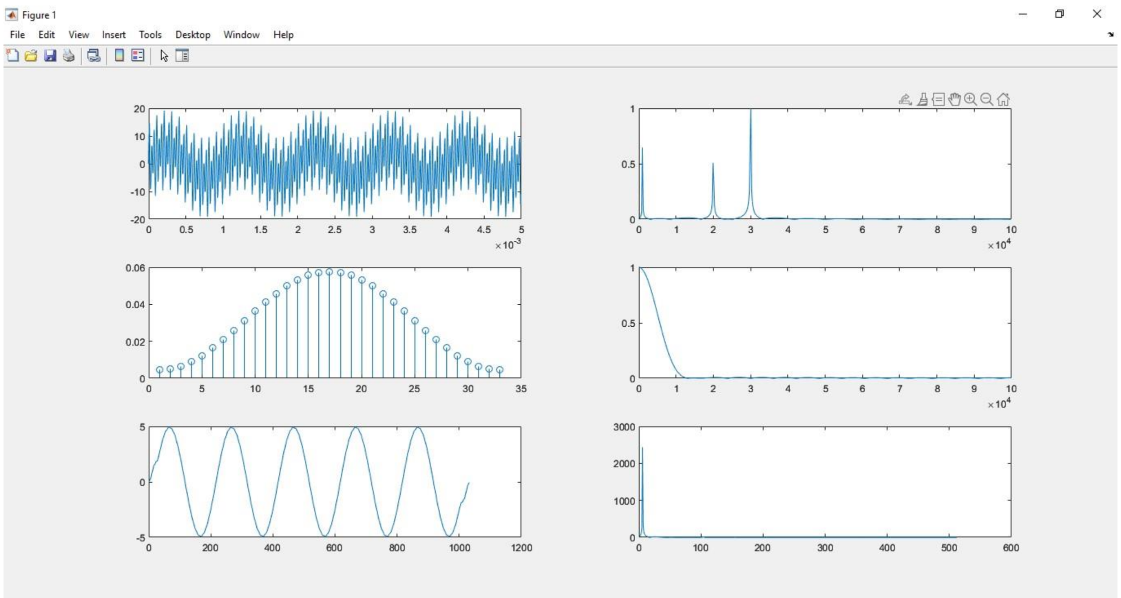

Figure 8: Output signal after low pass filter

The output signal will typically have a lower amplitude than the input signal, since the filter is designed to remove high-frequency noise and interference that may have been present in the original signal. The amount of attenuation will depend on the filter characteristics, such as the filter order and the cut-off frequency. The output signal may also have a slight delay compared to the input signal, since the filter introduces a phase shift in the signal. The amount of delay will depend on the filter settings and the filter order, with higher order filters typically introducing more delay than lower-order filters.

After removing unnecessary noise, we get this output signal. We used this low pass filter in our app to reduce extra noise from the stethoscope. It characterizes the sensor data, removes noise, enhances the signal, and transmits the stethoscope signal over cell phone voice service and Ethernet. It compares the quality of the two received signals at the doctor's end. Healthcare professionals then collect and analyze data to determine and diagnose the diseases.

4.3 ECG Filters

Low pass filters are commonly used in telemedicine systems to filter out high-frequency noise and interference from biological signals, such as electrocardiogram (ECG) or electroencephalogram (EEG) signals. In telemedicine, ECG signals are often transmitted over long distances or through wireless networks, which can introduce noise and interference. The low pass filter is used to remove the high-frequency noise and interference from the ECG signal, which could otherwise interfere with the analysis and interpretation of the signal. A low pass filter works by allowing low-frequency signals to pass through while attenuating or blocking high-frequency signals. The filter is designed to have a cut-off frequency, which is the frequency at which the filter begins to attenuate the signal. For example, a low pass filter with a cut-off frequency of 50 Hz would allow signals below 50 Hz to pass through while attenuating signals above 50 Hz. In telemedicine systems, the low pass filter is typically applied to the ECG signal before it is transmitted or recorded. The filter is designed to remove high-frequency noise and interference while preserving the low-frequency components of the ECG signal that are clinically relevant.

ECG R Peak: In electrocardiography (ECG), the R peak refers to the highest positive deflection in the ECG waveform. It represents the depolarization of the ventricles of the heart during a cardiac cycle. The R peak is a significant feature of the ECG signal and is used to measure several important parameters related to heart function. Here are some key points about the R peak in an ECG:

1. Location: The R peak is typically observed as a distinct upward deflection in the ECG waveform. It occurs after the QRS complex, which consists of the Q wave, R wave, and S wave.

2. Amplitude: The amplitude of the R peak corresponds to the magnitude of the electrical potential generated by ventricular depolarization. It can vary depending on factors such as electrode placement and the strength of the electrical signal.

3. R-R Interval: The distance between consecutive R peaks in an ECG is known as the R-R interval. It represents the duration of one cardiac cycle and is used to calculate heart rate

(number of beats per minute). By measuring the time between R peaks, abnormalities in heart rhythm, such as arrhythmias, can be identified.

4. QRS Complex: The R peak is part of the QRS complex, which reflects the electrical activity associated with ventricular depolarization. The QRS complex includes the Q wave (first negative deflection), the R wave (first positive deflection), and the S wave (negative deflection after the R wave).

5. Analysis: The characteristics of the R peak, such as its amplitude, shape, and duration, are analyzed to assess various cardiac parameters. These parameters include heart rate, heart rhythm, conduction abnormalities, and the presence of conditions like myocardial infarction (heart attack) or ventricular hypertrophy.

6. Measurement: ECG analysis software or medical professionals often use automated algorithms or manual measurements to determine the location and amplitude of the R peak accurately. This information is crucial for diagnosing and monitoring heart conditions.

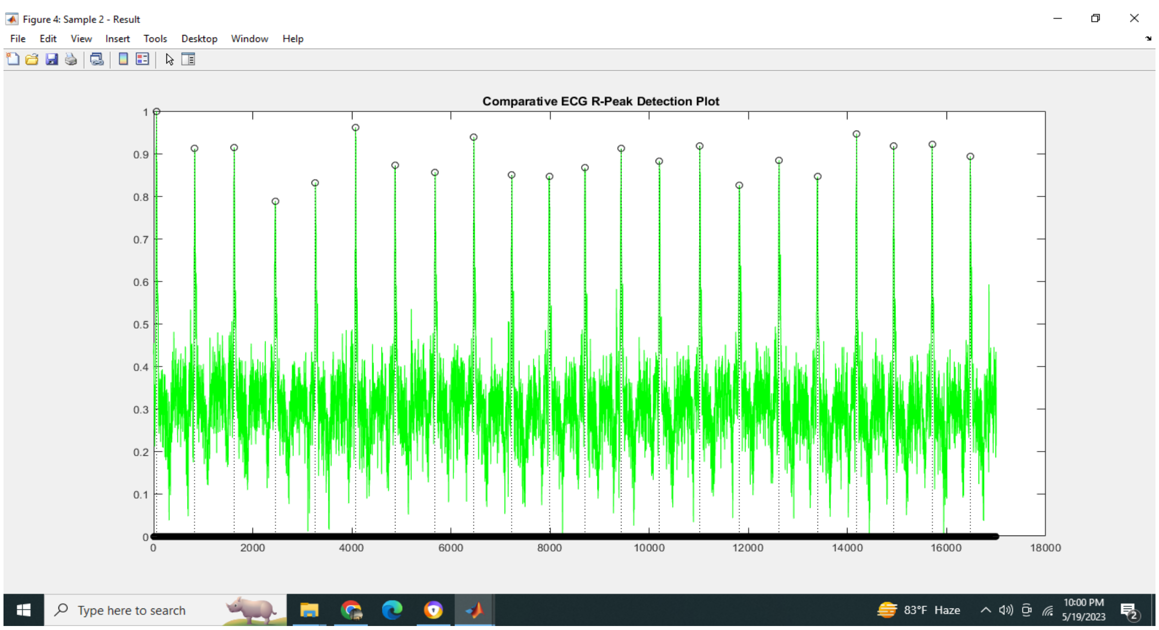

Figure 9: Comparative ECG R-Peak Detection Plot

Average Heart Rate: Calculating the average heart rate from an ECG plot in Matlab involves preprocessing the signal, detecting the QRS complex, and measuring the time interval between successive QRS complexes.

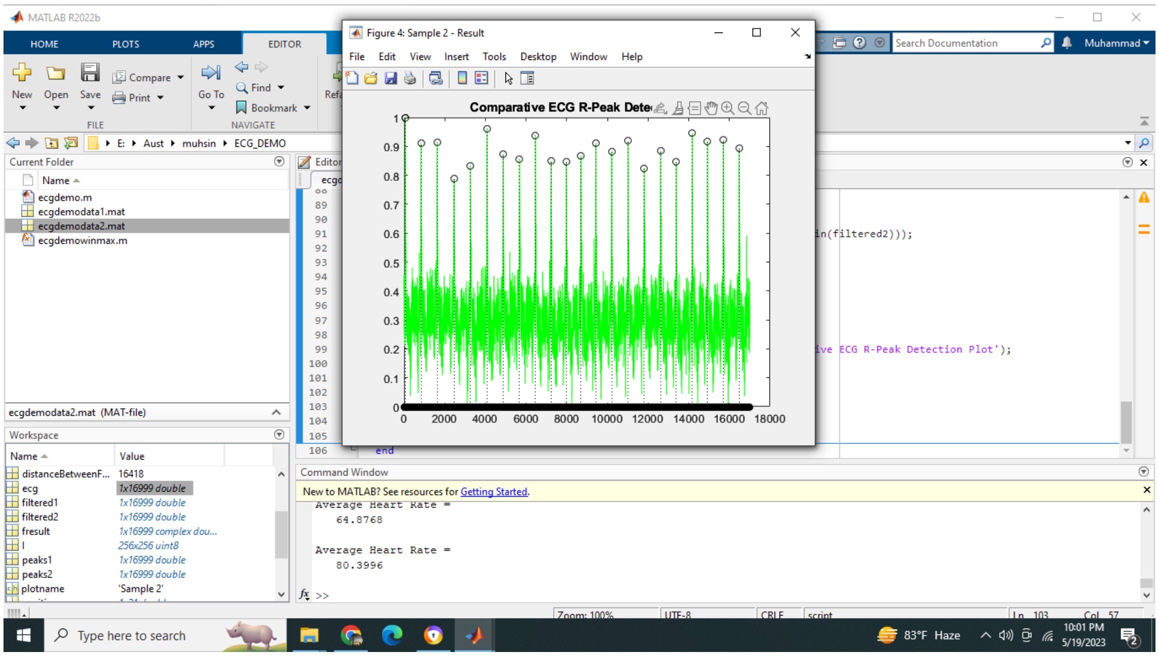

Figure 10: Average Heart Rate from ECG plot

Original ECG and Filtered ECG: The original ECG signal represents the raw electrical activity of the heart and may contain noise and artifacts that can interfere with the analysis and interpretation of the signal. The filtered ECG signal is a processed version of the original signal that has been filtered to remove high-frequency noise and interference. The filtered ECG signal is typically smoother and easier to interpret than the original ECG signal.

By comparing the original ECG and the filtered ECG signals, it is possible to visualize the effect of the filtering process and assess the quality of the filtered signal. In general, a well-designed filter will remove high-frequency noise and interference while preserving the low-frequency components of the ECG signal that are clinically relevant.

Figure 11: The original ECG and Filtered ECG plot

Heart rate after Applying the Filter: The heart rate calculated after applying an ECG filter in Matlab should be more accurate and reliable than the heart rate calculated from the raw ECG signal. This is because the filtering process removes high-frequency noise and interference that can affect the accuracy of the heart rate calculation. The resulting heart rate can be used to assess the patient's cardiac function and monitor changes in heart rate over time.

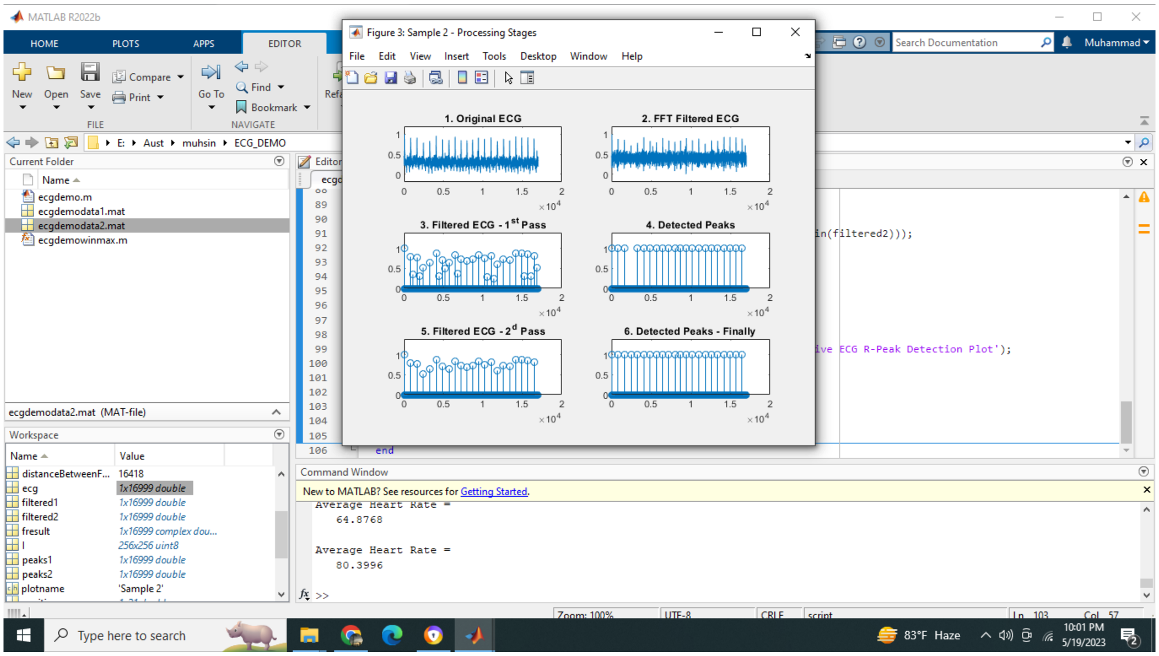

Figure 12: Heart rate signal after applying filter.

ECG Plot Comparison: The ECG signal is loaded, and a low pass filter is applied to the signal using the "lowpass" function. The original ECG signal and the detected R-peaks are then plotted using the "subplot" function to create four separate plots in the same figure. The first plot shows the original ECG signal. The second, third, and fourth plots show the ECG signal with the detected R-peaks overlaid in red for the three different algorithms or settings. By comparing the different plots, it is possible to assess the performance of the different R-peak detection algorithms or settings and choose the one that provides the most accurate and reliable results.

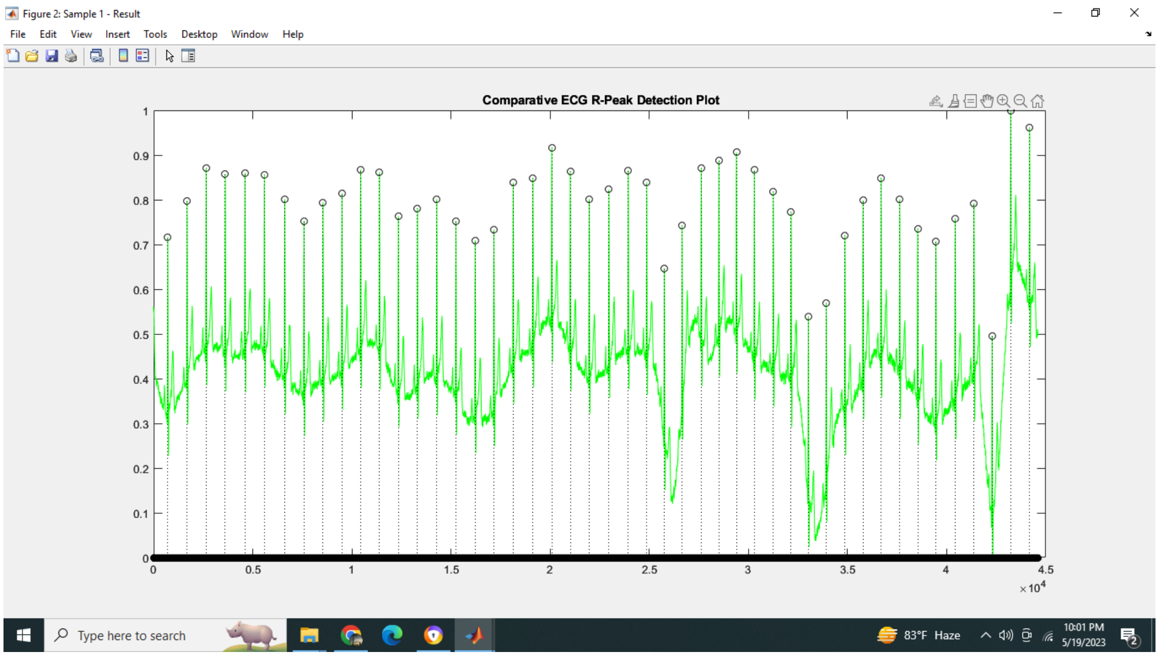

Figure 13: Comparative ECG R-Peak Detection Plot

FFT Filtered Plot: In Matlab, the Fast Fourier Transform (FFT) can be used to analyze the frequency content of a filtered ECG signal. The resulting plot shows the frequency content of the filtered ECG signal, with the x-axis representing frequency in Hertz and the y-axis representing the magnitude of the signal at each frequency. The FFT plot can be used to identify the frequency components of the signal and assess the effectiveness of the filtering process in removing high-frequency noise and interference.

Figure 14: FFT filtered ECG signal.

Processing Stage:

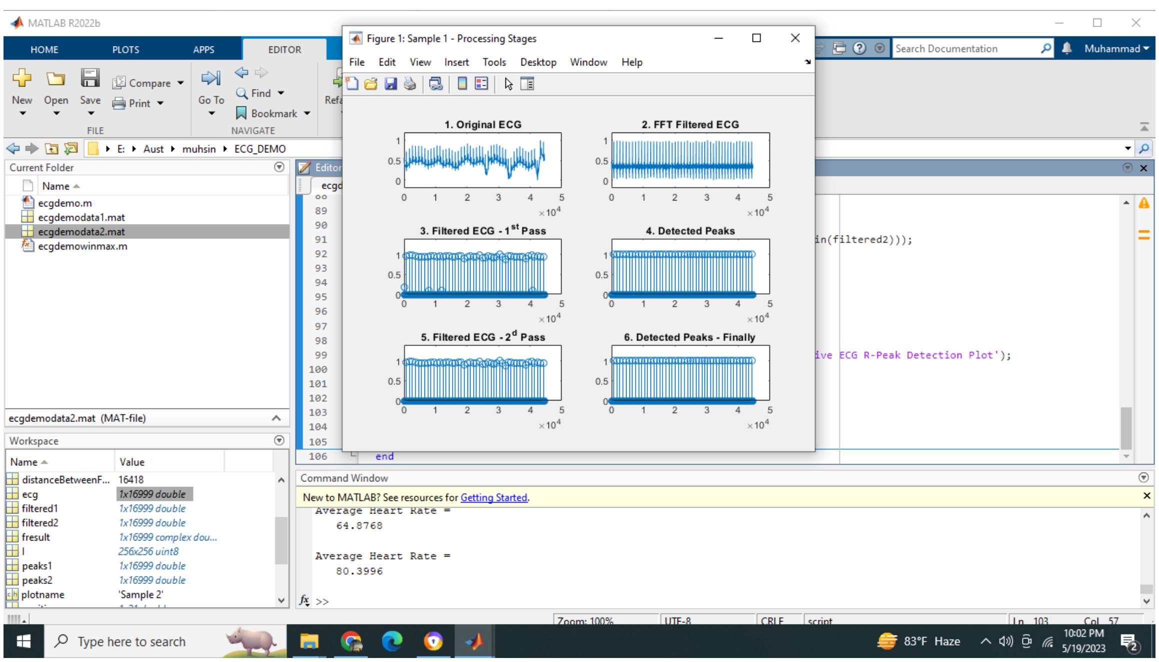

Figure 15: Filtered ECG

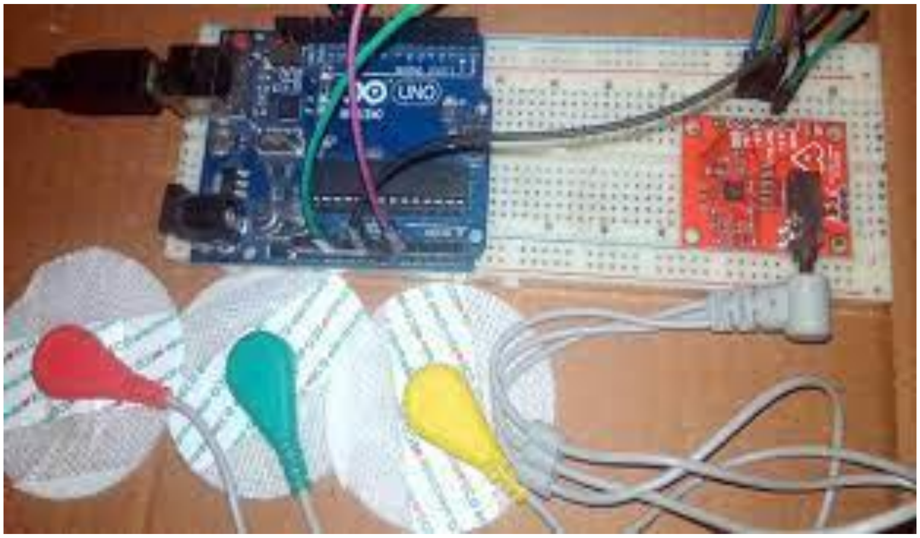

Figure 16: AD8232 ECG Sensor & Arduino Interfacing with ECG Graph

4.4 Applying low pass filter in our android app:

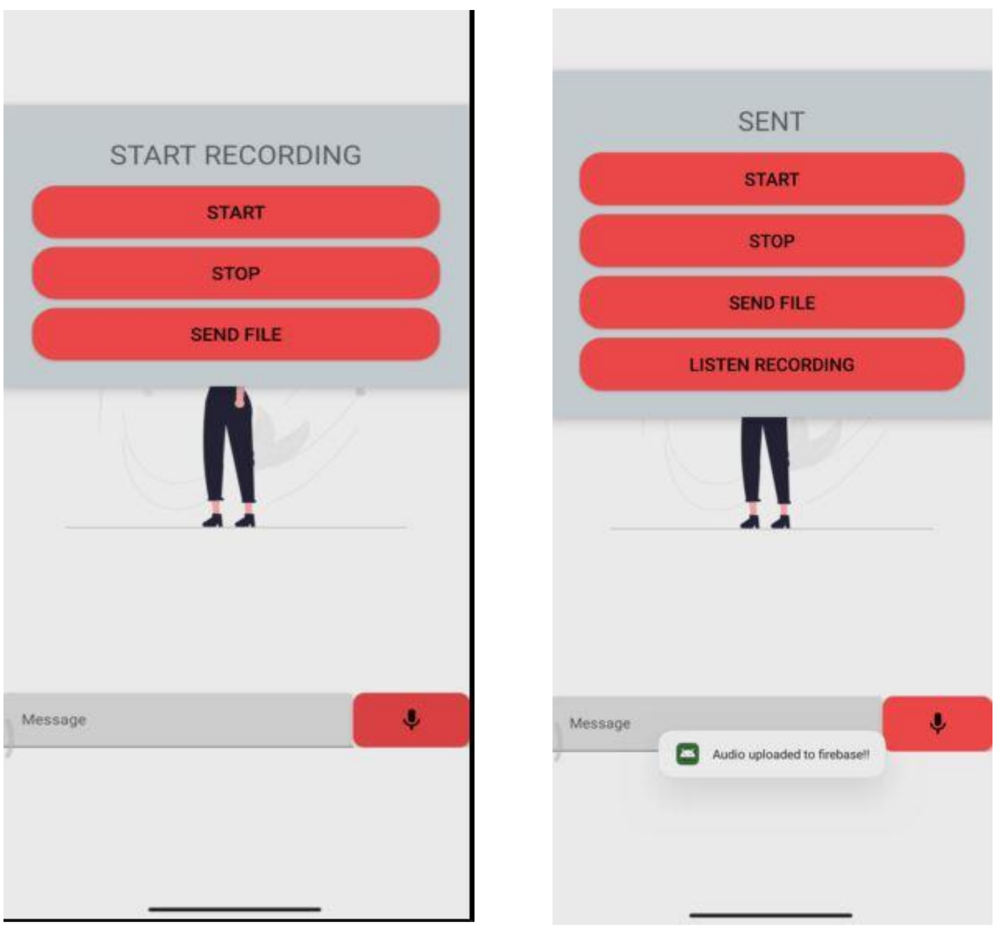

Figure 17: Audio transmission system

We applied a low pass filter in an audio signal in your Android app by using the Android AudioTrack and AudioRecord classes. Firstly, we created an AudioRecord object to capture the audio data. Then applied an AudioTrack object to play back the filtered audio data. After that, we can easily initialize a buffer for the audio data. Then we applied the low pass filter to the audio data. Then passed the filtered audio data to the AudioTrack object for playback. And finally, we store that signal in our Firebase database for future use and real-time transmission.

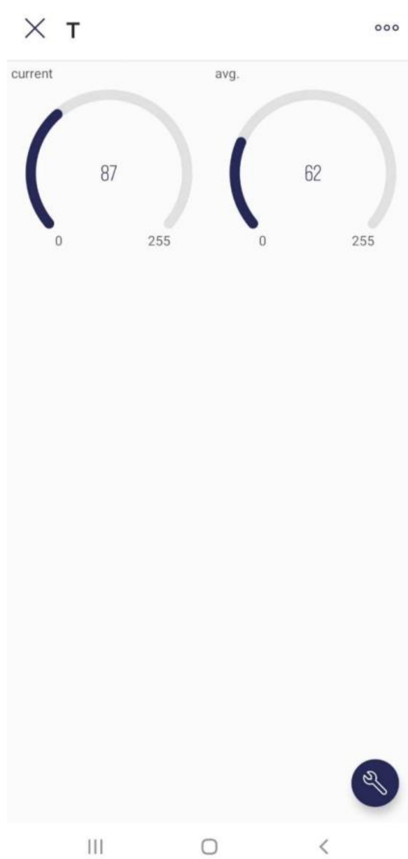

Figure 18: SPo2 Sensor Output in the app

We get this output directly from the SPo2 sensors through the database to show in our app.

CHAPTER FIVE

Conclusion And Future Work

5.1 Conclusion

In rural areas, people suffer a lot from the lack of proper health care and proper medical care service. This project will help people from rural areas to get the exact healthcare guide they need and meet the need for a professional doctor in a rural area. It gives efficient and low-cost medical care service at home. It is easy to use and gives accurate medical data such as heartbeat, heart rate, and oxygen saturation and it also cancels the extra noise so that the data are accurate. By using this app and project people in the rural area get better communication with professional doctors and people also save valuable time. This app stores the patient's data in the database. So, the application helps people to get better and digital real-time service from home.

In conclusion, this thesis on telemedicine will show promising results in improving healthcare accessibility and quality, especially in rural and remote areas. Telemedicine technology has enabled healthcare providers to remotely diagnose and treat patients, reducing the need for in-person visits and improving patient outcomes. However, there are still some challenges that need to be addressed, such as ensuring patient privacy and data security, as well as overcoming technical barriers in areas with poor internet connectivity. With continued advancements in telemedicine technology and more widespread adoption, it is expected that telemedicine will play an increasingly important role in the healthcare industry, providing patients with convenient and cost-effective access to quality healthcare services.

5.2 Future Work

We will implement Machine learning algorithms that will call emergency services (e.g., ambulance, nurse). A sudden spike in the ECG signal or anomaly in collected data will trigger the emergency alert. Map API will be used to find nearby medical assistance or diagnostic centers. Using Arduino sensors to record diabetes data can be a useful and convenient way to monitor blood glucose levels and other important health metrics. We will also try to implement Glucometer sensor to record diabetes data, patients can have a better understanding of their health and make more informed decisions about their treatment. The data can be easily stored and shared with healthcare providers, allowing for more personalized and effective care.

REFERENCES

1. M.Shamim Hossaina Ghulam Muhammad “Cloud-assisted Industrial Internet of Things (IIoT) – Enabled framework for health monitoring”.

2. Shyamal Patel “A review of wearable sensors and systems with application in rehabilitation” Journal of Neuro Engineering and Rehabilitation Northeastern University.

3. U. Satija, B. Ramkumar, and M. S. Manikandan, “Real-time signal quality-aware ecg telemetry system for iot-based health care monitor- ing,” IEEE Internet of Things Journal, vol. PP, no. 99, pp. 1–1, 2017.

APPENDIX

AD8232 ECG Sensor

Source Code:

```
int ECG_LO_PLUS_PIN = 10;
int ECH_LO_MINUS_PIN = 11;
int ECH_OUTPUT_PIN  = A0;
int ECG_SENSOR_VALUE;

#include "ssd1306h.h"
#include "MAX30102.h"
#include "Pulse.h"
#include <avr/pgmspace.h>
#include <EEPROM.h>
#include <avr/sleep.h>

// Routines to clear and set bits
#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif


SSD1306 oled;
MAX30102 sensor;
Pulse pulseIR;
Pulse pulseRed;
MAFilter bpm;

#define LED LED_BUILTIN
#define BUTTON 3
#define OPTIONS 7

static const uint8_t heart_bits[] PROGMEM = { 0x00, 0x00, 0x38, 0x38,
0x7c, 0x7c, 0xfe, 0xfe, 0xfe, 0xff,0xfe, 0xff, 0xfc, 0x7f, 0xf8, 0x3f,
0xf0, 0x1f, 0xe0, 0x0f,0xc0, 0x07, 0x80, 0x03, 0x00, 0x01, 0x00, 0x00,
0x00, 0x00,0x00, 0x00 };

//spo2_table is approximated as  -45.060*ratioAverage* ratioAverage +
30.354 *ratioAverage + 94.845 ;
const uint8_t spo2_table[184] PROGMEM =
{ 95, 95, 95, 96, 96, 96, 97, 97, 97, 97, 97, 98, 98, 98, 98, 98, 99, 99,
99, 99,99, 99, 99, 99, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,
100, 100, 100, 100, 100, 100,100, 100, 100, 100, 99, 99, 99, 99, 99, 99,
99, 99, 98, 98, 98, 98, 98, 98, 97, 97, 97, 97, 96, 96, 96, 96, 95, 95,
95, 94, 94, 94, 93, 93, 93, 92, 92, 92, 91, 91,90, 90, 89, 89, 89, 88, 88,
87, 87, 86, 86, 85, 85, 84, 84, 83, 82, 82, 81, 81,80, 80, 79, 78, 78, 77,
76, 76, 75, 74, 74, 73, 72, 72, 71, 70, 69, 69, 68, 67, 66, 66, 65, 64,
63, 62, 62, 61, 60, 59, 58, 57, 56, 56, 55, 54, 53, 52, 51, 50, 49, 48,
```

```
47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 31, 30, 29,
28, 27, 26, 25, 23, 22, 21, 20, 19, 17, 16, 15, 14, 12, 11, 10, 9, 7, 6,
5, 3, 2, 1 } ;


int getVCC() {
  //reads internal 1V1 reference against VCC
#if defined(__AVR_ATmega1284P__)
  ADMUX = _BV(REFS0) | _BV(MUX4) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);  //
For ATmega1284
#else
  ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);  // For
ATmega328
#endif
  delay(2); // Wait for Vref to settle
  ADCSRA |= _BV(ADSC); // Convert
  while (bit_is_set(ADCSRA, ADSC));
  uint8_t low = ADCL;
  unsigned int val = (ADCH << 8) | low;
  //discard previous result
  ADCSRA |= _BV(ADSC); // Convert
  while (bit_is_set(ADCSRA, ADSC));
  low = ADCL;
  val = (ADCH << 8) | low;

  return (((long)1024 * 1100) / val) / 100;
}

void print_digit(int x, int y, long val, char c = ' ', uint8_t field = 3,
const int BIG = 2)
{
  uint8_t ff = field;
  do {
    char ch = (val != 0) ? val % 10 + '0' : c;
    oled.drawChar( x + BIG * (ff - 1) * 6, y, ch, BIG);
    val = val / 10;
    --ff;
  } while (ff > 0);
}


/*
     Record, scale  and display PPG Wavefoem
*/
const uint8_t MAXWAVE = 72;

class Waveform {
  public:
    Waveform(void) {
      wavep = 0;
    }

    void record(int waveval) {
      waveval = waveval / 8;       // scale to fit in byte
```

```
      waveval += 128;              //shift so entired waveform is +ve
      waveval = waveval < 0 ? 0 : waveval;
      waveform[wavep] = (uint8_t) (waveval > 255) ? 255 : waveval;
      wavep = (wavep + 1) % MAXWAVE;
    }

    void scale() {
      uint8_t maxw = 0;
      uint8_t minw = 255;
      for (int i = 0; i < MAXWAVE; i++) {
        maxw = waveform[i] > maxw ? waveform[i] : maxw;
        minw = waveform[i] < minw ? waveform[i] : minw;
      }
      uint8_t scale8 = (maxw - minw) / 4 + 1; //scale * 8 to preserve
precision
      uint8_t index = wavep;
      for (int i = 0; i < MAXWAVE; i++) {
        disp_wave[i] = 31 - ((uint16_t)(waveform[index] - minw) * 8) /
scale8;
        index = (index + 1) % MAXWAVE;
      }
    }

    void draw(uint8_t X) {
      for (int i = 0; i < MAXWAVE; i++) {
        uint8_t y = disp_wave[i];
        oled.drawPixel(X + i, y);
        if (i < MAXWAVE - 1) {
          uint8_t nexty = disp_wave[i + 1];
          if (nexty > y) {
            for (uint8_t iy = y + 1; iy < nexty; ++iy)
              oled.drawPixel(X + i, iy);
          }
          else if (nexty < y) {
            for (uint8_t iy = nexty + 1; iy < y; ++iy)
              oled.drawPixel(X + i, iy);
          }
        }
      }
    }

  private:
    uint8_t waveform[MAXWAVE];
    uint8_t disp_wave[MAXWAVE];
    uint8_t wavep = 0;

} wave;

int  beatAvg;
int  SPO2, SPO2f;
int  voltage;
bool filter_for_graph = false;
bool draw_Red = false;
uint8_t pcflag = 0;
```

```
uint8_t istate = 0;
uint8_t sleep_counter = 0;

void button(void) {
  pcflag = 1;
}

void checkbutton() {
  if (pcflag && !digitalRead(BUTTON)) {
    istate = (istate + 1) % 4;
    filter_for_graph = istate & 0x01;
    draw_Red = istate & 0x02;
    EEPROM.write(OPTIONS, filter_for_graph);
    EEPROM.write(OPTIONS + 1, draw_Red);
  }
  pcflag = 0;
}



void Display_5() {
  if (pcflag && !digitalRead(BUTTON)) {
    draw_oled(5);
    delay(1100);
  }
  pcflag = 0;


}

void go_sleep() {
  oled.fill(0);
  oled.off();
  delay(10);
  sensor.off();
  delay(10);
  cbi(ADCSRA, ADEN);  // disable adc
  delay(10);
  pinMode(0, INPUT);
  pinMode(2, INPUT);
  set_sleep_mode(SLEEP_MODE_PWR_DOWN);
  sleep_mode();  // sleep until button press
  // cause reset
  setup();
}

void draw_oled(int msg) {
  oled.firstPage();
  do {
    switch (msg) {
      case 0:  oled.drawStr(10, 0, F("Device error"), 1);
        break;
      case 1:
       oled.drawStr(0, 0, "ECG : ");
```

```
        oled.drawStr(0, 20, F("PLACE FINGER for More info"), 1);



        print_digit(30, 0, ECG_SENSOR_VALUE, ' ', 3, 1);


        oled.drawChar(100, 0, voltage / 10 + '0');
        oled.drawChar(106, 0, '.');
        oled.drawChar(112, 0, voltage % 10 + '0');
        oled.drawChar(118, 0, 'V');

        break;

      case 2:

        oled.drawStr(0, 3, F("PULSE RATE"), 1);
        print_digit(86, 0, beatAvg);
        //wave.draw(8);
        oled.drawStr(0, 16, F("O2 Saturation"), 1);
        print_digit(98, 16, SPO2f, ' ', 3, 1);

        oled.drawStr(0, 24, F("Avg. O2 Satu."), 1);
        print_digit(98, 24, SPO2, ' ', 3, 1);
        oled.drawChar(116, 16, '%');

        oled.drawChar(116, 24, '%');
        break;
      case 3:  oled.drawStr(45, 9, F("AUST "), 2);

        //oled.drawXBMP(6,8,16,16,heart_bits);

        break;
      case 4:  oled.drawStr(28, 12, F("OFF IN"), 1);
        oled.drawChar(76, 12, 10 - sleep_counter / 10 + '0');
        oled.drawChar(82, 12, 's');
        break;
      case 5:  oled.drawStr(0, 0, F("BMP:"), 1);
        print_digit(25, 0, beatAvg);
        oled.drawStr(0, 15, F("SpO2:"), 1);
        print_digit(25, 15, SPO2);
        oled.drawXBMP(106, 8, 16, 16, heart_bits);
        break;
    }
  } while (oled.nextPage());
}

void setup(void) {

  Serial.begin(115200);
  pinMode(ECG_LO_PLUS_PIN, INPUT); // Setup for leads off detection LO +
  pinMode(ECH_LO_MINUS_PIN, INPUT); // Setup for leads off detection LO -
```

```
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT_PULLUP);
  filter_for_graph = EEPROM.read(OPTIONS);
  draw_Red = EEPROM.read(OPTIONS + 1);
  oled.init();
  oled.fill(0x00);
  draw_oled(3);
  delay(3000);
  if (!sensor.begin())  {
    draw_oled(0);
    while (1);
  }
  sensor.setup();
  attachInterrupt(digitalPinToInterrupt(BUTTON), button, CHANGE);
}

long lastBeat = 0;    //Time of the last beat
long displaytime = 0; //Time of the last display update
bool led_on = false;


void loop()  {
  sensor.check();
  long now = millis();   //start time of this cycle
  if (!sensor.available()) return;
  uint32_t irValue = sensor.getIR();
  uint32_t redValue = sensor.getRed();
  sensor.nextSample();
  if (irValue < 5000) {
    voltage = getVCC();
    checkbutton();
    draw_oled(sleep_counter <= 50 ? 1 : 4); // finger not down message

    delay(200);
    ++sleep_counter;
    if (sleep_counter > 100) {
      go_sleep();
      sleep_counter = 0;
    }
  } else {
    sleep_counter = 0;

    int16_t IR_signal, Red_signal;
    bool beatRed, beatIR;
    if (!filter_for_graph) {
      IR_signal =  pulseIR.dc_filter(irValue) ;
      Red_signal = pulseRed.dc_filter(redValue);
      beatRed = pulseRed.isBeat(pulseRed.ma_filter(Red_signal));
      beatIR =  pulseIR.isBeat(pulseIR.ma_filter(IR_signal));
    } else {
      IR_signal =  pulseIR.ma_filter(pulseIR.dc_filter(irValue)) ;
      Red_signal = pulseRed.ma_filter(pulseRed.dc_filter(redValue));
      beatRed = pulseRed.isBeat(Red_signal);
```

```
      beatIR =  pulseIR.isBeat(IR_signal);
    }
    // invert waveform to get classical BP waveshape
    wave.record(draw_Red ? -Red_signal : -IR_signal );
    // check IR or Red for heartbeat
    if (draw_Red ? beatRed : beatIR) {
      long btpm = 60000 / (now - lastBeat);
      if (btpm > 0 && btpm < 200) beatAvg = bpm.filter((int16_t)btpm);
      lastBeat = now;
      digitalWrite(LED, HIGH);
      led_on = true;
      // compute SpO2 ratio
      long numerator   = (pulseRed.avgAC() * pulseIR.avgDC()) / 256;
      long denominator = (pulseRed.avgDC() * pulseIR.avgAC()) / 256;
      int RX100 = (denominator > 0) ? (numerator * 100) / denominator :
999;
      // using formula
      SPO2f = (10400 - RX100 * 17 + 50) / 100;
      // from table
      if ((RX100 >= 0) && (RX100 < 184))
        SPO2 = pgm_read_byte_near(&spo2_table[RX100]);
    }
    // update display every 50 ms if fingerdown
    if (now - displaytime > 50) {
      displaytime = now;
      wave.scale();
      draw_oled(2);

    }
    Display_5();
  }
  // flash led for 25 ms
  if (led_on && (now - lastBeat) > 25) {
    digitalWrite(LED, LOW);
    led_on = false;
  }

  if ((digitalRead(ECG_LO_PLUS_PIN) == 1) ||
(digitalRead(ECH_LO_MINUS_PIN) == 1)) {
    Serial.println('!');
  }
  else {
    ECG_SENSOR_VALUE = analogRead(ECH_OUTPUT_PIN);
    // send the value of analog input 0:
    Serial.println(ECG_SENSOR_VALUE);
  }
}
```