Worcester Polytechnic Institute
Major Qualifying Project Report

WIRELESS WEARABLE SENSOR FOR BIOMETRIC DATA EXTRACTION FROM REFLECTIVE PPG WAVEFORM WITH REALTIME DATABASE

May 18, 2020

Advisor

Professor Ulkuhan Guler
uguler@wpi.edu
Undergraduate Students
Franco Baudino
Electrical and Computer Engineering
fabaudino@wpi.edu
Bill Chieng
Electrical and Computer Engineering
bchieng@wpi.edu
Fivos Kavassalis
Electrical and Computer Engineering
fikavassalis@wpi.edu

Contents

Abstract

Our project designed a wearable device that uses photoplethysmogram (PPG) waveform to measure and estimate various vital signs via custom-designed algorithms running on a mobile application. These vital signs include peripheral oxygen saturation, heart rate, respiratory rate and short/long term heart rate variability. The device wirelessly transmits accumulated data to a mobile phone and/or a personal computer over Bluetooth Low Energy (BLE). A mobile application was developed that can receive, process, and display the extracted vital sign information. This project explores an emerging technology of wearable biomedical devices with the addition of the Internet of Things (IoT) element. Possible applications are varied; however, this wireless monitoring system would allow health providers to more effectively monitor outpatient vital signs. In the wake of the ongoing Coronavirus Disease 19 (COVID-19) pandemic, the measurement of peripheral oxygen saturation would give an early warning of degrading respiratory health before the apparent manifestation of symptoms. The convenient use of the device in a mobile setting is especially relevant to current isolation precautions in place, in addition to its critical role in the improvement of the care of at-risk patients.

Executive Summary

The human body requires a steady supply of oxygen to maintain the health and proper functioning of organs. A lack of oxygen supply eventually leads to tissue damage and can result in death. Peripheral capillary oxygen saturation (SpO_2) is an estimate of the percentage of oxygenated hemoglobin in a person's blood and is considered to be a vital sign. A typical SpO_2 reading for a healthy person falls anywhere between 94% and 100%. If a healthy person's SpO_2 reading falls below 94%, medical intervention becomes necessary. This makes accurate and continuous SpO_2 measurement an important field of study.

There are multiple ways of measuring the saturation of oxygen in a patient's blood. The arterial blood gas method (ABG), as the name implies, involves obtaining a small sample of blood from a patient's artery and measuring the oxygen saturation level in it with the help of lab equipment. While this does present other information regarding blood pH and partial pressures of O_2 and CO_2, it only provides a measurement for the time at which the sample was taken. Contemporary transcutaneous oximetry uses electrochemical probes to estimate the amount of blood gasses that have diffused through the skin of a patient and calculate blood oxygen levels using that information. Transcutaneous measurement is continuous, but it requires some start-up time due to heating of the device and calibration procedures. Finally, pulse oximetry uses the intensity of reflected light from a patient's extremity, usually a finger, to calculate the patient's SpO_2. Pulse oximetry can provide continuous noninvasive SpO_2 readings without the need for start-up time or calibration procedures.

Due to the aforementioned benefits of pulse oximetry, it was chosen as the focus of our Major Qualifying Project (MQP). Pulse oximetry consists of two major components: a light source and a photodetector. The light source shines a specific wavelength of light into the patient's finger. The intensity of reflected light, at a particular wavelength, depends upon the oxygenation of hemoglobin in the blood. By measuring the intensity of the reflected light, we can calculate blood oxygen levels. Transmissive and reflective are the two methods for pulse oximeters. The labels in Figure 1 indicate whether the light source and the photodetector are facing each other or are on the same plane. However, it goes without saying that pulse oximetry is not a surefire method to assess patient vitals. The method is susceptible to noise artifacts, which include ambient light, motion, perfusion. For this reason, medical experts should view pulse oximetry as one of many tools when diagnosing or evaluating patients.

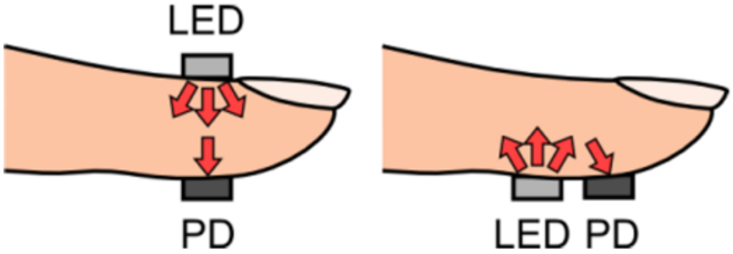


Figure 1: Transmissive (left) vs. reflective (right) pulse oximeter variations.

In order to help organize our project, and it has three main milestones. They are listed below in chronological order:

1. Functional Breadboard Design
2. Functional Stiff PCB Design
3. Functional Flexible PCB Design with Human Testing

Each milestone has a primary purpose or objective. The breadboard design helped to rapidly prototype various components and circuit designs without much commitment. The stiff PCB design further proved that the designed circuits were indeed functional and was used to debug any issues related to the PCB layout of analog and digital components. The stiff PCB also provided a final platform to perform human testing of our device. Due to unforeseen circumstance from COVID-19 pandemic, the final milestone had significant roadblocks. The time-sensitive nature of the project prevented the team from fabricating and testing the flexible PCB variant. With that being said, the team did deliver the layout of flexible PCB for future work.

There were different challenges faced throughout the design process of the wearable sensor. Therefore, each of the milestones was chosen to give the project enough flexibility to deal with unforeseen issues while, at the same time, continuously pushing towards a functioning sensor deliverable at the end of the project. In terms of software, the app design was be the most difficult, as it involved the use of various libraries and dependencies that had differing levels of functionality. Our primary expected outcome was the design of a wireless, wearable, and fully app-connected pulse oximeter.

Pulse oximeters are built around their light source and photodetector. Light-emitting diodes (LEDs) are usually used as light sources in most pulse oximeters. The intensity of reflected light must be measured for two different wavelengths in order to calculate SpO_2. Therefore, most designs use red, IR, or green LEDs as light sources. Various factors such as wavelength and skin color can significantly affect the intensity of reflected light from a patient's extremity due to this, most designs use a dedicated LED driver for each light source to adjust the intensity of the LED.

After light is shined into a patient's body, photodiodes are used to capture and convert the intensity of reflected light from the body into a small current. The current produced by the photodiode is sent into a transimpedance amplifier, which amplifies and converts it into a voltage. Depending on the expected noise and amplitude of the signal, the output voltage of the transimpedance amplifier can be fed into other filtering and amplifier stages before being sent to an analog-to-digital converter.

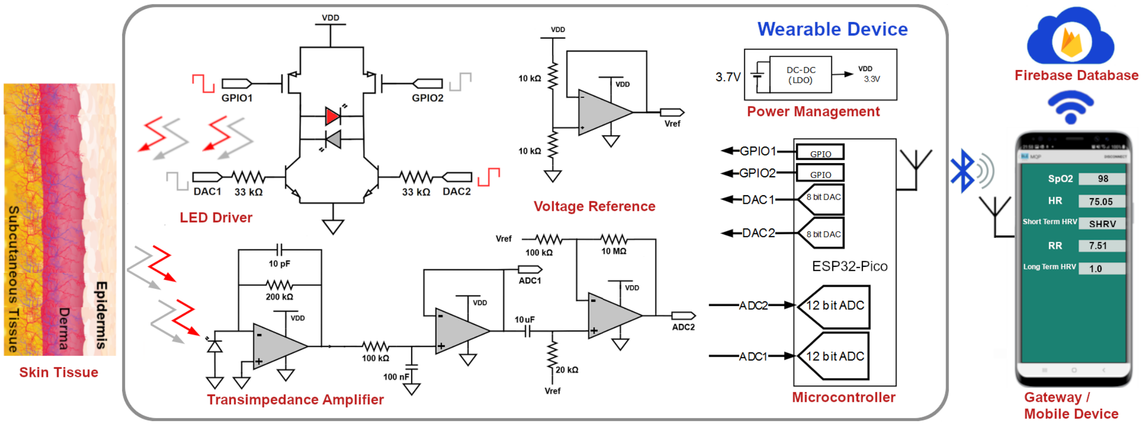


Figure 2: High-level block diagram of proposed design.

After digitization, the intensity of reflected red and IR lights can be used to calculate SpO_2 through the use of different equations and empirical models. Most medical SpO_2 measurements are displayed to the user via a fixed display. This project's design takes this a step further by incorporating a wireless transceiver to transmit the SpO_2 data to a custom phone app. The app is able to securely store all of the SpO_2 data in an online realtime database. This allows for patients to have more freedom in their daily lives while being monitored by their doctors remotely.

Human testing was done with one test subject. The subject wore the project's wearable sensor on their left hand's index finger and the commercial Masimo Rad-8 pulse oximeter on their right hand's index finger. Our device in its current state only produces one reading with every two minute reading while the commercial Masimo device continuously updates its SpO_2 and heart rate measurement. While we had access to the data produced by our sensor the Masimo's readings were recorded via video with a cell phone. The video was then used to create a weighted average of the

Masimo data by multiplying its SpO_2 measurement by the amount of seconds it was reported by the device. Then these weighted numbers were averaged by summing them and dividing by the total test time. In this way we were able to get a single reading for SpO_2 and heart rate from the Masimo device to compare to our wearable sensor's results.

Our project aimed to create a flexible wearable health sensor. We were able to develop a wearable health sensor that measured, stored to a realtime database, and displayed through a phone application the user's heart rate, respiratory rate, heart rate variability, and arterial stiffness. Despite challenges properly testing the device with complete human trails, we were able to show that our wearable sensor produces results that are comparable to the current state of the art for these measurements. Over four sets of measurements between our wearable sensor and the commercial Masimo pulse oximeter, we were able to achieve an average error of 0.4225% for SpO_2 and an average error of 2.006% for the heart rate. Furthermore, we were able to achieve the ballpark range for short-term heart rate variability albeit slightly higher indices values. Future work should be focused on implementing the design on flexible material and for a full set of human trails to be run.

List of Figures

List of Tables

1. Introduction

Health monitoring systems are important for modern medicine to diagnose and treat patients. With our device, we plan to target remote patient services (RPS). As described by University of Pittsburgh Medical Center, the benefits of RPS is as given [1]:

1. An extra layer of security
2. Reduced readmissions and costs
3. Compliance and continuity of care
4. Convenience and increased independence

When discussing the first point, in the current climate of COVID-19, remote monitoring is beneficial to reduce the amount of direct exposure of healthcare workers and to diagnose patients safely. The second point discusses the benefits of providing preventative care to patients as a mitigation effort to reduce lasting damage. RPS also allows medical practitioners to effectively monitor patient compliance to ensure the long-term welfare of the patient. The final point, which is also discussed by the American Heart Association [2], is the convenience of RPS and empowers patients to better manage their health at home rather than in a clinical setting. Our device looks into measuring patient's blood oxygen saturation as one of many vitals to evaluate patient health. This chapter explores the motivations for measuring blood oxygen levels and the current state of the art in this regard.

1.1. Motivations

The percent of oxygenated hemoglobin in a patient's blood is a direct indicator of a patient's health. A medical condition related to blood oxygen saturation is hypoxemia, which is a condition that the blood oxygen saturation is abnormally low [3]. This condition can progress to a more serious condition: hypoxia. According to the World Health Organization (WHO) defines hypoxia as the deprivation of oxygen in the tissues [4]. Further escalation of a deprived oxygen state can lead to necrosis, which refers to cell death and cause potentially irreversible damage if left untreated.

1.1.1. The importance of oxygen and the consequence from the lack thereof

The human brain "contains about 100 billion neurons" and accounts for 2% of the body mass while using 20% of the total body resting energy production [5]. As such, the brain and the nervous system require a continuous supply of oxygen to maintain healthy functionality.

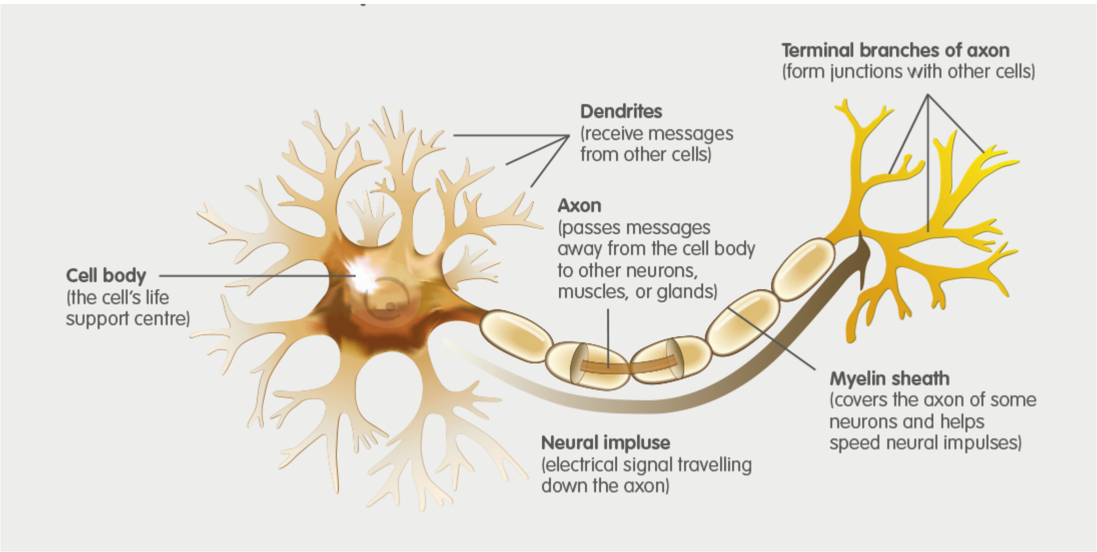


Figure 3: Neuron [6].

Without this constant supply of oxygen, neurons, like the one shown in Figure 3, cannot maintain certain functionality with the byproducts of anaerobic respiration, which will damage these cells and results in cell death and tissue necrosis, which could cause irreversible damage to the system [7]. For instance, if damage occurs in the central nervous system, the recovery of the axons involved is limited due to the low axon regeneration in the nervous system [8].

This scenario highlights one of the possibilities resulting from the lack of oxygen supply in tissues and organs. It is also applicable to muscular tissue, which, if not recognized, can result in organ dysfunction and potentially death [7]. For this reason, early detection, prevention, and intervention are vital to resolve different types of hypoxia.

1.1.2. Hypoxia

According to Myatt's journal article on SpO_2, the optimal SpO_2 range in an adult patient is 94% to 100 % [9]. Any inconsistency or irregularity, specifically with the value being below 94% [10] for adults, represents a clinical emergency. Although these are typical values, blood oxygen saturation varies depending on the type of patient. For example, in prenatal cases the recommended oxygen saturation level is 90 to 93%.

There are four different types of hypoxia: hypoxic, anemic, stagnant, and histotoxic [11]. Hypoxemic could be caused by pulmonary diseases, high-altitudes, and alteration in ventilation-perfusion coupling. Under high altitude conditions, the partial pressure of oxygen will decrease, thereby reducing the amount of O_2 going into the alveoli and, in turn, limiting the exchange process between the circulatory and the respiratory systems. Pulmonary disease, such as asthma, and chronic bronchitis decrease the amount of oxygen going into the bronchial, which reduces the arterial partial pressure of O_2. Some remedies include providing supplemental oxygen and oxygen therapy according to the patient's need.

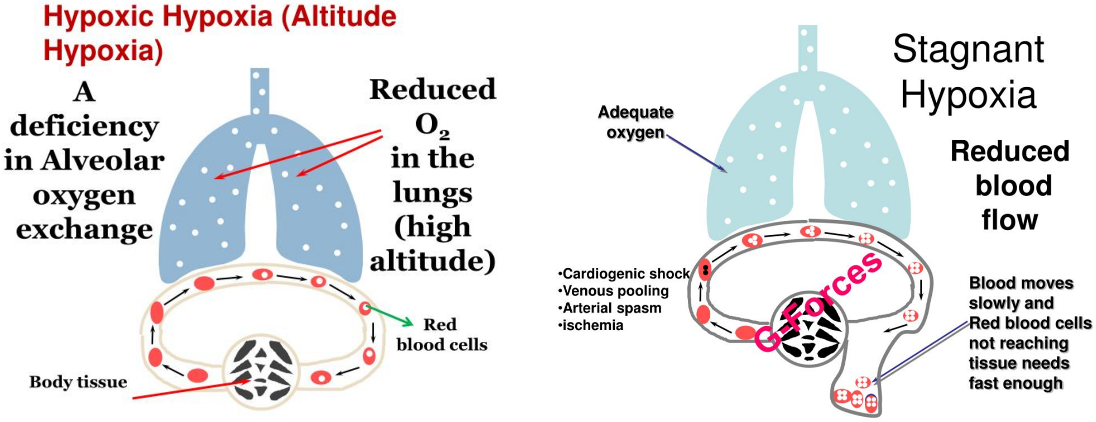

Hypoxic Hypoxia (Altitude Hypoxia)
A deficiency in Alveolar oxygen exchange
Reduced O_2 in the lungs (high altitude)
Red blood cells
Body tissue
Stagnant Hypoxia
Adequate oxygen
Reduced blood flow
•Cardiogenic shock
•Venous pooling
•Arterial spasm
•ischemia
G-Forces
Blood moves slowly and Red blood cells not reaching tissue needs fast enough

Figure 4: Altitude hypoxia [12] on the left, and stagnant hypoxia [13] on the right.

Stagnant hypoxia assumes that the partial pressure of oxygen is normal. The description of the condition mentions a decrease in the amount of blood through the capillary, which decreases oxygen delivery to the tissue resulting in the condition of hypoxia. It also refers to a situation where blood flow is low (hypoperfusion). This is a result of cardiovascular issues of two types: systemic (reduced cardiac output) and localized (ischemic). In this condition, "increasing inspired oxygen fraction does not resolve the issue". The root of the issue must be addressed, which may include the removal of a clot (ischemic) and other interventions that would improve cardiac output or peripheral blood flow.

Anemic hypoxia focuses on the oxygen-carrying capacity of the blood, where the reduced oxygen-carrying capacity triggers this type of hypoxia. The primary trigger is related to anemia, which is a disease where patients are deficient in hemoglobin in the blood. Another factor is carbon monoxide (CO) poisoning, and this is due to CO's higher affinity to bind with hemoglobin, thereby preventing oxygen from binding to the hemoglobin [11]. Therefore, the low blood cell count or abnormal

Hb will result in a low-oxygen delivery condition. These are conditions that pulse oximetry may not perform well under. Because pulse oximetry indirectly measures the partial pressure of oxygen in the blood by comparing oxygenated hemoglobin and deoxygenated hemoglobin, any issues that a patient has with their hemoglobin can create falsely positive pulse oximetry measurements. For example, a patient's blood may indicate saturation with O_2, but if CO is inhibiting O_2 from binding with Hb or the patient is anemic, then in reality no oxygen is exchanged with the targeted tissues.

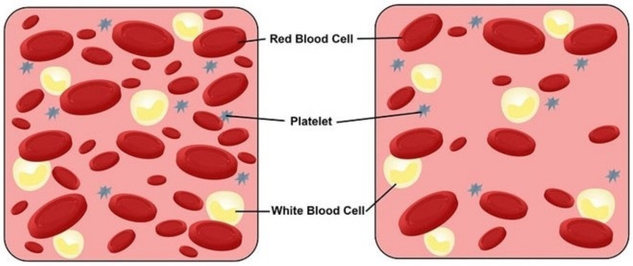


Figure 5: Anemic hypoxia, normal amount of red blood cells are presented on the left, anemic amount of red blood cells are presented on the right [6].

The final type of hypoxia, histotoxic, focuses on a cellular level, specifically, on the mitochondria, where much of energy in the form of ATP is produced to facilitate cellular processes [11]. A deficiency of O_2 prevents high energy yield and forces the cell to produce ATP through other means. These anaerobic processes produce byproducts that damage the cell's integrity. If prolonged, it could lead to cellular damage and eventual cell death.

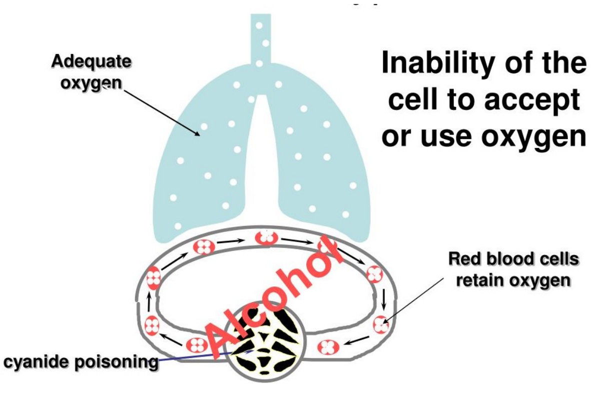

Adequate oxygen
Inability of the cell to accept or use oxygen
Alcohol
Red blood cells retain oxygen
cyanide poisoning

Figure 6: Histotoxic hypoxia [6].

As it can be seen, the symptoms and signs of hypoxia vary from patient to patient. This issue is compounded by determining the type of hypoxia afflicting the patient. Therefore, it is vital for medical personnel to correctly identify the cause of hypoxia rather than treating the onset of symptoms resulting from hypoxia. Failure to do so will prolong the oxygen-starved state, which may lead to necrosis.

1.1.3. Relevance to COVID-19

An article from the United State's CDC mentions the 2019 coronavirus is a respiratory virus that was first identified in Wuhuan, Hubei Providence, in China[14]. The analysis of the viral genome indicates similar characteristics to that of human severe acute respiratory syndrome (SARS) and the Middle East respiratory syndrome (MERS) [15]. The family of these coronavirus primarily affects the human respiratory system. Some of the symptoms related to COVID include fevers, fatigue, and particularly dyspnoea, where patients have difficulty or labored breathing [14] as indicated in the following figure. Furthermore, other serious conditions can arise from COVID-19 which includes viral pneumonia, hypoxemia, and respiratory distress syndrome (ARDs) [17]. These respiratory complications are associate with blood oxygen saturation since the virus disrupts the gas exchange process of the lungs. Furthermore, testimonials from doctors treating COVID-19 had accounts of patients with abnormally low SpO_2 despite not showing respiratory issues. Specifically, one account from Doctor

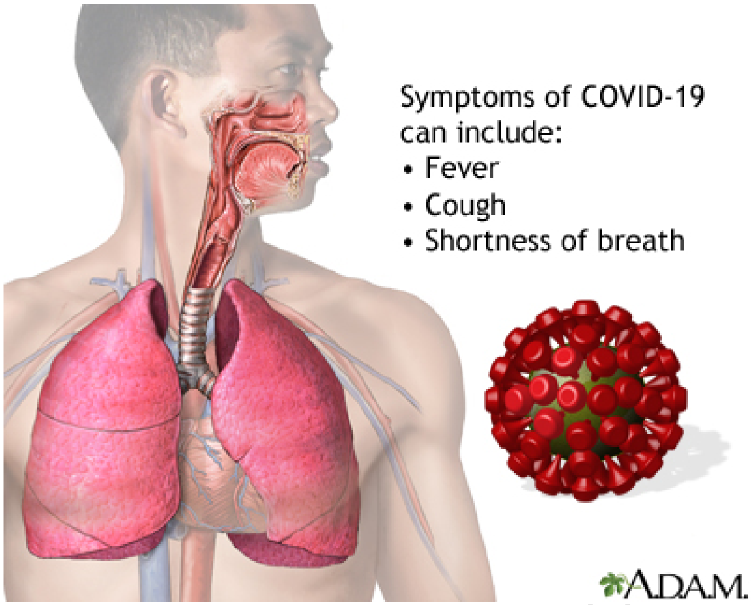


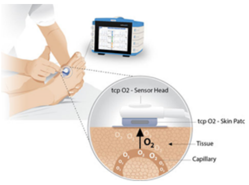
Figure 7: COVID-19 info-graphic [16].

Levitan reported seeing patients with blood oxygen saturation levels as low as 50%. These typical values are fatally low if left untreated [18]. Doctors are coining these extreme cases of extremly low blood oxygen saturation, "silent hypoxia." This is where preventative care, such as pulse oximetry, can be utilized as a method for early detection of abnormal levels of peripheral blood oxygen saturation, which will be detailed in the following sections.

1.1.4. Methods to Measure Oxygen Saturation

There are a couple of methods to monitor a patient's blood oxygen saturation in order to safeguard against the inflicted conditions above. These are illustrated in Table 1.

Table 1: Methods to measure blood oxygen saturation

Method	Pulse Oximetry SpO_2 	Arterial Blood Gas ABG After a pulse is found, a blood sample is taken from the artery ADAM.	Transcutaneous Oximetry $PtcO_2$  tcp O2 - Sensor Head tcp O2 - Skin Patc O2 Tissue Capillary
Invasiveness	Noninvasive	Invasive	Noninvasive
Procedure	A probe is attached to the patient's peripherals.	A blood sample is drawn from an artery, typically via an arterial stick from the wrist, groin, or forearm [19].	Sensors (electrodes) are applied to the body. These sensors detect and estimate the blood gases diffusing through the skin [20].
Information	SpO_2, average pulse rate, photoplethysmogram (PPG) [21].	pH, partial pressure of O_2 (PaO_2) and CO_2 ($PaCO_2$), bicarbonate, oxygen saturation (SaO_2).	$PtcO_2$ and $PtcCO_2$.
Monitoring	Continuous.	At the time of sample collection.	Continuous, requires 10-15 minute start up, and recalibration every 4-8 hours [22].

With arterial blood gas tests, the procedure requires medical personnel to draw blood samples from the patient. ABG can provide a wide set of information about a patient's condition, as listed in Table 1. For this reason, it is considered a gold standard for guiding respiratory management [23].

However, ABG has two main drawbacks. ABG is an inherently invasive procedure for obtaining vitals and a single test only provides a snapshot of the vitals at the time of sampling. Therefore, in order to determine a respiratory pattern, multiple ABG tests are needed to identify trends - these ABG tests, on average, cost around \$100-200 per test [24]. In contrast, pulse oximetry is a noninvasive alternative to measure blood oxygen saturation. Conversely, measuring SpO_2 is straightforward, and it is as simple as properly clipping the measuring site with a probe.

Furthermore, pulse oximetry can measure oxygenation continuously over a time period rather than discretely. With that being said, pulse oximetry provides a limited amount of information regarding the patient compared to ABG. Another issue arises with the oximetry probe, which is susceptible to issues such as low perfusion states, ambient light, and motion artifact.

In terms of do-it-yourself (DIY), there are at-home pulse oximeters available in the market for patients to use. However, ABG tests are limited to the hospital setting, where specially trained personnel can safely extract arterial blood. Pulse oximetry is widely adopted in intensive care and emergency care to routinely monitor patients. This is in part due to its ease of quickly measuring oxygen saturation while also measuring the average heart rate of a patient using the same oximetry probe. A new focus is looking to utilize pulse oximetry to monitor the conditions of newborns [25].

Another method to measure oxygen saturation is through transcutaneous oximetry, which utilizes electrochemical probes to extract these values noted in the table above. Like ABG, with the contemporary transcutaneos oximetry devices, the procedure requires a medical professional to attach these probes in appropriate locations. Another drawback is the startup time due to the startup of the heating element, and the need to recalibrate the sensors due to the drift observed in the sensor.

1.1.5. Heart Rate Variability

Heart rate variability is defined as the physiological phenomenon of variation in the time interval of sequential heartbeats. Measurements describing heart rate variability can extract essential information regarding a human's physiological state. For example, heart rate variability is mostly used to detect sympathetic activation, which is commonly addressed as the "fight-or-flight" response. In other words, sympathetic activation in the autonomic nervous system occurs in response to a perceived threat to survival. Furthermore, shifts in sympathetic (i.e. nervous system activates fight-or-flight response) and parasympathetic (i.e. nervous system restores body to a state of calm) activity can depict various physiological conditions or events. For example, when experiencing significant blood loss, the sympathetic nervous system is activated in order to control arterial blood pressure. Similar results in the nervous system can also be observed with airway obstruction or even swift changes in emotional states. Finally, other indications that can be provided by long-term monitoring of heart rate variability could be prolonged mental stress or different cardiovascular disorders as they tend to display particular heart rate fluctuation patterns [26].

1.2. Current State of the Art

This section covers the current technology related to pulse oximetry. Peripheral blood oxygen levels can be calculated from the PPG data collected from sensors. The following sections explain some of the principles and calculations involved in this process.

1.2.1. Pulse Oximeters

Pulse oximetry is widely utilized in the clinical setting, for instance, during anesthesia, surgery, and other patient-monitoring applications. This tool is widely adapted, with the American Society of Anesthesiologist recommending pulse oximetry as essential for monitoring oxygen saturation in 1986 [27]. With the rising cost of healthcare, this noninvasive method to measure SpO_2 seems to be considerably cheaper compared to ABG tests [28]. This is primarily due to the fact that pulse oximetry utilizes an indirect (optical) method to measure oxygen saturation. A simplified calculation for oxygen saturation through pulse oximetry is given by the following equation:

$$SpO_2 = \frac{HbO_2}{HbO_2 + Hb}.$$

Pulse oximetry only considers two functional hemoglobin: HbO_2 represents the concentration of oxygenated hemoglobin, and Hb, the concentration of deoxygenated hemoglobin. A pulse oximeters use light-emitting diodes that shine light into a tissue bed and a photocollector that collects the resulting non-absorbed or reflected light.

1.2.2. How Do Pulse Oximeters Work?

The pulse oximeter operates according to Beer-Lambert's law, which states that there is a relationship between "absorbance and concentration of an absorbing species" [29]. Utilizing Beer-Lambert's law to oxygen saturation, the intended species are oxyhemoglobin and deoxygenated hemoglobin. This clinical method uses light intensities at specific wavelengths to estimate the oxygen saturation in the blood with a photodiode. Pulse oximetry takes advantage of the differing optical absorbance for oxyhemoglobin (HbO_2) and deoxyhemoglobin (Hb). These differing characteristics allow different types of hemoglobin to absorb light differently.

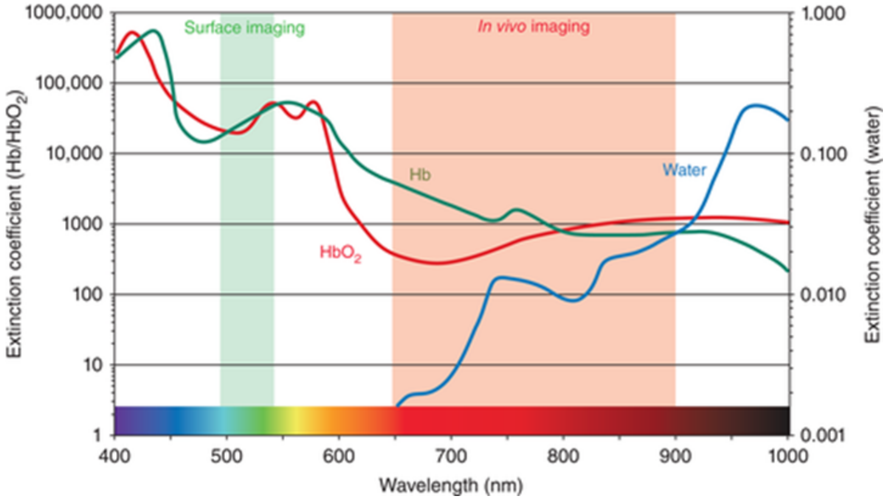


Figure 8: Extinction coefficients of Hb, HbO_2 over visible and NIR wavelengths [30].

Conventional pulse oximeters use two wavelengths of interests red (650 nm) and IR (940 nm) to calculate oxygen saturation. Based on Figure 8, at the red wavelength, there is a considerable difference between the absorption of HbO_2 and Hb. Conversely, there is a relatively small difference in absorption at the infrared spectrum. Furthermore, Figure 8 illustrates that Hb absorbs more red light compared to HbO_2, and conversely, HbO_2 absorbs IR light compared to Hb. Along with design references and Figure 8, these are the principal wavelengths used in conventional pulse oximeters. Alternatively, emerging developments in SpO_2 devices allow the devices to measure blood oxygen saturation with other wavelengths such as green and red LEDs [31]. As mentioned in Lochner's all-organic pulse oximetry solution, "the green wavelength is comparable to the difference at near-infrared wavelengths" [31]. At its core, pulse oximetry essentially compares how much a selected light is absorbed by the blood, which depends on the concentration of HbO_2 and Hb. Based on the incident light, the photodiode will convert the unabsorbed/reflected light into a current proportional to the intensity of the received light. This signal is then conditioned through user-defined filters and amplification stages to then be delivered into a microcontroller for additional processing. The pulse oximeter can extract a corresponding PPG waveform. Figure 9 illustrates the expected signal with defined characteristics of the waveform.

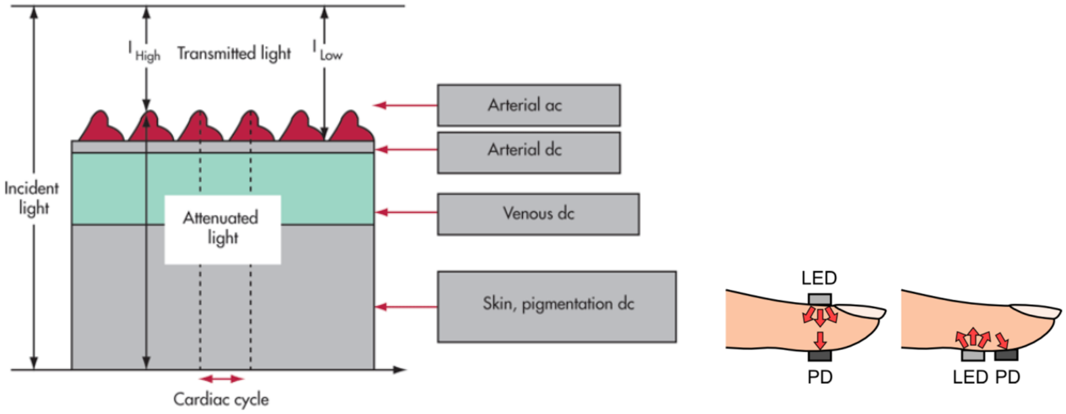


Figure 9: Tissue illustrating arterial pulsation [32] on the left, and different styles of pulse oximetry [33] on the right.

Overall, the waveform provides information regarding pulsatile blood volume change within the measured site. This pulsatile nature of the arterial blood makes up the AC component of the PPG. Alongside, is the DC component from the non-pulsatile part of the signal, which the AC signal rides on. The DC portion is made up of venous blood, skin, and tissue. When relating the PPG to the cardiac phases, systole is the point where the heart contracts and pumps blood into the arteries. This period corresponds to the time the volume of blood in the tissues increases and consequently light is attenuated according to Beer-Lambert Law concentration, which is indicated by I_{High} in Figure 9. In contrast, diastole has blood entering the heart, which decreases the amount of blood in the vascular bed, which effectively increases the amount of light collected by the photodiode represented by I_{Low}. In terms of percentages that make up the PPG signal, the pulsatile AC signal can be as low as 2%, with the remaining 98%, an unchanging DC waveform. For this reason, pulse oximetry is susceptible to error if the probe is placed incorrectly or displaced. Furthermore, specialized circuit are required to extract this small pulsatile component for signal processing. With that being said, these values, the "ratio-of-ratio" of light absorbed, change due to the pulsatile nature of arterial blood. The resulting ratio is utilized to calculate oxygen saturation.

The same PPG waveform used to calculate a patient's oxygen saturation can also be used to calculate a person's heart rate variability that was mentioned in section 1.1.5. This means that greater insight about a patient's health state can be delivered to health care providers without the need for additional sensing equipment.

1.2.3. Transmissive vs. Reflective

There are two types of optical arrangements for pulse oximetry: transmissive, and reflective, as shown in Figure 9. Traditional pulse oximeters use a transmissive design. The LEDs and the photocollector are placed at the opposite ends of a tissue bed. These LEDs shine through the tissue bed with the unabsorbed light being collected on the opposite end with the photodiode. Typically, these transmissive probes are placed on fingers, earlobe, and toes, which are limited to the extremities. In contrast, the reflective arrangement has the photodiode and LEDs on the same plane with each other, as illustrated on the rightmost finger of Figure 9. This configuration offers flexibility for the potential placement of anywhere on the patient rather than being limited to the patient's extremities [33].

2. Project Overview

2.1. Design

Our proposed design makes SpO_2 monitoring more versatile. Our final prototype was made on a stiff PCB while trying to make the form-factor as small as possible. Our design also incorporates the use of a phone application and realtime database to process and store the PPG data. This allows us to extract other useful health signals from the PPG signal apart from just SPO_2. The app also allows the user to easily check their vital signs and the realtime database could function as a resource for future health care providers.

In terms of circuitry, our design will use one red LED and one infra-red LED in a reflective configuration to measure the oxygenated and the deoxygenated hemoglobin in the blood. The LEDs will be periodically pulsed on and off. Some of the light produced by the LEDs will be absorbed in the body, and the rest will be reflected. Our device will capture some of this reflected light using a photodiode. The photodiode will have a responsivity spectrum wide enough to create current from both reflected red and IR wavelengths. This current will then be fed into a transimpedance amplifier that will turn the current into a voltage. This signal voltage will be fed into an on-chip analog-to-digital converter (ADC) inside our microcontroller. Once the PPG waveform is sampled with the ADC, the microcontroller stores received waveforms in its onboard memory before sending over the data to a mobile device via BLE connection. At this point, the algorithms implemented within the mobile application perform the necessary calculations to extract the health measurements. Alongside this, the data is also pushed onto a realtime database to allow users to track vital patterns.

2.2. Project Organization

This report begins by addressing our motivations for measuring blood oxygen levels within a patient. The relevance of these measurements with regard to different types of hypoxia, and COVID-19 are also covered. Chapter 1 also reviews the current state of the art for measuring blood oxygen levels and the benefits/costs of different possible measurement techniques. Chapter 2 outlines the proposed design for this project's pulse oximeter, as well as the project's organization and possible challenges.

Chapter 3 contains the background research we conducted on each of the blocks in our system-level diagram. Equations for expected voltages, currents, and gains in our circuit are explained here in detail. This chapter explores the constraints associated with each block of our design and ultimately defines the criteria for the various value analyses that were conducted while choosing parts.

Chapter 4 covers the proposed design of the system, which consolidates all of the design considerations in Chapter 3 from abstraction to a defined configuration of the system. This section includes various simulations to ensure proper functionality before implementation. Chapter 4 also covers the various algorithms we implemented to extract relevant measurements from the PPG data.

Chapter 5 lists the various tests for the physical design of the system alongside experimental results. Chapter 6 tests the integration of the analog circuitry with the microcontroller and the phone application measurement-extraction algorithms. For the health measurements of SpO_2 and heart , we compare the results of our pulse oximeter device to commercially available pulse oximeters. Respiratory rate and heart rate variability are also verified to be within acceptable ranges of values based on results from published papers in the field.

Chapter 7 and 8 discuss the potential recommendations for future work regarding this project and our conclusions for the project as a whole.

2.3. Project Challenges

We expect to face some challenges during our project. First of all, the design of a pulse oximeter device that is competitive to the state of the art will be difficult due to the plethora of existing work related to SpO_2 measurement. We hope to address this issue by extracting other relevant measurements from the PPG signal. From our background research we have not seen other devices incorporate SPO_2, HR, RR, HRV, and Arterial stiffness measurements all on the same device. Second, the unforeseen shutdown of WPI due to the COVID-19 pandemic during the last term of our MQP produced problems related to accessing our prototype device from the NECAMSID lab. Furthermore our plans for human trials had to be changed. In terms of accessing the lab we were able to retrieve our prototype and testing equipment with the help of our advisor. While we were able to get our IRB form approved before WPI shutdown, instead of implementing human trials we were only able to test our device on one of our team members.

3. Background Research

The background chapter will focus on the constraints and necessary considerations related to each block in the block diagram. Our target end-user and constraints will ultimately guide our design choices. Below is the high-level block diagram of our proposed design.

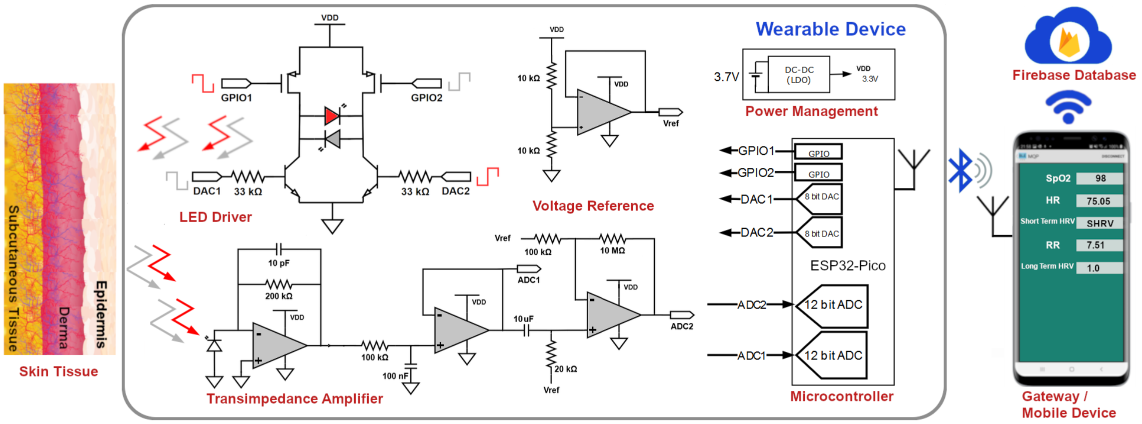


Figure 10: High-level block diagram of the proposed design.

3.1. Photodiode and Light-Emitting Diodes

Noninvasive pulse oximetry requires the light emitted from a light-emitting diode (LED) to enter the patient's body and reflect onto a photo-detector. The amount of oxygen in the patient's blood and in turn the intensity of the incident light determines the amount of light detected. This section will discuss how LEDs and photodiodes work as electrical emitters and detectors of light. Equations to calculate the intensity of light from the surface of the patients' skin and the intensity of expected reflected light at the photo-detector will be presented.

3.1.1. LED Function

The research and manufacturing processes concerning LEDs is vast and diverse. For our proposed design, we need to understand the physics behind the production of light by an LED. Photons are produced by electrons dropping to a lower quantum energy level and recombining with a hole [34]. The energy of the photon will be equal to the energy difference with the levels that the electron dropped.

To design our pulse oximeter, there are a few key specifications that are important to understand. We can assume that for our design the operating voltage of the LEDs will be below our desired supply voltage. The operating current of the LED will depend on the intensity of the light we would like to produce. The LED current will also dictate how much the circuit will draw power and accordingly how long the system will be able to run on a given battery. The most important specification for our device is the output light intensity of the LED. This light intensity will dictate how much light is available to be reflected onto the photodiode and, in turn, tell us how much current we can expect from the photodiode. The equations to calculate current in the photodiode will be discussed in the following sections.

The intensity of any light source is measured as power over a given area. Typically, this is denoted as lumens per square centimeter. It is important to note that intensity can also be referred to as irradiance. Our paper will use the term intensity. Most datasheets for LEDs do not directly express a number for intensity. Instead, they note the base unit of light in candela or millicandela. The unit of candela takes the total lumens a light source produces and divides it by 4π steradians. A steradian is analogous to the radian of a unit circle. Steradians are the unit of measure for the surface area of spheres. Similar to how a unit circle has 2π radians as its circumference, a unit sphere has 4π steradians of surface area.

This is useful for anyone who is using a light source rated with a specific number of candela, to calculate the expected light intensity over an area a chosen distance away from the light source.

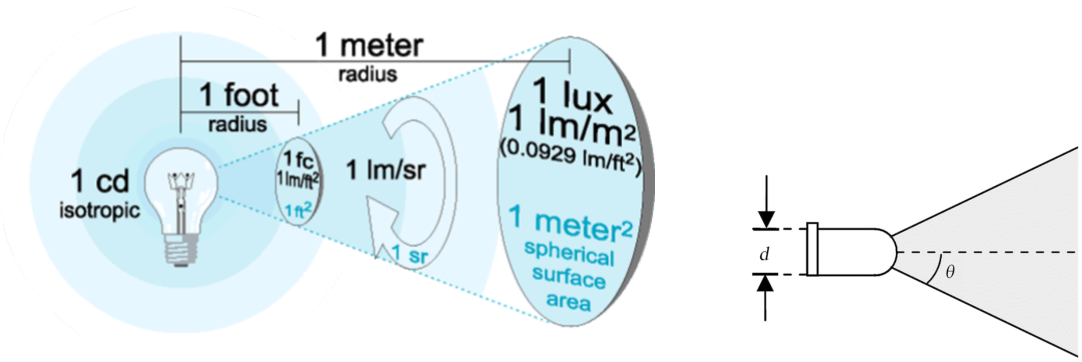


Figure 11: Irradiance relationship to candela [35] on the left, Viewing Angle of an LED [36] on the right.

The image above offers an intuitive way of looking at the way light intensity (irradiance) changes as a function of distance from the source. Another figure of importance that is related to LEDs is their viewing angle. LEDs typically emit light in a specific cone, as shown in Figure 11(right), that begins inside the LED and expands outwards away from the LED. The viewing angle represents this cone of light and is the angle at which the light intensity output by the LED is half of the light intensity output at 0 degrees. Figure 11 shows an LED with a 2D representation of this viewing angle and cone from which the majority of the LEDs light intensity is being emitted.

The viewing angle is important because two LEDs can emit the same number of candela but one may emit this light in a more focused manner. Using the equation below we can calculate the steradians produced by an LEDs viewing angle

$$\Omega = 2\pi(1 - cos(\frac{\theta}{2})). \tag{1}$$

In this equation θ is the viewing angle of the LED. Using this equation we can calculate the intensity of light an LED produces at a distance r. For example, an LED with a viewing angle of 30^o has .214 steradians. If we consider that the LED produces 1cd then the LED will produce .214 lumens in the cone produced by its viewing angle.

3.1.2. Photodiode Function

Photodiodes have energy bands that are gradually more offset throughout the structure of the material since they are PN junctions [37]. When photons hit electrons in low energy bands, the electrons absorb the photon's energy and move up energy level bands. This movement towards higher energy bands creates a small current that can be amplified. The following figure is a simple diagram of the movement of electron and holes within a photodiode due to photon excitation.

We can approximate the amount of current dependent on the amount of incident light on the photodiode. The equation below shows this relationship [38]

$$I_{\text{PHOTO}} = \alpha q \frac{P_{\text{L}} A}{\frac{hc}{\lambda}}. \tag{2}$$

The values of q (the electron charge), h (Planck's constant), and c (speed of light) are all constants of nature. The α variable is the quantum efficiency of the photodiode and is the number of electrons produced per photon. However, most data sheets report a responsivity value labeled R and has units of A/W. Responsivity for a photodiode is normally plotted against different wavelengths of light on the x-axis. It captures the efficiency of a photodiode to convert light of a specific wavelength to a photocurrent. Moreover, The relationship to this responsivity with quantum efficiency is given by the equation below [39]:

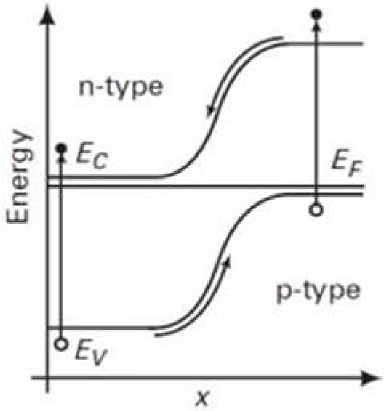


Figure 12: Simplified energy band structure of a photodiode

$$R = \alpha q \frac{1}{(\frac{hc}{\lambda})}, \tag{3}$$

which simplifies Equation 2 into the form

$$I_{\mathrm{PHOTO}} = R \times P_{\mathrm{L}} \times A. \tag{4}$$

The remaining variables $\mathrm{P_L}$ and A relate to the light intensity on the sensor and the active area of the sensor. Therefore, the current we can expect out of the photodiode we choose can be calculated by multiplying the responsivity with the intensity of light that reached the active area of the photodiode. This equation will help us estimate the expected current out of the photodiode and guide the design choices for this project.

3.2. Transimpedance Amplifiers

Due to the low-level output current generated by the photodiode, an amplifier is needed to convert the generated current (photocurrent) to a sizeable voltage signal for the microcontroller in order to interpret varying current signal. This conversion requires a current-to-voltage converter, commonly referred to as a transimpedance amplifier, as shown in Figure 13.

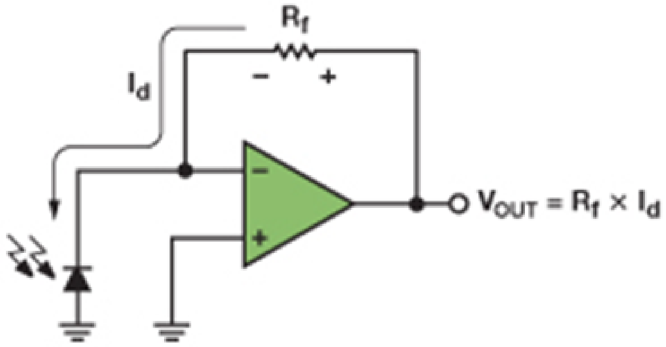


Figure 13: Transimpedance amplifier [40].

3.2.1. Transimpedance Operation

In Figure 13, the photodiode is configured in photovoltaic mode and illustrates that a zero bias voltage is across the photodiode due to the virtual ground at the inverting terminal from the grounded non-inverting terminal of the operational amplifier (op-amp). The photocurrent $\mathrm{I_d}$ flows through the feedback resistor $\mathrm{R_f}$ that generates a corresponding output voltage in the following equation

$$V_{\mathrm{out}} = -R_{\mathrm{f}} \times I_{\mathrm{d}}. \tag{5}$$

This relationship demonstrates that the output voltage is proportional to the generated photocurrent (light intensity), with the gain of the amplifier given by the feedback resistor $\mathrm{R_f}$.

3.2.2. Transimpedance Design Considerations

A non-ideal op-amp has issues with respect to the input capacitance at the op-amp's inverting terminal. This input capacitance is given by the diode junction capacitance and the input capacitance of the amplifier. The capacitance introduces stability and bandwidth issues into the circuit, resulting in unwanted oscillations in the output voltage. A feedback capacitor, $\mathrm{C_f}$, can be placed parallel to the feedback resistor to compensate for the effects of the input capacitance [41]. This prevents the output from oscillating at the expense of the bandwidth of the circuit. The following equation can provide an optimal feedback capacitance [41]

$$C_{\mathrm{f}} = \sqrt{\frac{C_{\mathrm{I}}}{2\pi R_{\mathrm{f}} f_{\mathrm{u}}}}, \tag{6}$$

where the value of the feedback capacitor considers the input capacitance from the photodiode and the op-amp ($\mathrm{C_I}$), feedback resistor ($\mathrm{R_f}$), and the unity-gain frequency of the op-amp ($\mathrm{f_u}$). The thermal noise produced by the feedback resistor is another circuit non-ideality to consider. The thermal noise can be modeled by the following equation

$$V_{\mathrm{Noise}} = \sqrt{4kTBR}, \tag{7}$$

where the variables T and B are the temperatures in Kelvin and bandwidth of the system. The value of k is given by Boltzmann's constant. There is an essential trade-off consideration involving the feedback resistor. As mentioned in Webster on pulse oximetry design [41], the feedback resistance should be as large as possible while contradicting the thermal noise, which suggests that the signal-to-noise ratio improves according to the square root of the feedback resistance, furthermore, it is desired to have the resistor noise dominate the noise generated by the amplifier. Where an article discussing amplifier noise mentions that "noise generated by the amplifier itself is small compared to noise from the incoming source" to have optimal noise performance [42]. Additionally, for low-noise applications with source resistance would prefer CMOS amplifier due to the negligible current noise [43].

The last design consideration examines the type of op-amp to utilize in this TIA. Analog Device's design tool recommends a rail-to-rail amplifier. This is essential in single supply circuits to maximize the range of the power supply [44]. Additionally, to maximize the sensitivity of the amplifier, design references recommend low bias current op-amps, such as field-effect transistor (FET) op-amps, in order to reduce the voltage offset generated with the feedback resistor.

3.3. Filtering

The next consideration is filtering the desired signal from a range of frequencies. This requires a circuit to selectively attenuate a range of frequencies for operation and pass the desired frequencies through this signal conditioning. Figure 14 shows examples of typical frequency responses.

Ideally, filters should have a brick-wall characteristic, which passes all frequencies within the passband and rejects any frequencies in the stopband. However, Figure 14 illustrates realistic filter responses. For instance, low and high pass filters (Figures

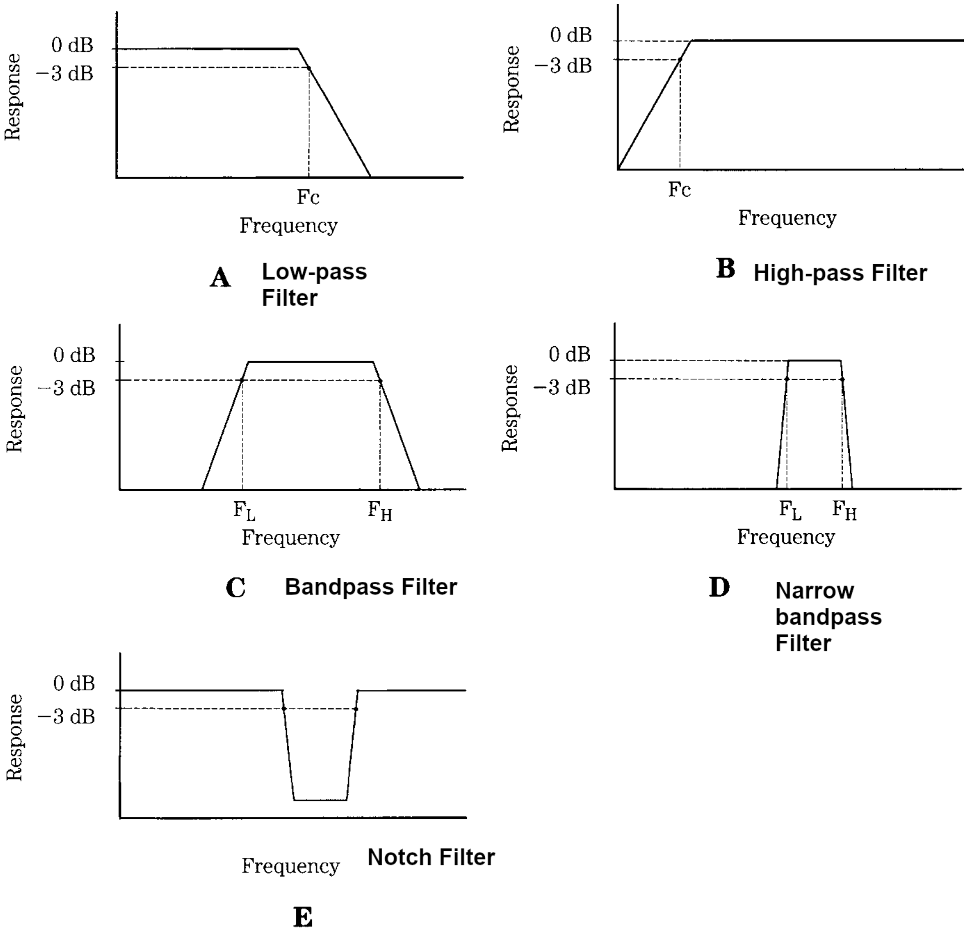


Figure 14: Filter frequency response characteristics [45].

14A and 14B) at the 3 dB attenuation frequency will continue to roll-off depending on the amount of poles or the order of the filter. For first-order filters, this roll-off is given by -20 dB per decade.

3.3.1. Analog Filtering

These analog filters are made up of electrical components such as resistors, capacitors, inductors, and op-amps (to achieve active filters). Due to the variety of components, there are many design considerations with analog filters. Additionally, there are various filter topologies ranging from Butterworth to Chebyshev, with differing frequency responses. These include differences in ripples at the passband and stopband and the roll-off slope.

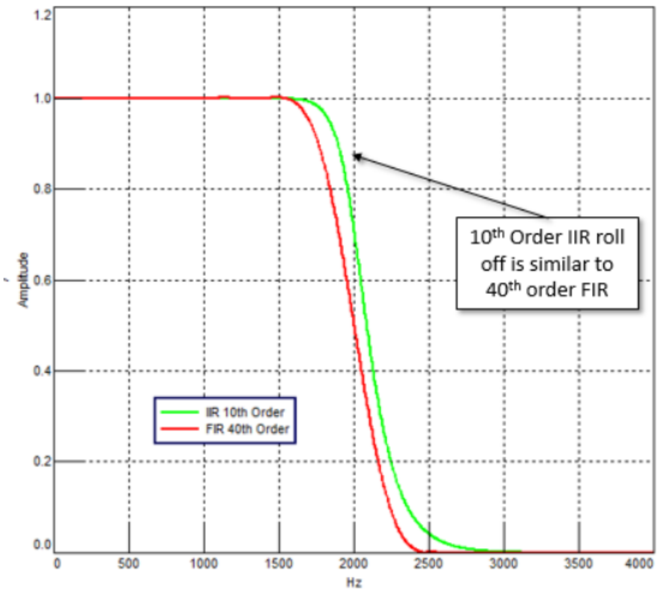
Designers can create a higher-order filter to provide a steeper attenuation slope at the cutoff frequency. However, the drawback of increasing the order is that the design complexity also increases. Increasing design complexity tends to increase component counts which may bring up issues regarding board space. When considering adaptability, analog filters are limited by the designer's initial specifications. In the case where the designer needs to adjust a cutoff frequency after a design has been implemented, the designer would have to reevaluate the component values used in the circuit to achieve the new desired frequency. Then non-idealities such as component tolerances, environmental influences, and thermal noise would have to also be reevaluated.

3.3.2. Digital Filtering

Conversely, digital filters have identical functions to their analog counterparts. A primary difference is that digital filters work with discrete signals sampled by the ADC. While digital filters are relatively inexpensive, they may introduce noise when converting a continuous signal to the digital format for digital signal processing (DSP) [46]. The two major types of digital filters are finite impulse response (FIR) and infinite impulse response (IIR). FIR utilizes the current sample and the previous input to achieve the current output. Conversely, the IIR filters require a feedback loop that depends on the output and previous inputs. Furthermore, an IIR low order filter will be computationally faster than FIR, which would require a higher order FIR to achieve the same result of a low order IIR (Figure 15). However, when considering time delay, IIR will introduce a delay in the filtered signal, whereas the FIR will be constant. With that being said, FIR filters are generally more stable than IIR filters, which may experience non-linear phase and stability issues [47].

One of the key advantages of digital filters is that, because they are implemented using transfer functions in code, they can be easy modified through programming. Therefore, it is easier to change the frequency response of the filter by adjusting the coefficients of its transfer function. In terms of size, digital filters require little to no additional components, apart from a microprocessor. Unlike analog filters, digital filters are capable of filtering low frequencies [48].

In contrast, a significant advantage with analog filters is that there are no aliasing issues compared to a digital filter, which is experienced due to the ADC. Another benefit is that they are not limited by the microprocessor's computational power. Finally, an analog filter typically removes the unwanted signal before it is digitized, as shown in Figure 16. This lowers the ADC requirement, whereas only using digital filters, both the desired and undesired frequency after ADC has digitized the signal



10th Order IIR roll off is similar to 40th order FIR

Figure 15: Digital frequency response FIR vs. IIR [47].

[48]. Ultimately, a mixture of both analog and digital filters might be the optimal approach to achieve the benefits of both types of filters.

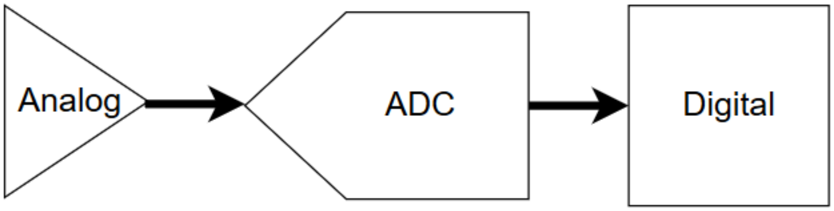


Figure 16: Generic signal chain.

3.4. Microcontrollers

Microcontrollers are a fundamental module of an embedded system, since they are designed to control the actions and features of that system. The microcontroller's role in this project is to control the LEDs, collect the PPG waveform data and transmit these data over to our smartphone application. This subsection will cover the design options and value analysis for the microcontroller of the device. The team's goals regarding the microcontroller were to choose one with a very small size but with satisfactory memory and processing power for our project. Furthermore, since it was required to transmit an array of PPG values, the team investigated the possibility of investing in a microcontroller with wireless communications capabilities. Our team's design options, value analysis and complete functionality with regards to this project's microcontroller is shown in Section 4.6.

3.5. Wireless Connectivity

Wireless connectivity is an important consideration in this design since one of our goals is to display the data gathered by the pulse oximeter on a smartphone app. The transmission of these data to the user's phone will be done through a wireless communication protocol. Such a protocol is a system of rules which confers on two or more devices the potential to transmit information to each other. More specifically, it defines the rules, syntax, semantics, synchronization of communication, and error recovery methods of the communication among devices. A wireless communication protocol uses the air as its medium, instead of a wire or cable, and information is transmitted through electromagnetic, radio, and microwave signals. We considered multiple wireless communication protocols for our application, namely Wi-Fi, Bluetooth BR/EDR (i.e. Basic Rate/Enhanced Data Rate), Bluetooth Low Energy and ZigBee. In this chapter, we delve into each protocol and detect the applications it is the best fit for, while in the next chapter, we compare them and determine, which one we consider to be the best suitor for our project.

Wi-Fi is a technology based on the IEEE 802.11 suite of standards that uses radio frequencies (RF) to expand Ethernet-based local area networks (LAN) to devices enabled by Wi-Fi, thereby giving devices the ability to transmit and receive information from the Internet. Wi-Fi uses the Internet protocol to establish communication between the LAN and endpoint devices. A connection through Wi-Fi is made possible with the help of a wireless router that is connected to the network and allows the devices of this network to access the Internet.

One of the disadvantages of Wi-Fi is its susceptibility to interference, depending on the RF environment it operates in. Different signals in the environment have the capability of interfering with other data transmissions. However, as a solution, Wi-Fi has the potential to broadcast on two frequencies (i.e. 2.4 and 5 GHz), and thereby assists the signal in cutting through the noise and delivering a faster, stronger signal from the router to an endpoint device. Applications that Wi-Fi is the best option for are LAN videos, e-mails, and web applications that need high data rate network connections (i.e. 1Mbps to 1Gbps).

Bluetooth is a standardized protocol for transmitting and receiving data via 2.45 GHz link. It is best suited for short-range, low-power, low-cost wireless interconnection of mobile phones, computers and other electronic devices. Bluetooth transmits and receives radio waves in a band of 79 different channels (i.e. frequencies) that are reserved by industrial, scientific, and medical devices (ISM) and are set apart from radio, television, and cellphones. Furthermore, Bluetooth is more secure than wireless networks that operate over longer ranges, such as Wi-Fi. The applications that Bluetooth is best for include device-to-device file transfers, mobile credentials, and enabling wireless speakers or headsets.

Bluetooth Low Energy (BLE) is another wireless personal area network technology that is ideal for low-power, low- cost IoT applications. Like Bluetooth, BLE operates in the 2.4 GHz ISM band. It utilizes 40 RF channel, where each channel is 2 MHz wide. However, it has a few significant differences with Bluetooth. Bluetooth Low Energy has much lower power consumption and therefore enables wearable applications to run relatively longer with small battery capacity. BLE also remains constantly in sleep mode unless connection is initiated. Finally, since the data rate is high (i.e. 1 Mbps), the time it takes to establish a BLE connection is only a few milliseconds.

ZigBee is a 2.4 GHz mesh local area network (LAN) that was developed as a specification of IEEE 802.15.4 for suite of high-level communication protocols used for making personal area networks (PAN) with small low-power digital radios. ZigBee devices can send information to devices over long distances by passing data through a mesh network of intermediate devices. ZigBee networks have a defined rate of 250 kbps and are secured by 128-bit encryption keys. Zigbee is mostly used in low-data-rate applications that demand high scalability, secure networking and long battery life. It is considered simpler and less expensive than Bluetooth and Wi-Fi and is found mostly in home, building, or industrial automation applications. Some examples of these applications are controlled lighting, thermostats, home energy monitors, smart metering, and traffic management systems.

3.6. Analog-to-Digital Converters (ADC)

ADCs are a crucial component of our design, as they enable us to digitize an analog voltage into bits that we can perform algorithms on. The most typical types of ADCs are flash, successive approximation (SAR), and sigma delta [49]. One of the main trade-offs for ADCs is in terms of their speed and accuracy. The speed of an ADC is called the conversion time or sampling rate. The signal of interest for us is the PPG signal, which encompasses a frequency range from 0 Hz to 15 Hz. Therefore, speed is not a limiting factor when choosing specifications for an ADC. However, quantization error, an issue related to the resolution of the ADC, directly affects the amount of error in ADC measurements. The quantization error (V_{QE}) of an ADC can be calculated using the equation below

$$V_{\mathrm{QE}} = \frac{V_{\mathrm{D}}}{2^n - 1}, \tag{8}$$

where n is the numbers of bits and V_D represents the full-scale input voltage range. There are other types of errors that can be introduced to an ADC measurement. These include gain error, linearity error, offset error, and a missing code error. The figure below illustrate these types of errors with respect to the output of an ideal ADC.

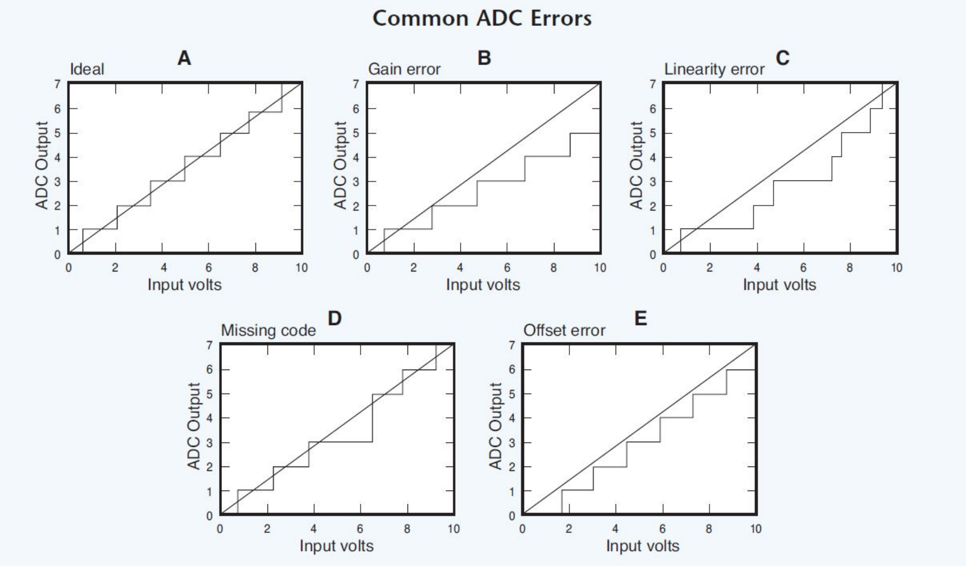


Figure 17: Common ADC errors [50].

Gain and offset error can sometimes be mitigated by changing the gain and DC level of the amplifier driving the input of the ADC. This is an issue that can be anticipated while designing but must ultimately be "tuned" with a built system. Typically, the missing code and linearity errors are either calibrated at the factory or are noted on the datasheet and must be dealt with by the user in some way.

3.7. LED Driver

The maximum current that a microcontroller general input or output pin (GPIO) can source or sink presents a limitation for the types of loads that the controller can provide power for. The role of LED drivers is to provide adequate power to LEDs while being controlled by a microcontroller. Some LED drivers circuits can be as simple as to merely have ON and OFF states. Alternatively, other designs allow the user to change the amount of power being delivered to an LED. For our application, we need to be able to control both the ON and OFF state of the LED and the power being delivered. Thus, the MCU can utilize a digital-to-analog converter (DAC) to control the current flowing through the intended LED [51]. The following figure illustrates two circuit toplogies one from a Texas Instrument reference design that uses the AFE4403 analog front end for pulse oximeters [32] and another from LEDnique [52].

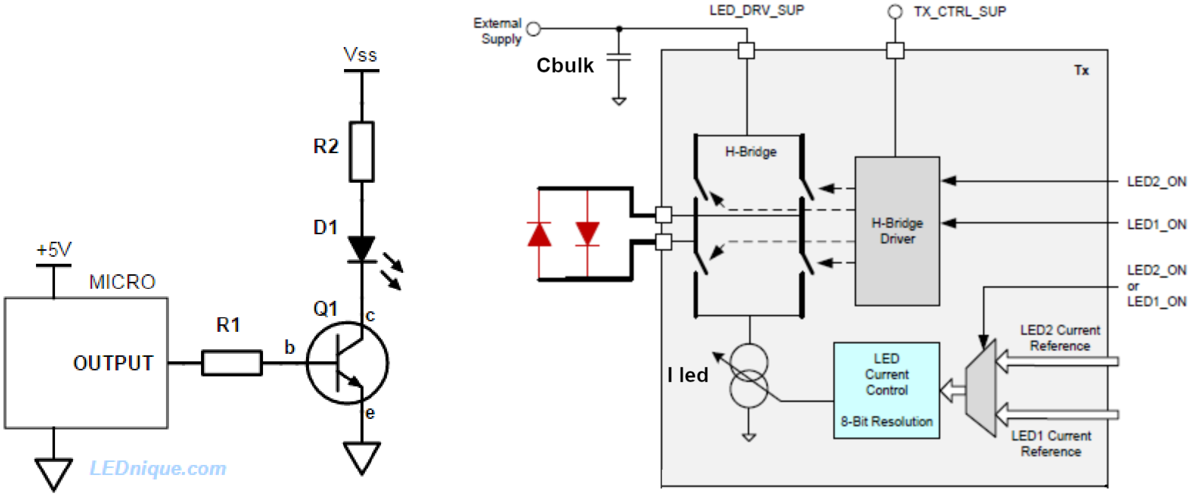


Figure 18: NPN transistor driver [52] on the left, and H-bridge driver [32] on the right.

As stated previously, the circuit illustrated in Figure 18 uses a NPN transistor to either work as a driver or a switch to power the LED. This also serves as a buffer between a low-voltage control circuit, as in the case with the microcontroller and a high-voltage load circuit (the LED load).

Alternatively, an H-bridge configuration can also be used to drive LEDs. While comparing the two, the H-bridge is a rather complex circuit compared to NPN transistor drive circuit in terms of control signal timing, but offers flexibility to switch LEDs off and adjust the driving current through the LEDs. With this in mind, an important consideration is choosing a topology according to the package of LEDs.

For instance, the H-bridge configuration requires that the LEDs are anti-parallel. This refers to the components in parallel, but opposite in polarity.

3.8. Power Management: Low Drop-Out Voltage Regulator

There are two linear regulators of interests for this project: standard linear regulators and low dropout linear regulators. The primary difference between the two is the dropout voltage, which is the minimum amount of voltage required across a regulator to output its nominal voltage regulation. Figure 19 shows the working operation of a voltage regulator. In the figure, a 5 V voltage regulator with a 1 V dropout voltage requires a minimum 6 V at the input for proper functionality.

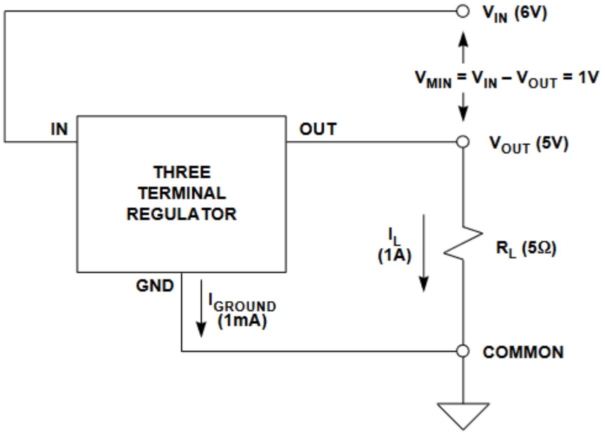


Figure 19: A typical three terminal voltage regulator [53].

The generalized equation for the dropout voltage is given by

$$V_{\text{dropout}} = V_{\text{in}} - V_{\text{out}}. \tag{9}$$

In terms of power consumption, Equation 10 presents how the dropout voltage is related to the overall power consumption of the regulator

$$P_{\text{D}} = (V_{\text{in}} - V_{\text{out}})(I_{\text{L}}) + V_{\text{in}}I_{\text{ground}}, \tag{10}$$

where I_L is the load current, and I_{ground} is the amount of current dissipated through the third terminal while operating. Intuitively, from Equation 9, a low drop-out voltage conserves more power and is therefore more power efficient compared to a typical linear voltage regulator. This is due to the fact that load current and dropout voltage have more influence on power dissipation (larger in magnitude) compared to the rest of the terms in Equation 10. Some of the important specification related to such regulators include low quiescent current, dropout voltage, and power supply rejection ratio (PSRR) [54]. Low quiescent current is the difference in current that goes into the input compared to the current that goes to the output[55]. This current is used by the LDO to regulate the output power and is not delivered to the load. Therefore, for low power applications the quiescent current should be reduced as much as possible to improve device functionality using a battery. The dropout voltage is the difference between the input voltage and the output voltage. This quantity is crucial because as mentioned before it dominates the power consumption of a low-dropout regulator. Finally, the PSRR is the ability of the LDO to prevent voltage ripple on its input from affecting the output voltage. The PSRR is typically plotted against frequency. These features of the LDOs described above are especially attractive for battery-powered electronic devices. These families of regulators maintain regulation even with small differences between supply voltage and load voltage [56], which leads to a proper and efficient power management system to reduce the overall power consumption.

4. Proposed Design and Implementation

This chapter explains the intended proposed design for our pulse oximeter system. This includes the various value analysis to select the components of the design. Furthermore, the chapter also includes various simulations generated from LTspice.

4.1. Proposed Design

Our proposed design will make SpO_2 monitoring more versatile. Our final prototype will be made as small as possible, and we will use a flexible PCB. A compact design with a flexible PCB allows the device to easily accommodate different body parts and body shapes. This approach will remove the necessity to attach a SpO_2 monitor to only a patient's extremities.

After conducting the initial value analysis for amplifiers in Appendix A, the first family of amplifiers we will be using is the ADA4500-1/2. These amplifiers have low input bias currents and functions rail to rail. Both specifications are important to reduce noise and improve the output swing of our signal. However, in the final iteration of the system, the appropriate amplifier would be LTC2067, in order to reduce the overall consumption of the system.

Another value analysis determined the potential photodiodes and LEDs appropriate for the system. Based on the value analysis in Appendix C, the desired photodiodes are SFH2201 and SFH7013. Although SFH 2201 (photodiode) and SFH 7013 (LEDs) scored highest in the review, special consideration was made to choose the EPM-4001 (photodiode) and ELM-4002 (Red and IR LEDs). Although they scored relatively low on the analysis, the primary reason behind their choice is the fact that these products work in tandem with each other, specifically in SpO_2 solutions. Another consideration is to comply with the manufactures' guidelines. Specifically, many LED components and photodiodes had a disclaimer to receive authorization in critical applications such as life support systems.

The third set of value analysis looked at the optimal microcontroller choice for our system. Based on the value analysis in Appendix B, which compared its features to those other families of microcontrollers, ESP32 was the chosen microcontroller for the system. After digitizing the signal, the microcontroller will adjust the DAC voltage accordingly in order to control the LED driver. This voltage fed into the BJTs controls the current flowing through the LEDs. Ultimately, the current of the LEDs is proportional to the light intensity emitted. Controlling the light intensity allows our system to control the overall signal that the photodiode will be outputting. This provides flexibility to calibrate the intensity of the light going into the user based on the output signal swing.

Along with calibration, the ESP32 will store ADC readings and transmit data over to a BLE-connected device. Further details of the microcontroller program details making can be found in Appendix G.1. The following sections provide additional details of each subblock.

4.2. Infrared and Red LED Driver

The LED driver will control the two different wavelengths necessary for the pulse oximetry application. As mentioned previously, the ESP32 pins does not provide enough driving current to power the LEDs. Therefore, a specialized LED circuit is required to control the LEDs. The chosen circuit topology for the LED driver takes upon TI's H-bridge LED driver due to the selected LED package [32]. In this configuration, the PMOS operates as a switch to turn on the corresponding LEDs, while the alternating npn BJTs controls the light intensity the LEDs as a technique to control the DC level. Additionally, the ESP32 microcontroller provided the necessary LED logic to control the LED driver circuit. This included switching the corresponding PMOS on (logic low) and off (logic high). In terms of timing, the ESP32 was programmed at 10 kHz at 50% duty cycle. This meant that the LED was on 50% of the time. In the case with 10 kHz switching frequency, it was off for 50 µs, 25 µs for one LED and another 25 µs for the other LED.

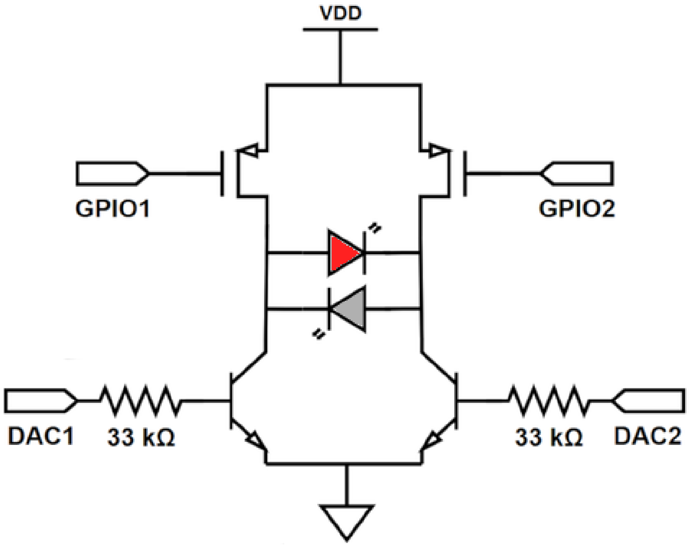


Figure 20: LED driver circuit.

4.3. Analog Front End

This subsection discusses the analog signal conditioning of the design. The overall analog front end consists of the transimpedance amplifier, first-order low pass filter along with a first-order high pass filters, non-inverting amplifier, and a reference circuit, which is illustrated in the following figure. From Figure 21, a transimpedance amplifier with a gain of 200 kΩ was utilized to achieve gain for both AC and DC components of the photocurrent. It was important to realize that a significant gain at this stage could amplify the signal such that the DC component of the PPG signal rails to the supply voltage, thereby saturating the AC pulsatile signal. The output of the TIA was fed to a set of filters.

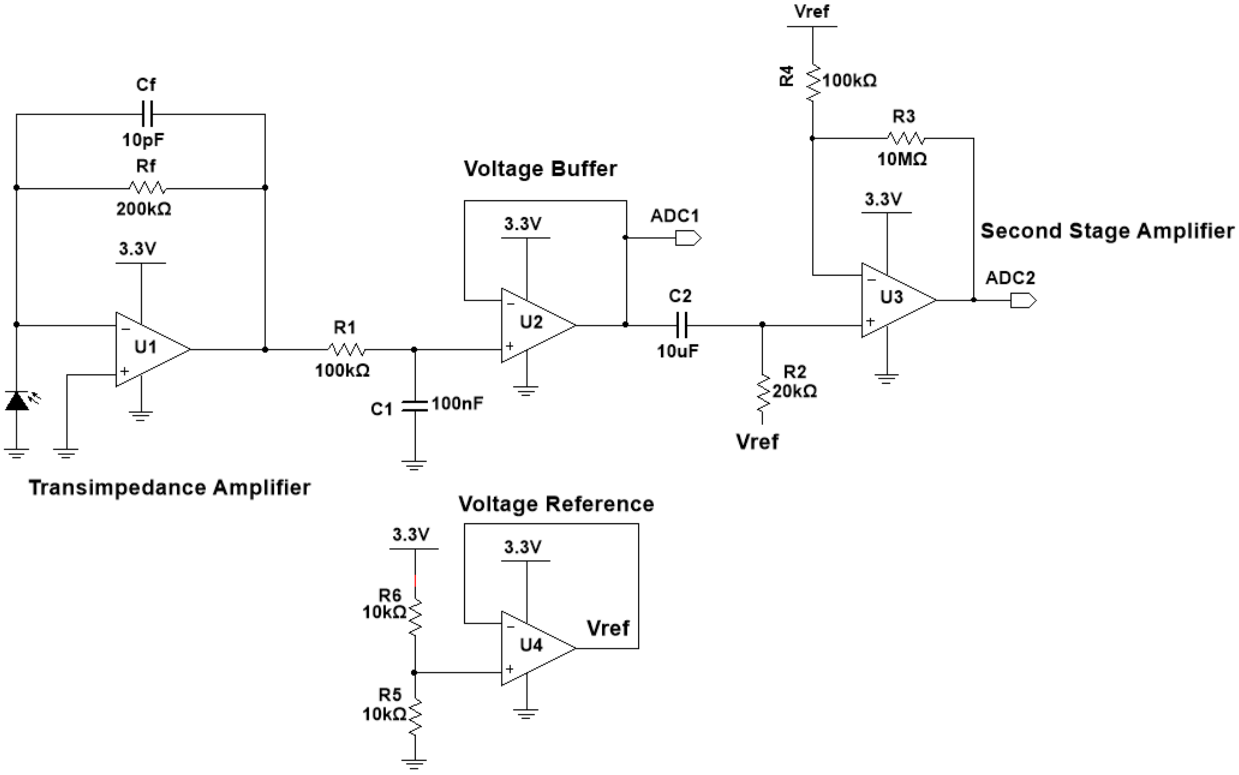


Figure 21: Analog front-end circuit.

The filters' design relied on the appropriate bandwidth of the PPG signal, where the fundamental frequency ranges from approximately 1 Hz [57]. With these design constraints, a high pass of 0.7 Hz was constructed to block the initial DC signal (high pass filtering) and a low pass of 15 Hz to remove high-frequency noise such as power line frequency (60 Hz) and any harmonics. A voltage buffer is placed between the filters to prevent loading effects. This subblock represents the analog signal conditioning described in Figure 16.

The design of the AC gain stage takes into consideration of a Texas Instrument reference design [58]. The design utilized a non-inverting amplifier with a high-pass filter preceding the non-inverting input terminal. A mid-supply reference is required with a single supply system due to the pulsatile (AC) nature, which may contain negative voltage values. Therefore a reference circuit is necessary to bias the AC component of the signal within the single supply range of the amplifier. This voltage reference circuit consists an op-amp and a voltage divider at the non-inverting input terminal. This circuit topology is illustrated in the bottom circuit of Figure 21.

Relating to this new circuit topology for the AC coupled non-inverting gain stage, the gain of the amplifier is given by the following equation,

$$G = 1 + \frac{R7}{R8}. \tag{11}$$

In order to ensure proper functionality with our design, a simulation was performed with LTspice according to Figure 21. The photodiode was modeled as a current source. The resulting transient simulation given is below:

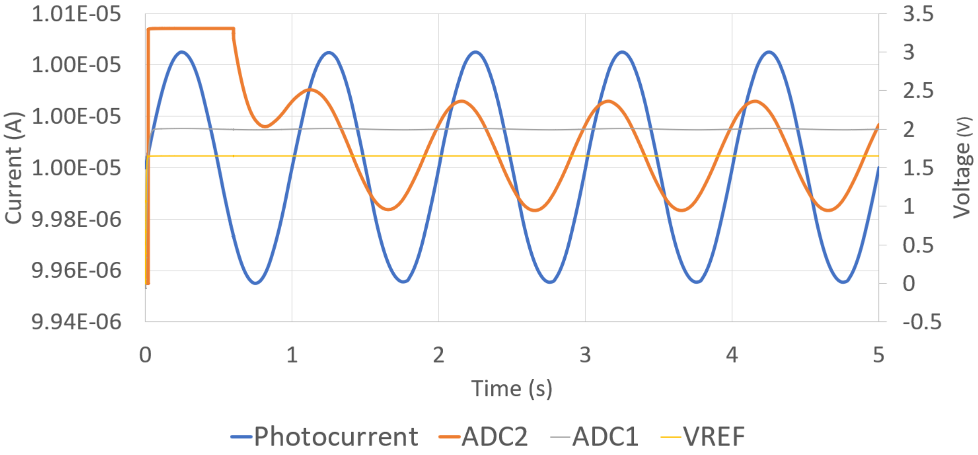


Figure 22: LTspice AFE transient simulations.

In Figure 22, the blue waveform corresponds to the input to the system (simulated photocurrent) and corresponds to the left y-axis. The remaining signals correspond to the voltage y-axis scale. The gray wave form is the simulated DC and AC component of the input waveform, which illustrates a pulsatile component superimposed on the DC signal. Vref is the midsupply voltage reference (1.65 V) needed for the AFE to bias up the AC signal to utilize the full range of the single power supply system. The final orange waveform is the amplified AC signal. There are initial transients in the startup, which shows a slight distortion. This eventually settles into the appropriate AC signal. Additionally, an AC analysis was also performed, as shown in Figure 23 with the blue solid-line graph indicating the magnitude of the signal and the orange line with the phase response.

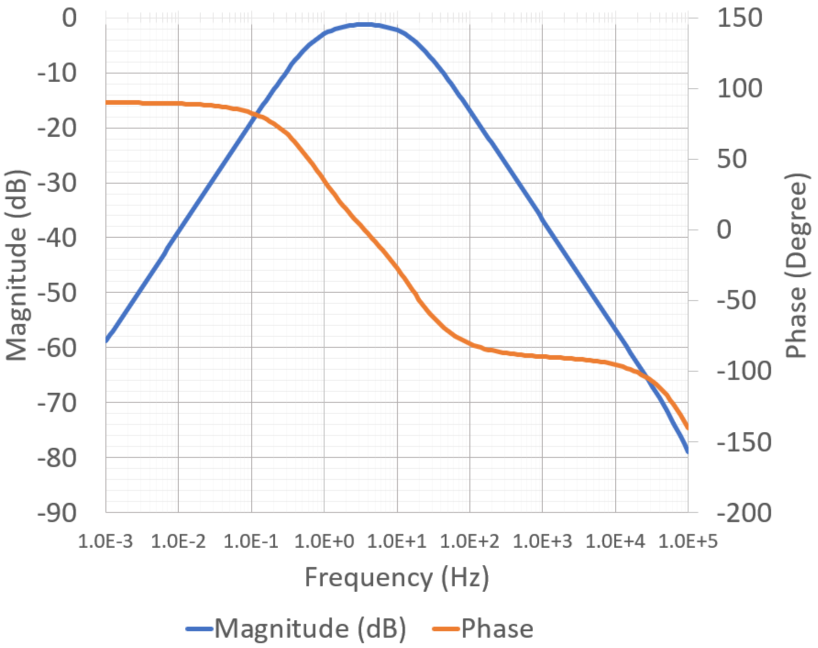


Figure 23: LTspice AFE magnitude and phase response.

When utilizing the LTspice cursors, the respective 3dB cutoff frequency is 0.785 Hz and 15.9 Hz, thereby showing consistency with the simulations that a bandpass filter can be implemented to filter the desired frequencies.

4.4. Low Dropout Voltage Regulator

Our system requires a simple power management system, where it delivers the appropriate operating voltage and current to our pulse oximeter system. A proposed solution is a low dropout voltage regulator (LDO), to provide an "inexpensive way to regulate an output voltage that is powered from a higher voltage input" [59]. The Linear Technology's LT1763 was chosen for the project's LDO due to the convenience of a SO-8 package and desired specification for the output voltage of 3.3 V. This LDO is also capable of supply up to 500 mA with 300 mV of dropout and was designed for battery powered applications with a low quiescent current of 30 µA. Because of these reasons the LT1763 was chosen for our design. From the data sheet, the recommended circuit configuration of the regulator is illustrated in the following Figure 24.

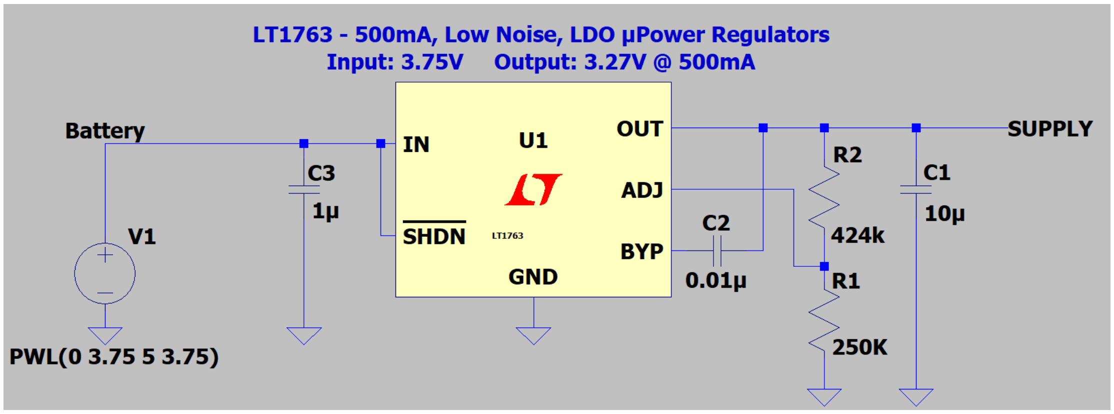


Figure 24: LT17633 Datasheet Circuit.

Additionally, the datasheet detailed a formula to solve for the required resistances for a desired output voltage. From this equation, Vout is specified as 3.3 V (desired output voltage), R1 as 250 kΩ, and I_{ADJ} as 30 nA. All these values are based on the datasheet's recommendation, and the only unknown is R2 (which was calculated to be approximately 424 kΩ), I_{ADJ} refers to the control current flowing into the ADJ/Sense pin of the regulator to adjust output voltage.

$$V_{\text{OUT}} = 1.22V(1 + \frac{R2}{R1}) + (I_{\text{ADJ}})(R2) \tag{12}$$

This was necessary for simulation purposes to ensure that the regulator can be configured to output the necessary voltage. However, when ordering the part, we chose for a fix 3.3 V variant to eliminate the need for an adjustable output voltage supply. This was done to reduce the number of components required for the overall system. Although 10 µF was utilized in the simulations, the minimum recommended capacitor is 3.3 µF, which is required to prevent instability at the output. The following figure demonstrates the respective input voltage into the LDO and the corresponding output voltage.

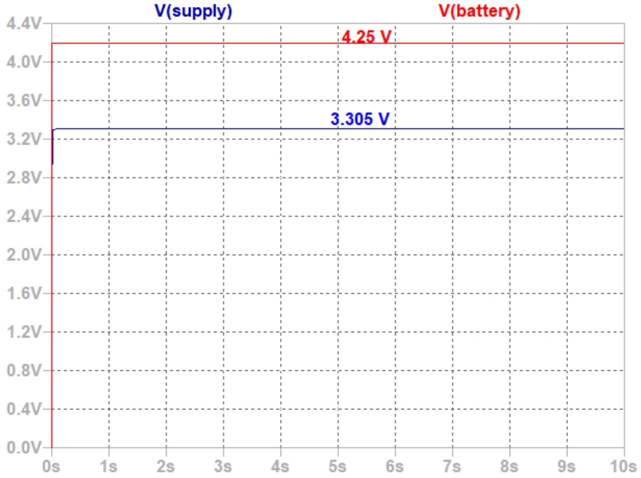


Figure 25: LT1763 Simulation.

4.5. Analog-to-Digital Converters

In this portion, the analog-to-digital converters and digital-to-analog Converters utilized in this project are the integrated units within the ESP32. The ADC is characterized by a 3.3 V reference voltage and a 12-bit output sample. Because we want to reduce the overall noise in our system, we tried to model and account for the ESP32's ADC's non-linearities. This is especially important because could be dealing with signals in the millivolt range.

To model the ADC's non-linearity we swept the input of the ADC from 0V to 3.3 V over a short period of time. Simultaneously we were recording the output of the ADC's digital signal. The plot below shows the results of our characterization of the ADC's non-linearities:

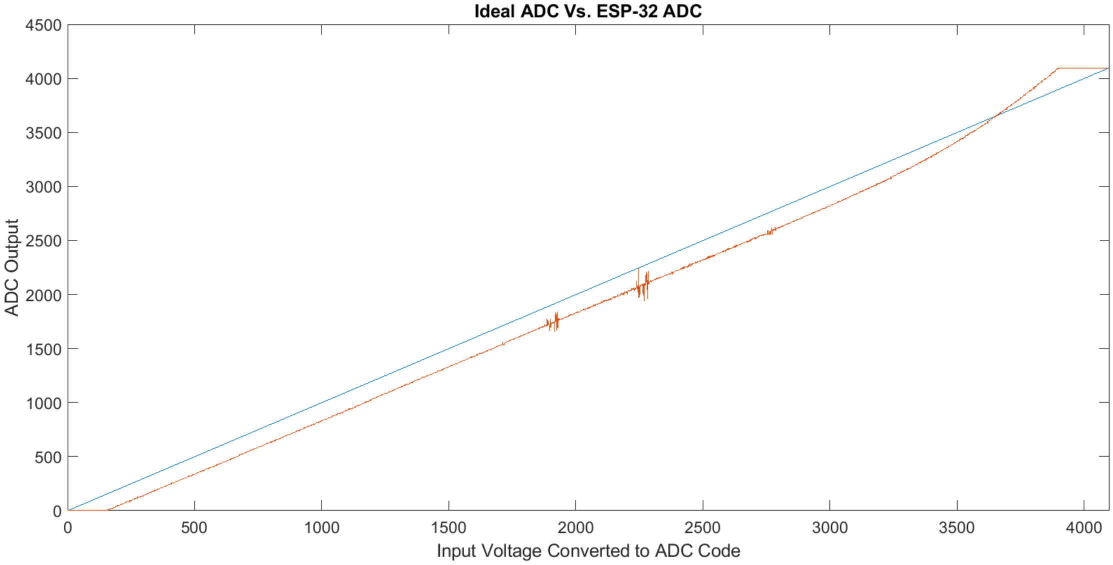


Figure 26: Linearity characterization of the ESP-32 ADC.

In the Figure 26, the X axis represents the expected output of the ADC and the Y axis is the actual result of the ADC after the sweep. The blue line represents the 'perfect' ADC with a perfect output. The red line represents the ADC onboard the ESP32. We can see that beyond a value of 3250 the ADC does not produce reliable results. This means that we should try to keep the input of the ADC below 2.61V to ensure that our signal is not distorted by non-linearities.

4.6. Microcontroller, Wireless, App, Firebase integration

This section first discusses the microcontroller and wireless decision process. The latter half focuses on the microcontroller functionality, the Android application implementation, the Firebase realtime database that backs up the smartphone application, and the wireless communication that was established to provide an end-to-end integration for this project.

4.6.1. Microcontroller Options

The microcontroller options that are considered for the project are (i) ESP32 Development Board, (ii) Wemos D1 mini, (iii) ATTiny85, (iv) Teensy, and (v) Particle Photon. The advantages and disadvantages of each option are detailed in the value analysis chart in Appendix B.

4.6.2. Microcontroller Value Analysis

For the value analysis process, we use following criteria: (i) CPU, (ii) memory, (iii) storage, (iv) wireless communications, (v) dimensions, (vi) cost, (vii) open-source, (viii) power consumption, (ix) ADC, and (x) DAC.

The weights and score scales for each of the criteria are detailed in Appendix B. These weights were chosen based on the importance of each criterion for our specific application. Each weight is on a scale from 1-5 and is located at the row of the corresponding criterion, at the third column under each microcontroller board. The rating of each microcontroller for each criterion, which is also on a scale from 1-5, can be found in the second column of that row. The microcontroller with the highest score was the ESP32 Development Board from Espressif Systems, in this microcontroller value analysis.

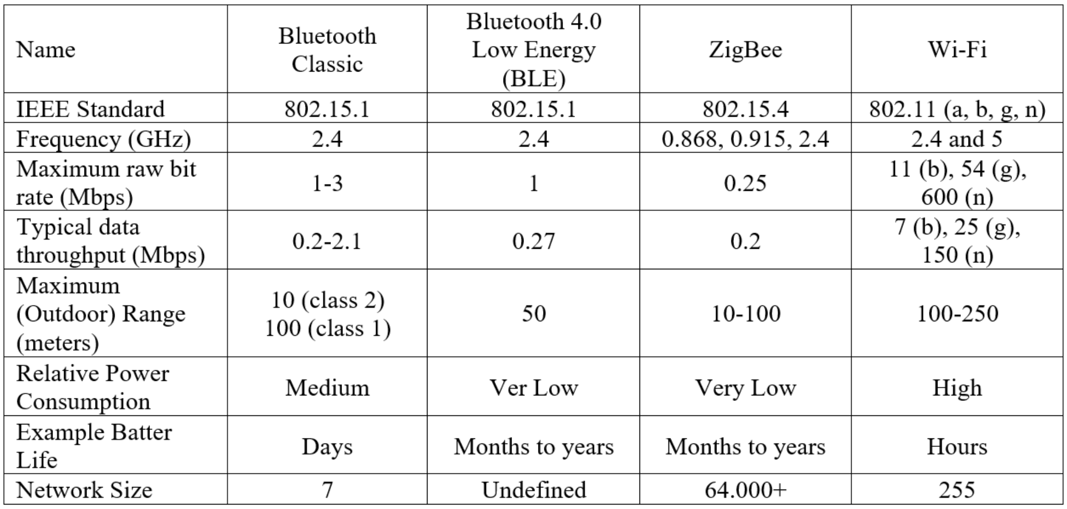
The most important reasons for this choice are summarized as follows. The ESP32 has the Tensilica Xtensa 32-bit LX6 microprocessor as its main processor and consists of two cores. The ESP32 Development Board has 520KB RAM, which was also the largest. Having enough memory is essential for this project since it is required to store multiple minutes of PPG waveform data at a time. Furthermore, it has a clock frequency of up to 240 MHz, which was the highest compared to the other four microcontrollers. The clock frequency is important in our application because while our signal may be in the 1-10Hz range, our device still needs to properly sample the signal at 250Hz and store the data in the RAM. Originally, we also believed we were going to implement data-extraction algorithms on the ESP32 rather than in the phone application. In this case, a microcontroller with a high-frequency clock is important to continuously sample, store, and process the data from our analog-front end circuitry.

One of the project goals is to transmit the collected data wirelessly, so choosing a microcontroller with wireless communication capabilities was vital. The ESP32 has built-in Wi-Fi, Bluetooth BR/EDR and Bluetooth Low Energy capabilities, which gave the team a satisfactory array of wireless protocols. The ESP32 was also one of the smallest in terms of dimensions, which is important because we aim to create a portable wearable device. Other advantages include its low cost and the fact that it is an open-source platform with detailed documentation. Compared to the other microcontrollers, it was rather average in terms of power consumption. However, the ESP32 offers different sleep modes, such as modem sleep mode, light sleep mode, deep sleep mode, and hibernation mode, that can drastically reduce power consumption.

Another point of interest for selecting the ideal microcontroller for this application was to choose one with a high-quality ADC with high resolution to have increased accuracy when sampling the PPG waveform for the red and infrared LEDs. The ESP32 integrates two 12-bit SAR ADCs supporting 18 measurement channels. Although one of the two ADCs has some restrictions in its use, we only required one ADC and two channels to extract all the information needed for this application. Finally, the last criterion for selecting a microcontroller was the DAC. The ESP32 has two 8-bit DACs, which was ideal for this application since one would be utilized to control the current flowing through the red LED, and the other would be used to control the current flowing through the infrared LED.

4.6.3. Wireless Communication Protocol Selection

Out of all the wireless communication protocols that we researched, we decided to select Bluetooth Low Energy for this application, as it displays highly reliable connectivity between devices in a short distance, which fits our project since we want to transmit information from a wearable device to the user's phone. It is compatible with many phone devices, and it generally has a convenient connection process, something that sets it apart from the ZigBee communication protocol. Furthermore, the lower power consumption compared to the standard Bluetooth protocol can match ZigBee's very low power consumption. On the other hand, when comparing Bluetooth Low Energy to Wi-Fi, we noticed that Wi-Fi, even though it is a popular, fast, and reliable protocol, requires a significant amount of power and is relatively excessive for our application. A brief comparison of the protocols that were discussed is displayed in Figure 27.



Name	Bluetooth Classic	Bluetooth 4.0 Low Energy (BLE)	ZigBee	Wi-Fi
IEEE Standard	802.15.1	802.15.1	802.15.4	802.11 (a, b, g, n)
Frequency (GHz)	2.4	2.4	0.868, 0.915, 2.4	2.4 and 5
Maximum raw bit rate (Mbps)	1-3	1	0.25	11 (b), 54 (g), 600 (n)
Typical data throughput (Mbps)	0.2-2.1	0.27	0.2	7 (b), 25 (g), 150 (n)
Maximum (Outdoor) Range (meters)	10 (class 2) 100 (class 1)	50	10-100	100-250
Relative Power Consumption	Medium	Ver Low	Very Low	High
Example Batter Life	Days	Months to years	Months to years	Hours
Network Size	7	Undefined	64.000+	255

Figure 27: Comparison of wireless communication protocols.

4.6.4. Microcontroller Functionality

The functionality of the microcontroller can be divided into two different modules: extraction of the PPG waveform for each LED (i.e. red and IR) and wireless transmission of PPG waveform data to our Android application. For the former, the microcontroller was required to turn the LEDs on and off, sample the DC and AC part of the PPG waveform for each LED and either average them (i.e. DC) or store the data values in an array (i.e. AC), so they are ready for transmission. For the latter, the microcontroller was responsible for connecting with the user's smartphone and transmitting the stored data to our Android application over a BLE link when the connection between the two devices has been established.

To complete the first module while ensuring high accuracy, our team decided to utilize a 500 Hz timer interrupt. The interrupt service routine (ISR) only signaled the interruption and delegated the more complex handling to the main loop of the program. This implementation ensured that the ISR would run as fast as possible without engaging in long operations to avoid running into runtime issues. In the main loop, when the timer interrupt was triggered, the program went into a short critical section to guarantee synchronization between the main loop and the ISR. The shared variable used to signal the program of the timer interrupt occurrence was modified. Then, the interrupt handling code is executed.

At that point of the program, a finite-state machine was executed. Initially, the red LED turned on and off at a 250 Hz frequency with a 50% duty cycle. Moreover, 8 raw samples relating to the AC portion of the PPG signal were read in by the ADC every time the red LED turned on. Then, these 8 raw samples were averaged to generate one sample for every time the red LED was on. Therefore, the sampling rate for the PPG signal was equal to 250 Hz. This sampling frequency was definitely sufficient for our application, since the normal frequency range of a PPG signal is around 0.5 Hz and 5 Hz, which is much less than 125 Hz. Consequently, the Nyquist theorem is satisfied with these conditions. The total number of samples that were averaged from the 8 raw data values, was 16384 for the PPG signal related to the red LED. This equates to around 1 minute and 5.5 seconds of sampling. The equation that was used to derive the total time it took to record all the samples is displayed as

$$Sampling\ Duration\ [s] = \frac{\#\ of\ Samples}{Sampling\ Rate\ [Hz]}. \tag{13}$$

Other than sampling for data values of the AC-related part of the PPG signal, the microcontroller also sampled one value at a time for the DC part of the signal whenever the red LED was turned on, meaning that the DC signal of the PPG waveform was sampled at a rate of 250 Hz too. The DC signal was sampled by the ADC on another GPIO pin (ADC1) connected after the low-pass filter and buffer amplifier in Figure 21. All the samples of the DC part of the signal were averaged for each LED session. Once all the samples were read in for the red LED, the microcontroller switched to turning the IR LED on and off and performed an identical procedure in order to receive data for the PPG waveform from the IR LED as well. The brightness of the LEDs was controlled by the two DACs, which were set 0.906 V. Once all samples were stored and both sampling duration times were completed, the data containing the DC components and the AC-related PPG waveform for the red and IR LEDs were prepared for wireless transmission.

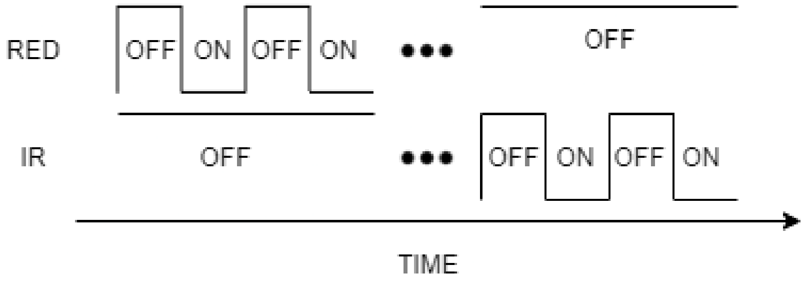


Figure 28: LED timing.

To perform wireless transmission of the collected data from the device to the user's smartphone application, two UUIDs (Universally Unique Identifier) were defined; one for the BLE service and one for the BLE characteristic. Then, a BLE device was initialized with the name "MQP" and was set as a server. The Android application was the client for this project. Once our device was defined as a server, a service for that BLE server was created with the UUID defined earlier. Subsequently, a characteristic for that service was defined with the other UUID. The characteristic's properties were READ and WRITE. When this structure was set, our device started the service and also started advertising for the client (i.e. smartphone application).

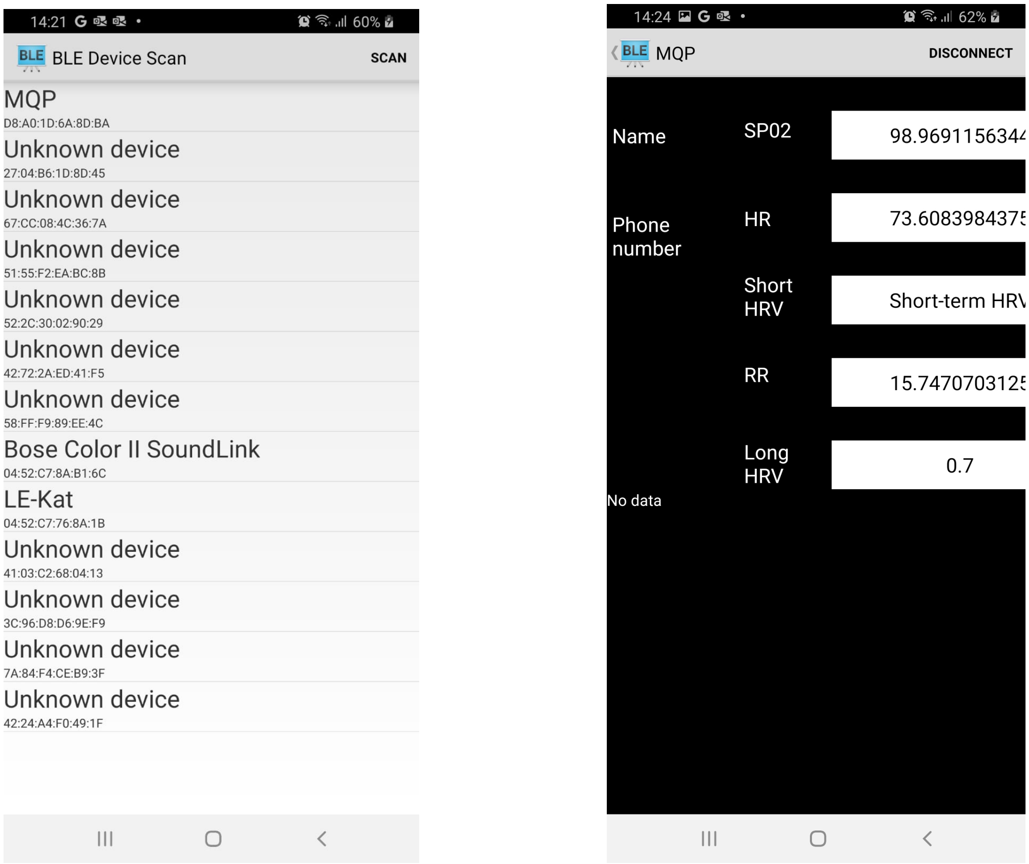
To check if a connection had been established over BLE, the "BLEServerCallbacks" protocol was utilized. A class was defined that inherited the protocol, where the methods "onConnect(BLEServer* pServer)" and "onDisconnect(BLEServer* pServer)" were implemented. The first method was called when the server established a connection, while the second was called when the server destoroyed a connection. Therefore, when "onConnect(BLEServer* pServer) was called a global variable was set to "true", whereas when "onDisconnect(BLEServer* pServer)" was called, that variable was set to "false". Finally, when the two devices were connected, our device transmitted all the stored values sequentially to the app over BLE.

4.6.5. Android Application Implementation

This section describes the functionality of the Android application that was developed for this project. The flow and an overview of the actions taken from the app in each step are detailed below.

The smartphone application started with the "Device Scan Activity", which scans for all the BLE devices in close vicinity. At the screen, a list was displayed with all the BLE devices that were detected in the scanning. BLE device scanning lasted for 30 seconds. However, if the user would like to scan for longer periods, pressing the button "SCAN" at the top right corner initiated scanning again for 30 seconds. If the user would like to stop scanning they could press the "STOP" button any time while scanning for BLE devices. Figure 29 displays the screen while in this activity.

Subsequently, the user had the option to choose the BLE device they would like to connect to. In our case, the BLE device named "MQP" was selected since that was the the name we gave our BLE device. When connected, the application moved to "Device Control Activity". In this activity, the app was repeatedly receiving the PPG waveform data that were being transmitted from the microcontroller. Once all the data was received, the app performed the implemented algorithms for oxygen saturation, heart rate, respiratory rate, and heart rate variability. In this activity, the app also displayed the most recent measured value for SpO_2, heart rate, respiratory rate, and long-term heart rate variability that resulted from our algorithms and were pulled from the Firebase Realtime Database. Moreover, it included five different buttons for blood oxygen level, heart rate, respiratory rate, short-term and long-term heart rate variability respectively. Figure 29 (right) shows the layout of this activity.



MQP
D8:A0:1D:6A:8D:BA
Unknown device
27:04:B6:1D:8D:45
Unknown device
67:CC:08:4C:36:7A
Unknown device
51:55:F2:EA:BC:8B
Unknown device
52:2C:30:02:90:29
Unknown device
42:72:2A:ED:41:F5
Unknown device
58:FF:F9:89:EE:4C
Bose Color II SoundLink
04:52:C7:8A:B1:6C
LE-Kat
04:52:C7:76:8A:1B
Unknown device
41:03:C2:68:04:13
Unknown device
3C:96:D8:D6:9E:F9
Unknown device
7A:84:F4:CE:B9:3F
Unknown device
42:24:A4:F0:49:1F

Name
Phone number
No data

SP02	98.9691156344
HR	73.608398437
Short HRV	Short-term HR\
RR	15.747070312
Long HRV	0.7

Figure 29: BLE monitor in Android application (left) and the layout of "Device Control Activity" (right).

Other than the button for short-term heart rate variability, when the user pressed the buttons, the application moved to a new activity which displayed a graph of all the measured values for the parameter corresponding to the button pressed. The graphs pulled all the recorded values from Firebase, because the results from the algorithms were instantly pushed to the database. An example of one of these graphs is shown in Figure 30.

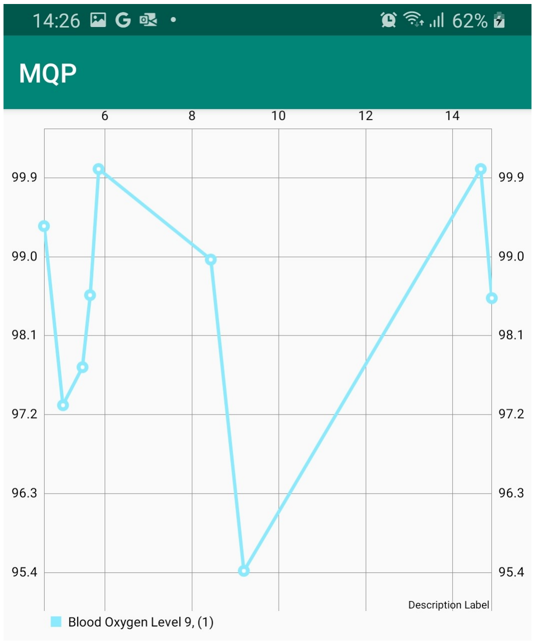


Figure 30: Graph of all the recorded values for oxygen saturation.

In the case where the button corresponding to short-term heart rate variability was pressed, the user is prompted to a new activity with six buttons, one for each short-term HRV index that was measured. The layout of the activity mentioned is depicted in Figure 31.

When one of these six buttons were pressed, like in the device control activity, the application moved to a new activity displaying the recorded values of a short-term HRV index measurement which corresponded to the button that was pressed. Similarly, the activity pulled from Firebase in order to populate the graph with the values that were returned by the algorithms and pushed into the database. In the Figure 32 below, a graph of a short-term HRV index, named SDNN (i.e. standard deviation of NN intervals) is shown.

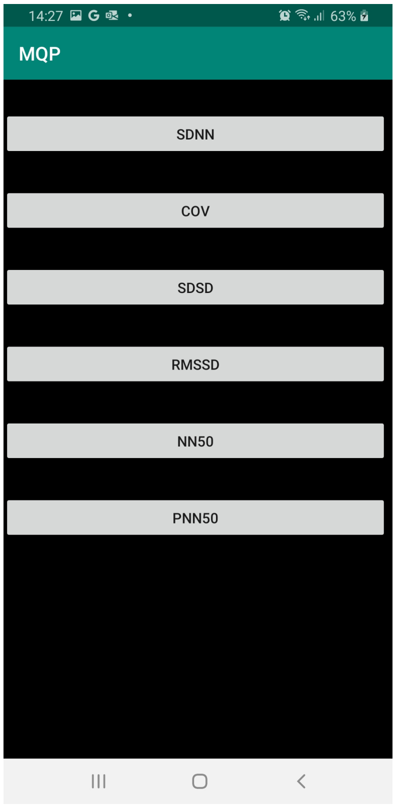


Figure 31: Layout of activity showing the six short-term HRV indices measured.

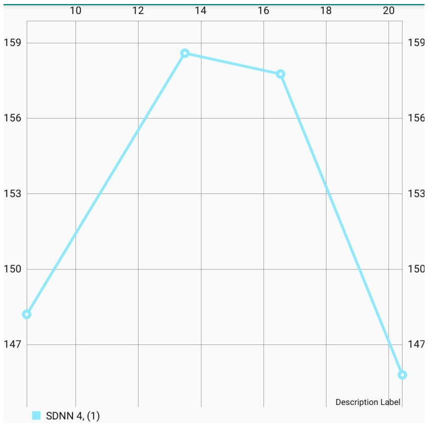
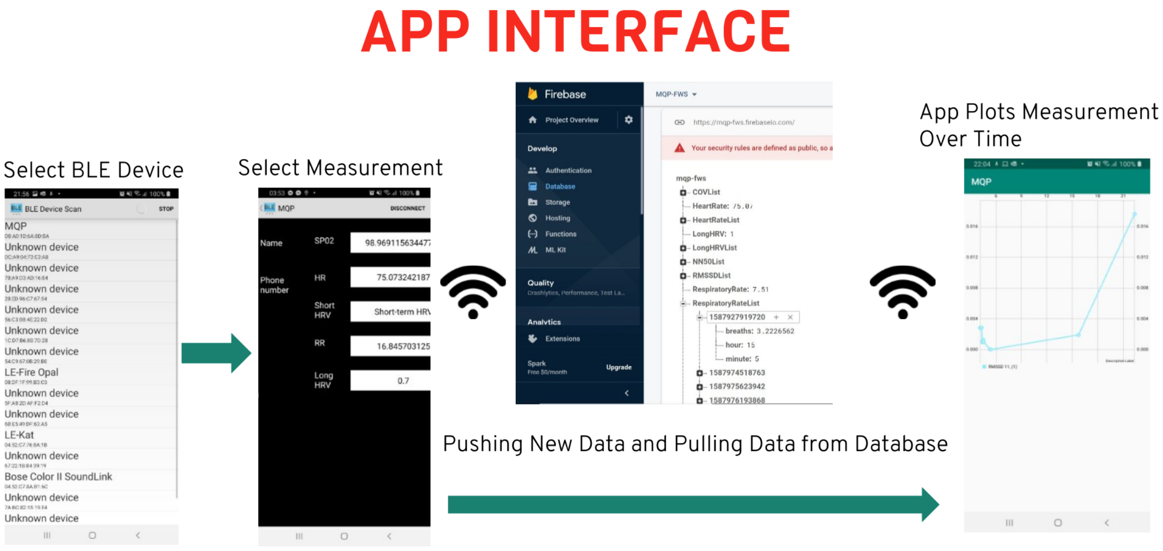


Figure 32: SDNN measurement plotted in the user's phone application.

A brief overview of the flow of our smartphone application which was described above, can be seen in Figure 33.



MQP
DB:A0:1D:6A:8D:BA
Unknown device
DC:A9:04:73:C3:A8
Unknown device
78:A9:D3:AD:16:84
Unknown device
28:ED:96:C7:67:54
Unknown device
56:C3:B8:4E:22:D2
Unknown device
1C:D7:B6:80:7D:28
Unknown device
54:C9:67:0B:20:B0
LE-Fire Opal
08:DF:1F:99:B3:C3
Unknown device
5F:A8:2D:AF:F2:D4
Unknown device
6B:E5:49:DF:63:A5
LE-Kat
04:52:C7:76:8A:1B
Unknown device
67:22:1B:84:39:19
Bose Color II SoundLink
04:52:C7:8A:B1:6C
Unknown device
7A:BC:82:15:19:E4
Unknown device

Name	SP02	98.969115634477
Phone number	HR	75.073242187
	Short HRV	Short-term HRV
	RR	16.845703125
	Long HRV	0.7

https://mqp-fws.firebaseio.com/

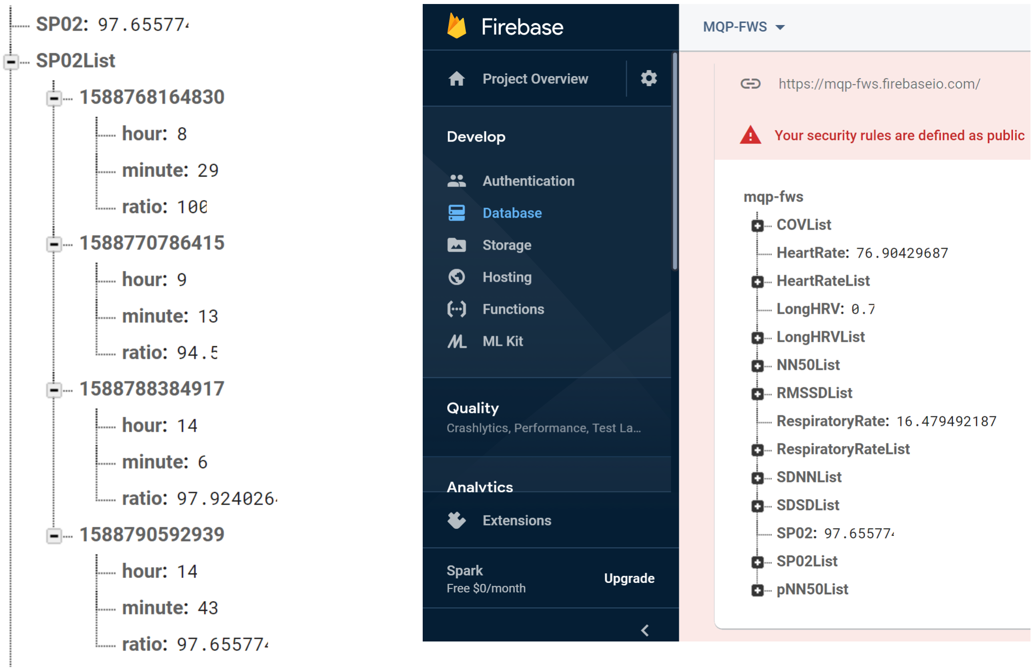
Your security rules are defined as public, so a

```
mqp-fws
├ COVList
├ HeartRate: 75.07
├ HeartRateList
├ LongHRV: 1
├ LongHRVList
├ NN50List
├ RMSSDList
├ RespiratoryRate: 7.51
└ RespiratoryRateList
   ├ 1587927919720
   │  ├ breaths: 3.2226562
   │  ├ hour: 15
   │  └ minute: 5
   ├ 1587974518763
   ├ 1587975623942
   └ 1587976193868
```

Figure 33: High-level flow of our Android application.

4.6.6. Firebase Realtime Database Structure

This section focuses on the structure of the database which was used in the project. The Android application utilized the realtime database's resources by pushing the outputs from the implemented algorithms to it and then pulling them from it when the application was instructed to show a graph of one of the health parameters that were measured. Therefore, structuring the data correctly in the database was essential. All the health parameters (i.e., every health parameter other than the short-term HRV indices) that were displaying the most recently measured value had an individual entry in the database where that value was held. Furthermore, each health parameter had its list in the database, where their children (i.e., elements of the list) were organized with UNIX timestamps. Each timestamp had three children. One for the resulting measurement that was outputted from the algorithm, one for the hour and one for the minute for which this measurement was taken. We observe the overall and the individual structure of the health parameters and demonstrate in Figure 34, on the right, where all the entries in the database are listed for each health parameter. Entries are within their locations where the application pushed the new measurements as elements and the entries that were used for the application to pull the most recent measurement for each parameter. On the other hand, the left side of Figure 34 shows the two different entries in more detail since the UNIX timestamps and their three children can be seen.



```
SP02: 97.65577
SP02List
    1588768164830
        hour: 8
        minute: 29
        ratio: 100
    1588770786415
        hour: 9
        minute: 13
        ratio: 94.5
    1588788384917
        hour: 14
        minute: 6
        ratio: 97.924026
    1588790592939
        hour: 14
        minute: 43
        ratio: 97.65577
```

```
mqp-fws
    COVList
    HeartRate: 76.90429687
    HeartRateList
    LongHRV: 0.7
    LongHRVList
    NN50List
    RMSSDList
    RespiratoryRate: 16.479492187
    RespiratoryRateList
    SDNNList
    SDSDList
    SP02: 97.65577
    SP02List
    pNN50List
```

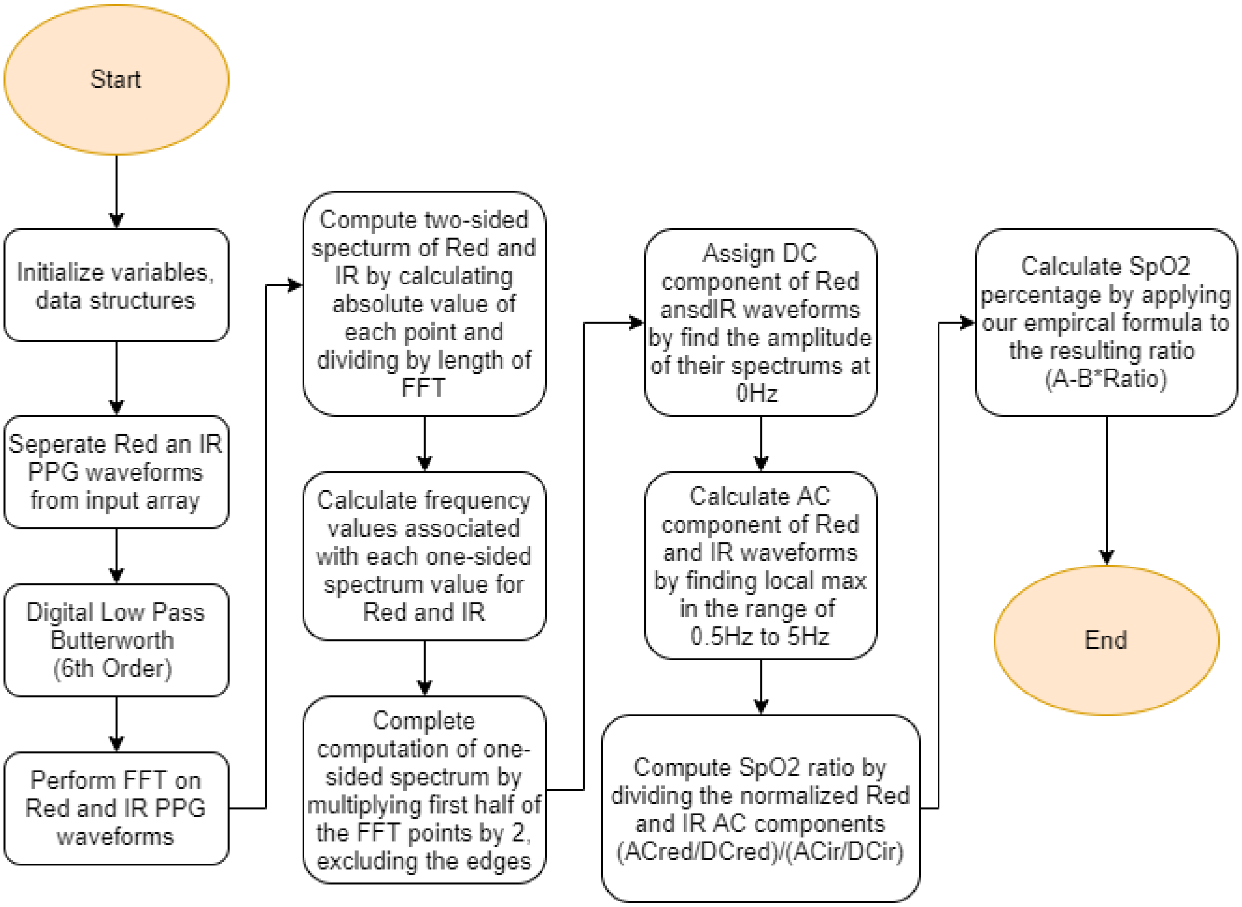
Figure 34: SpO_2 database information (left) and overall Firebase database (right).

4.7. Data Extraction Algorithms

This section discusses the various algorithms implemented within the Android mobile application to extract vital sign parameters such as peripheral oxygen saturation, heart rate, respiration rate, and heart rate variability from the collected PPG signals.

4.7.1. Peripheral Oxygen Saturation (SpO_2)

The focal point of pulse oximetry is Peripheral Oxygen Saturation. The process to extract SpO_2 is illustrated in Figure 35.



Start
Initialize variables, data structures
Seperate Red an IR PPG waveforms from input array
Digital Low Pass Butterworth (6th Order)
Perform FFT on Red and IR PPG waveforms
Compute two-sided specturm of Red and IR by calculating absolute value of each point and dividing by length of FFT
Calculate frequency values associated with each one-sided spectrum value for Red and IR
Complete computation of one-sided spectrum by multiplying first half of the FFT points by 2, excluding the edges
Assign DC component of Red ansdIR waveforms by find the amplitude of their spectrums at 0Hz
Calculate AC component of Red and IR waveforms by finding local max in the range of 0.5Hz to 5Hz
Compute SpO2 ratio by dividing the normalized Red and IR AC components (ACred/DCred)/(ACir/DCir)
Calculate SpO2 percentage by applying our empircal formula to the resulting ratio (A-B*Ratio)
End

Figure 35: SpO_2 extraction algorithm flow.

In order to obtain the blood oxygen level of the subject, spectral analysis was performed on the 2 minutes and 10 seconds of data that were transmitted from our device over BLE. Once the AC and DC components of the red and IR PPG waveform were calculated, an empirical formula that was derived from validating the results of the algorithms was implemented, to determine the SpO_2 value. This algorithm was developed in Android Studio and is analytically explained below.

From the algorithm code in Appendix G.2, the input data that were transmitted over BLE from our prototype to the Android application were populated into two arrays of length equal to 16384, which is the same as the number of samples that were transmitted for red and IR respectively. The first array held all the AC-related PPG data for the red LED, while the second contained all the AC-related PPG data for the IR LED. Furthermore, two elements from the data received over BLE were the DC components for the red and IR PPG waveforms. These values were read and stored in order to be used later. Next, a digital sixth order Butterworth Low Pass filter with a cutoff frequency of 5 Hz was applied to the AC-related PPG waveform data in order to reduce the noise of these two signals which were read by the microconroller's ADC and to accentuate the desired frequencies that were used to extract the AC components for SpO_2. Subsequently, the Fast Fourier Transform (FFT) was executed using two arrays of length 16384 for both the AC-related data of the red and the IR LED . The FFT was implemented in the app and was used as a helper function from the SpO_2 algorithm function. After transitioning to the frequency domain, the two-sided spectrum of the red and IR PPG waveforms was computed by calculating the absolute value of each data point and dividing by the length of the FFT, which is equal to 16384. Moreover, the one-sided spectrum for the two PPG waveforms was computed. Initially, the frequency values associated with each one-sided spectrum value for red and IR respectively was calculated. From this calculation, we were able to measure the resolution of the transforms, which was equal to 0.0153 Hz. The following equation displays how the resolution was deduced

$$Resolution\,[Hz] = \frac{Sampling\ Rate}{Length\,of\ FFT}. \tag{14}$$

Then, the computation of the one-sided spectrum was completed by multiplying the first half of the resulting values from the two-sided spectrum by 2, excluding the two extremes of that range. Finally, the AC component of the red and IR waveforms was determined by locating the peak in the full span of the cardiac frequency range (0.5 - 5 Hz) [26]. This process is illustrated in the following manner in Figure 36.

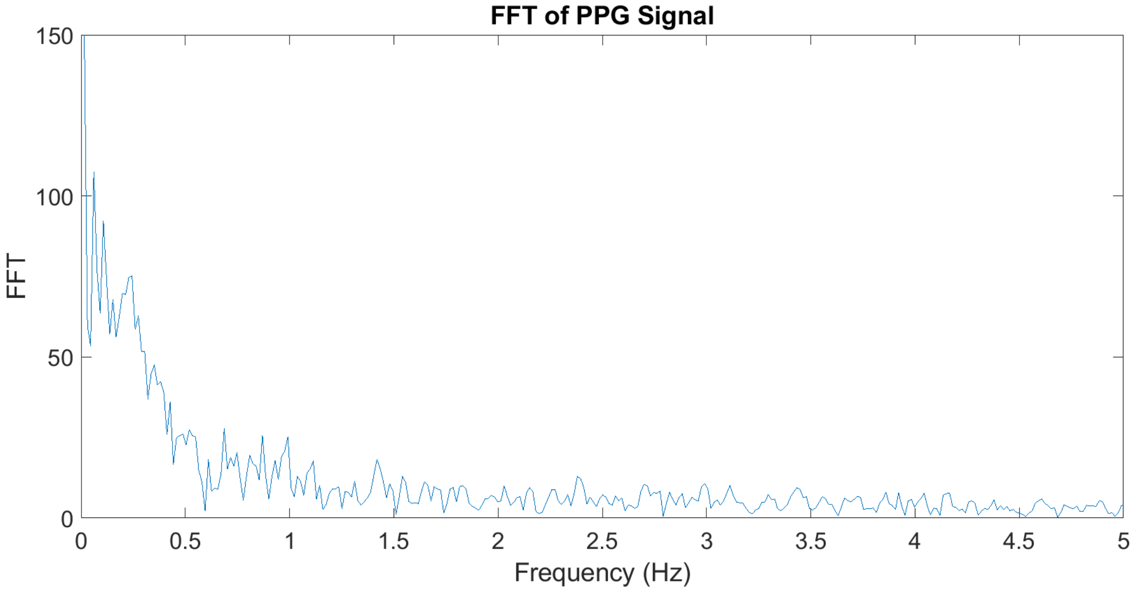


Figure 36: Spectral analysis of red and IR PPG signals.

From these extracted values, the algorithm was able to compute the SpO$_2$ ratio "R" by dividing the normalized RED and IR AC components, which was given by the following equation

$$R = \frac{\left(\frac{redACval}{redDCval}\right)}{\left(\frac{irACval}{irDCval}\right)}. \tag{15}$$

Finally, the algorithm was able to calculate the oxygen saturation (SpO$_2$) percentage by applying the following empirical formula,

$$SpO_2\% = 115 - R * 25, \tag{16}$$

[32] to the resulting ratio "R". The constants displayed in the equation above were validated by concurrently testing our device against a commercial pulse oximeter.

4.7.2. Heart Rate

The process of extracting heart rate mirrors closely the SpO$_2$ extraction. However, obtaining heart rate can be done with one PPG waveform as opposed to two when determining SpO$_2$ percentage. Additionally, the frequencies of interest for healthy patients range from 1 to 1.7 Hz (60 to 100 bpm); however, the algorithm searched up to 4 Hz to capture a greater span of the cardiac frequency range and potentially spot problematic heart rate levels. Once the peak and therefore, the harmonic was determined in this range of frequencies, the resultant is multiplied by 60 to get the final result as beats per minute. In other words, the heart rate was measured using the equation [26] [60],

$$Heart\ Rate\ [bpm] = 60 \ * \ Predominant\ Cardiac\ Frequency\ [Hz]. \tag{17}$$

The algorithm implemented to calculate heart rate for our project is detailed in the flowchart in Figure 37.

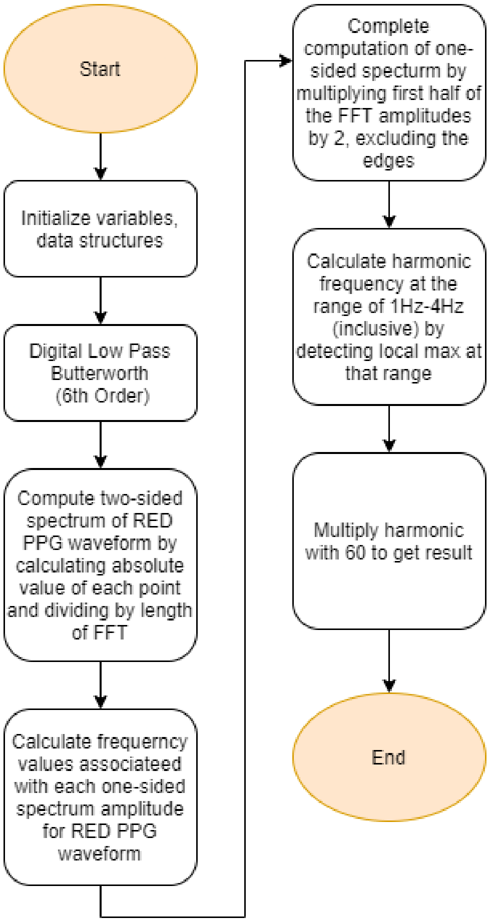

Start
Initialize variables, data structures
Digital Low Pass Butterworth (6th Order)
Compute two-sided spectrum of RED PPG waveform by calculating absolute value of each point and dividing by length of FFT
Calculate frequerncy values associateed with each one-sided spectrum amplitude for RED PPG waveform
Complete computation of one-sided specturm by multiplying first half of the FFT amplitudes by 2, excluding the edges
Calculate harmonic frequency at the range of 1Hz-4Hz (inclusive) by detecting local max at that range
Multiply harmonic with 60 to get result
End

Figure 37: Heart rate extraction algorithm.

4.7.3. Respiratory Rate

The extraction of respiratory rate from a given PPG signal follows the same protocols as the heart rate algorithm. However, the frequencies of interest are shifted towards closer to the DC component. In this case, the algorithm focused on the band between 0.05 and 0.7 Hz, which corresponds to 3 and 42 breaths per minute. The respiratory range of a healthy individual is considered to be 12 to 20 breaths per minute [61]. The predominant respiratory frequency is also multiplied by 60 to get breaths per minute, as displayed in the equation (respiratory rate denoted as "RR"),

$$RR\,[breaths\,/\,min] = 60\, *\, Predominant\ Respiratory\ Frequency\,[Hz]. \tag{18}$$

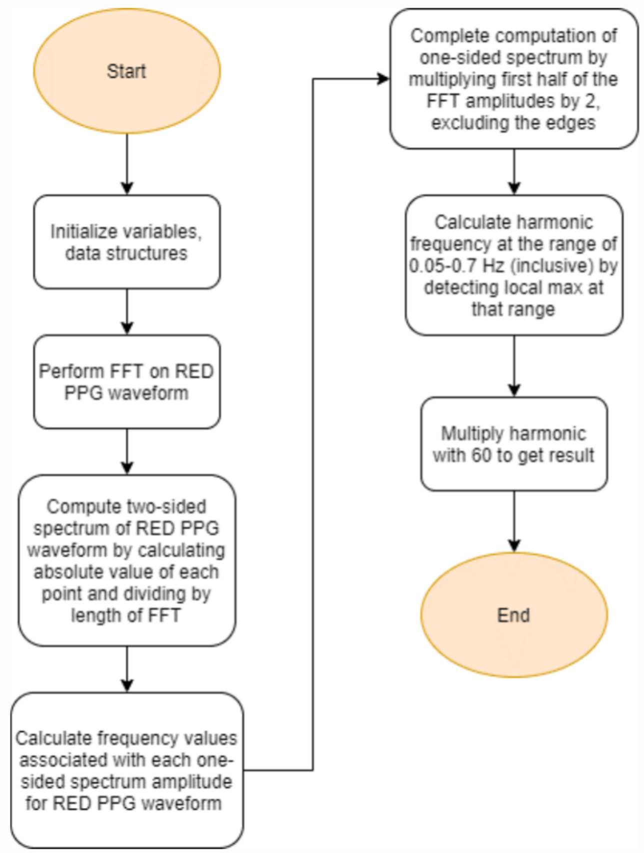

Start
Initialize variables, data structures
Perform FFT on RED PPG waveform
Compute two-sided spectrum of RED PPG waveform by calculating absolute value of each point and dividing by length of FFT
Calculate frequency values associated with each one-sided spectrum amplitude for RED PPG waveform
Complete computation of one-sided spectrum by multiplying first half of the FFT amplitudes by 2, excluding the edges
Calculate harmonic frequency at the range of 0.05-0.7 Hz (inclusive) by detecting local max at that range
Multiply harmonic with 60 to get result
End

Figure 38: Respiratory rate extraction algorithm.

An issue of concern might be that the frequency could be attenuated with the 0.8 Hz

High Pass filter implemented in the AFE. This could represent a challenge to correctly extract respiratory rate from the PPG waveform due to hardware limitations.

4.7.4. Heart Rate Variability

This section discusses the algorithms that were implemented to measure heart rate variability. Currently, there is not a universal agreement for the best HRV index. However, there are two main types of algorithms: short-term and long-term heart rate variability evaluation. For this project, algorithms were developed for both types. For the former, six HRV indices were calculated in the time domain. For the latter, spectral analysis was performed.

For the short-term heart rate variability evaluation the indices that were measured focused on the analysis of NN intervals that are the time intervals between two sequential systolic peaks as shown in Figure 39.

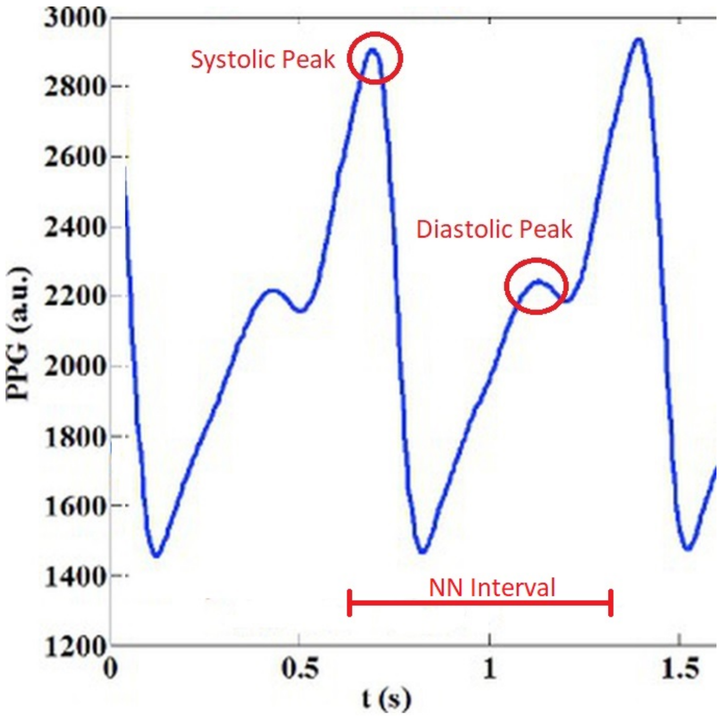


Figure 39: Short-term heart rate variability [62].

The first HRV index that was measured was SDNN, which is the standard deviation of NN intervals and was used to measure the overall variability in heart rate. Then, the coefficient of variance (i.e. COV) was calculated, which is essentially the normalized SDNN. The standard deviation of successive NN differences was also measured. This index reflects the parasympathetic influence on the heart. The number of successive NN differences greater than 50 ms was calculated. The reason for selecting the fixed threshold of 50 ms was due to the popularity of the selection of that threshold which originated from a 1984 study [63]. Finally, the proportion of successive NN differences greater than 50 ms was measured, which was introduced by Bigger[64]. It has proved to be a useful HRV index, providing diagnostic and prognostic data for different conditions [65]. More details about these indices and how they are calculated can be seen in Section 4.7.5.

For the long-term heart rate variability evaluation, the power spectral density of 24 hours of the PPG signal was calculated. Spectral decomposition of the recorded PPG signal can directly indicate changes in sympathetic and parasympathetic activity. More information for the implementation of the algorithm along with how it detects changes in the autonomic nervous system are displayed in Section 4.7.6.

4.7.5. Short-Term HRV

This section analyzes the algorithm that was implemented for the short-term heart rate variability evaluation. A list and descriptions of short-term heart rate variability indices is given in Table 2. Furthermore, the complete algorithm that was used to calculate the 6 short-term HRV indices is indicated in Figure 40.

Before starting to calculate all the short-term HRV indices, the previously mentioned sixth order Butterworth low pass filter was applied to the collected PPG signal. Afterwards, the first major step was to find the length of all the NN intervals of the filtered signal. The NN intervals were calculated by essentially locating the systolic peaks and then multiplying the signal's sampling frequency with the difference of the sequential peaks' array indices. More precisely, the algorithm found the peaks of the PPG waveform by detecting the first diastolic peak of that interval, then locating the first systolic peak by comparing data points to all of the previous 5 and all of the next 5 elements of the array that held the input PPG signal. Subsequently, the next diastolic and systolic peaks were found in a similar fashion. Once the NN intervals were found, the algorithm calculated SDNN by measuring the standard deviation of the NN intervals. COV (i.e. coefficient of variance) was calculated by dividing SDNN with the mean of the NN intervals. To calculate SDSD (i.e. standard deviation of successive NN differences) the algorithm had to first measure the difference between sequential NN intervals and place the results in a new array. Then, it measured the standard deviation of the resulting values. The RMSSD, was calculated by finding

Table 2: Short-Term HRV Indices

HRV Indices	Calculation Methods	Physiological Indications
SDNN	Standard deviation of NN intervals	Measure of the overall variability in HR
COV	Coefficient of variance (normalized SDNN)	Changes in overall variability independent of changes in mean NNinterval
SDSD	Standard deviation of successive NN differences	Reflection of parasympathetic influence on the heart.
RMSSD	Root-mean-square of successive NN differences	Reflection of parasympathetic influence on the heart.
NN50	Number of successive NN differences $>$ 50 ms	Index of vagal tone
pNN50	Proportion of successive NN differences greater $>$ 50ms (NN50 / Number of NN differences)	Normalized version of NN50, independent of HR

the rot-mean-square of the differences of sequential NN intervals. Finally, NN50 was calculated by finding the difference between two successive NN intervals and counting the number of times this difference was greater than 50ms. Then pNN50 is the proportion of succesive NN intervals that differ by more than 50ms over the total number of NN interval differences calculated.

4.7.6. Long-Term HRV

This section describes the algorithm that was used for the long-term heart rate variability evaluation. The flowchart of this algorithm is shown in Figure 41.

The first step for this algorithm to run, was to collect around 24 hours worth of PPG waveform data. Once all the PPG data are received, the algorithm performs spectral analysis to the data by initially applying the Fast Fourier Transform (FFT) and then calculating the power spectral density (PSD). Low frequency signal fluctuations reflect activity in the autonomic nervous system [26]. More precisely, sympathetic activity is defined by the power in the 0.01 to 0.15 Hz range, while parasympathetic activity is defined by the power in the 0.15 to 0.5 Hz range. Therefore, the algorithm located the greatest amplitude in both of these regions respectively. The peak in the 0.01 - 0.15 Hz region is called the "LF component", while the peak in the 0.15 - 0.5 Hz

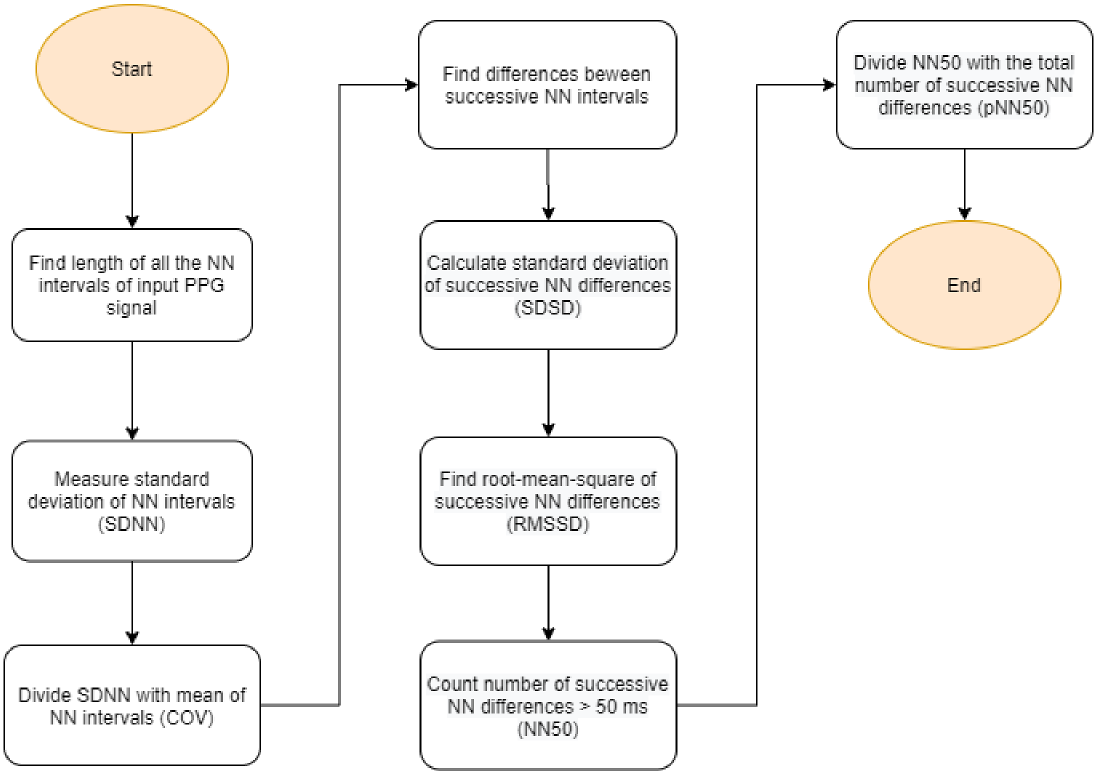

Start
Find length of all the NN intervals of input PPG signal
Measure standard deviation of NN intervals (SDNN)
Divide SDNN with mean of NN intervals (COV)
Find differences beween successive NN intervals
Calculate standard deviation of successive NN differences (SDSD)
Find root-mean-square of successive NN differences (RMSSD)
Count number of successive NN differences > 50 ms (NN50)
Divide NN50 with the total number of successive NN differences (pNN50)
End

Figure 40: Short-term HRV extraction algorithm.

region is called the "HF component". Then, the power ratio (LF/HF) was calculated which directly displays changes in sympathetic and parasympathetic activity. This ratio between sympathetic and parasympathetic activity represents long term HRV evaluation. For example, the figure below displays the power spectral density that shows the LF and HF components and their corresponding power ratios when a subject was laying down (a) compared to standing upright (b).

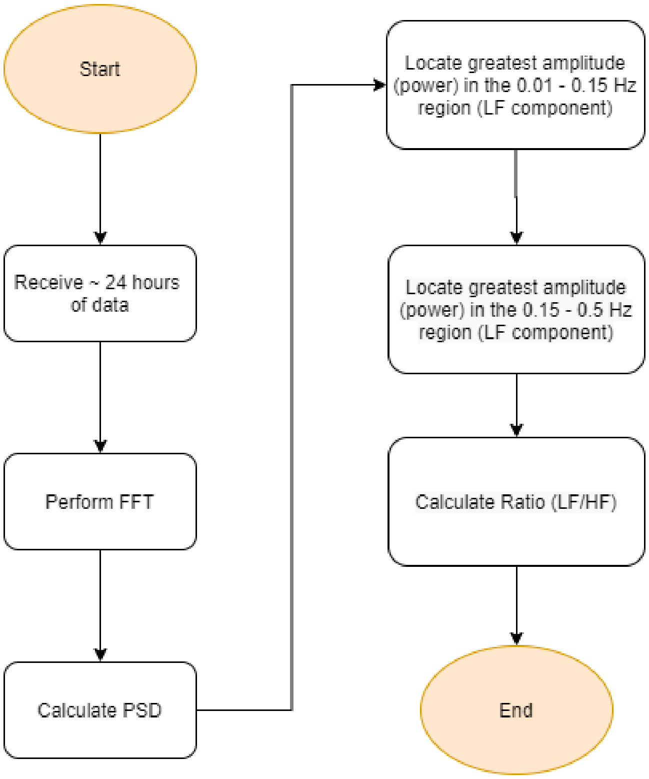

Start
Receive ~ 24 hours of data
Perform FFT
Calculate PSD
Locate greatest amplitude (power) in the 0.01 - 0.15 Hz region (LF component)
Locate greatest amplitude (power) in the 0.15 - 0.5 Hz region (LF component)
Calculate Ratio (LF/HF)
End

Figure 41: Long-term HRV extraction algorithm.

4.8. Proposed Power Consumption

This subsection discusses the potential power consumption of the device. The components of powers can be divided into three portions: the microcontroller operation, amplifier, and LED drivers. The following figure illustrates the proposed time operation for 18 minutes. This cycle of active and modem sleep modes would repeat indefinitely until the battery powering the device was discharged.

The power consumption related to the microcontroller operation takes into consideration the operating mode of the ESP32. This includes the active mode, where many of the ESP32's peripherals are active, which consists of the on-board bluetooth module. For this project's application, the active mode is required to send collected data to a mobile device. Based on our background research we expect this mode to consume roughly 60 mA while transmitting data. Due to the power-intensive nature of this mode, it is essential to switch the microcontroller's operation modes to reduce the system's current draw.

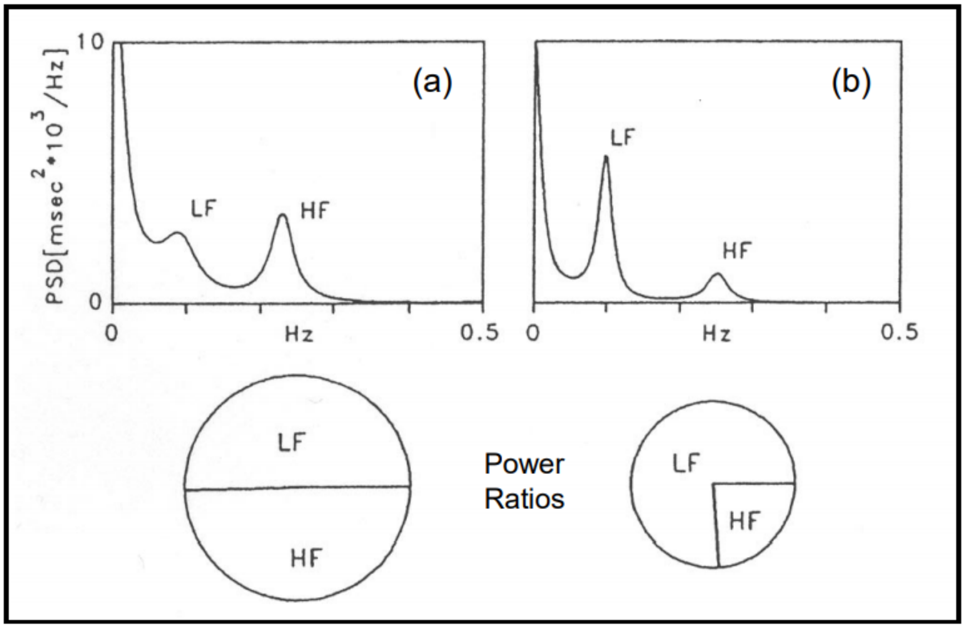
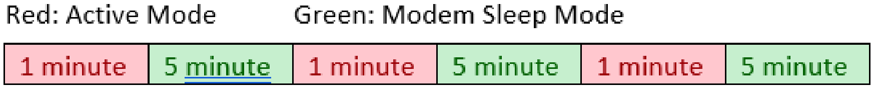


Figure 42: PSD indicating LF and HF components and their corresponding ratios.



Red: Active Mode Green: Modem Sleep Mode

🟥 1 minute	🟩 5 minute	🟥 1 minute	🟩 5 minute	🟥 1 minute	🟩 5 minute

Figure 43: Power chart of the microcontroller operations for 18 minutes.

The alternative power mode proposed is the modem sleep mode, where the radio, bluetooth, and WiFi are disabled. Figure 44 shows the difference in peripheral functionality between the two different power modes of the ESP32. It is intended to utilize modem sleep mode while collecting data and minimizing the overall power consumption rather than continuously operating in active power mode. From our background research we expect this mode to consume between 20 mA. Based on our power chart in Figure 43 we can expect our ESP32 to consume 26.66 mA per hour.

The second contributor of the power draw of the system are the amplifiers in the AFE. The following table details the current draws of various amplifiers from their respective datasheets. From this, there are tradeoffs between power and performance. With that being said, a major deciding factor is a low-current draw amplifier to reduce power consumption, which desired in portable devices.

The final power consideration we need to keep in mind is related the IR and red

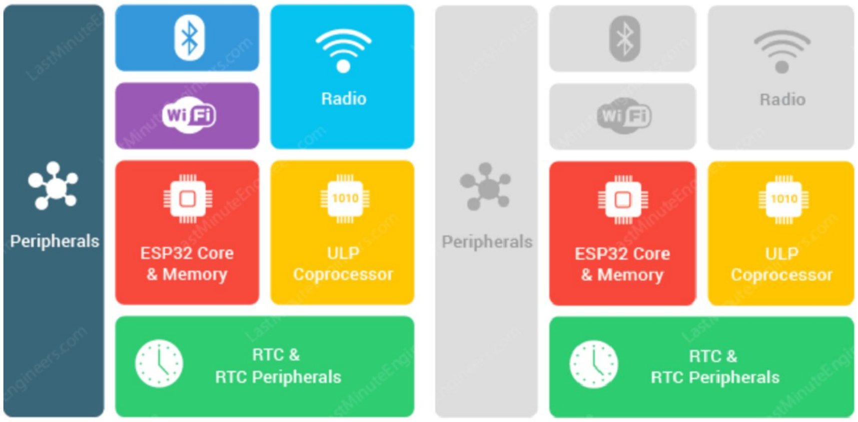


Figure 44: ESP32 Power Modes (Left:Active, Right:Modem Sleep) [66].

Table 3: Amplifier current draw comparison

Amplifier	Supply current per per amplifier	Estimated total current (4 Amplifiers)
AD00-2	1.5 mA	6 mA
LTC2064	2 µA	8 µA
LTC2067	10 µA	40 µA

LEDs. The power consumed by the LEDs is dominated by the LED power consumed to produce light and the amount of time that the LED produces light. The power to produce light is the LED current multiplied by the LED voltage. We can assume that because the LED is a diode its forward voltage does not change much over a range of currents. This means that the LED current primarily controls the LED power consumed. The time the LED is on is called the duty cycle. Using our LED driver we can control this duty cycle to control the consumed power. Ultimately, we need to turn on the red LED at a specific current, take a measurement, then turn it off, and repeat this process with the IR LED. We expect to switch both LEDs ON/OFF with a frequency of 250 Hz and a duty cycle of 50%. For our power calculations we will assume that while each LED is on it will consume its max rated current of 20 mA. In reality this current will be lower when taking measurements but we will use this as a worst case calculation. This means that we can expect the LEDs to consume 20 mA in total over the course of one hour of constant measurement. We expect the Table 4 details each device's current consumption.

By combining the expected power consumption of each device in our system we

Table 4: Power consumption of major blocks

Block	Current Consumption
AFE	40µ
ESP32	26.66 mA
LED Driver	20 mA

can calculate the total current consumption of our device to be 46.7 mA. This means that with a standard 1000 mAH battery our device would last roughly 21 hours if continuously used. A 1000 mAH battery comes in a variety of shapes and is typically 1’x2’, which is well suited for our wearable sensor.

5. Hardware Test Implementation

This section discusses the experimental tests conducted to ensure the proper functionality of each block in our high-level block diagram discussed in Chapter 3 from our proposed design in Chapter 4. The experimental results are also documented in this section.

5.1. Transimpedance Amplifier

The focus of testing the transimpedance amplifier is to ensure that the proper functionality and gain is achieved. In order to test this circuit, a current source is substituted in place of the photodiode. Based on the photodiode's datasheet, the current source would sweep in the order of 400 nA to 10µA.

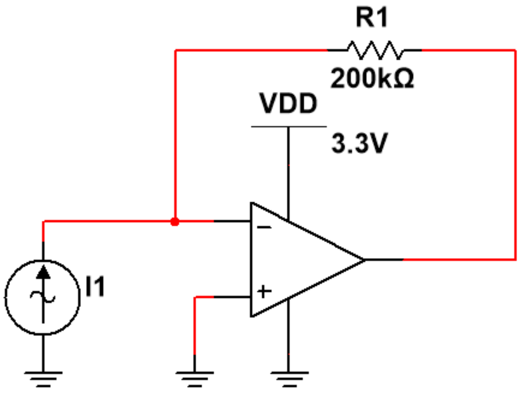


Figure 45: First stage TIA.

The theoretical gain of the circuit is 200 kΩ. The functional test is given by the following procedure:

1. Construct the TIA test circuit according to Figure 45.
2. Connect a current source (e.g., sourcemeter) into the inverting and non-inverting terminal (ground).
3. Attach the oscilloscope to measure the corresponding TIA output voltage and the varying input current source.
4. Record the source current with its respective output voltage.

The following figures demonstrate the results gathered from our testing of the transimpedance amplifier.

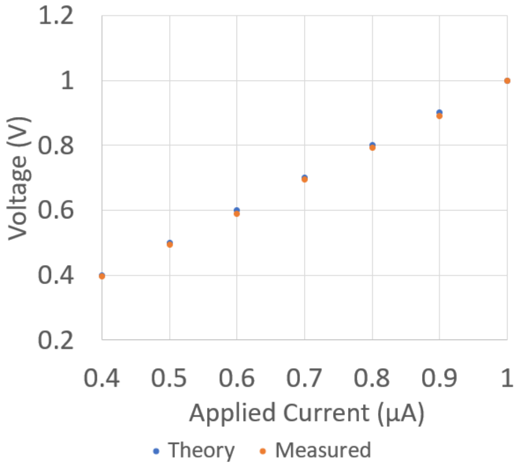


Figure 46: 400 nA - 10 µA input current and TIA output voltage.

As illustrated in Figure 46, there is a linear relationship between the injected current and the corresponding output voltage, which is consistent with the theoretical behavior described in Section 3.2. In the case with 10 µA, although the expected voltage is 10 V, the limitation of the power supply limits the output voltage to VDD.

5.2. Analog Filtering

The focus of this subsection is to compare the performance of the bandwidth filter to the LTspice simulations. The functionality test was done with an inverting amplifier with -1 gain fed to a single pole low-pass filter (15 Hz). In order to prevent potential loading issues, a voltage follower was placed between the low-pass filter and the subsequent single pole high-pass filter (0.7 Hz) to create a bandpass filter. The final gain stage is a non-inverting amplifier with a gain of 100. The implemented circuit is shown in Figure 47 with the corresponding bode plot of the entire circuit shown in Figure 48.

The test procedure is as follows:

1. Construct the test circuit according to LTspice circuit (Figure 47).
2. Connect the function generator to the inverting and the non-inverting terminal (ground).

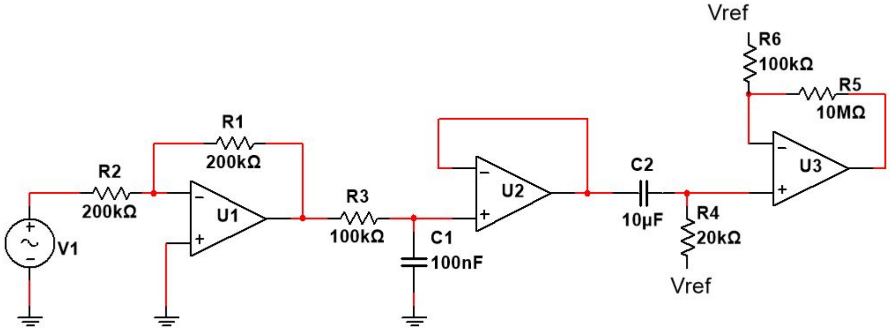


Figure 47: Analog filtering circuit.

3. Attach the oscilloscope probes to measure the incoming input signal and the output of the second-stage amplifier

4. Input a 30 mVp-p sine wave at the input terminal while sweeping the frequency from 0.1 Hz to 500 Hz (1:2:3:4:5).

The data collected from the procedure is specified in Appendix D. The corresponding plots through experimentation are shown in Figure 48. Due to the method used to measure at specific intervals, the disjointedness becomes apparent while transitioning between decades.

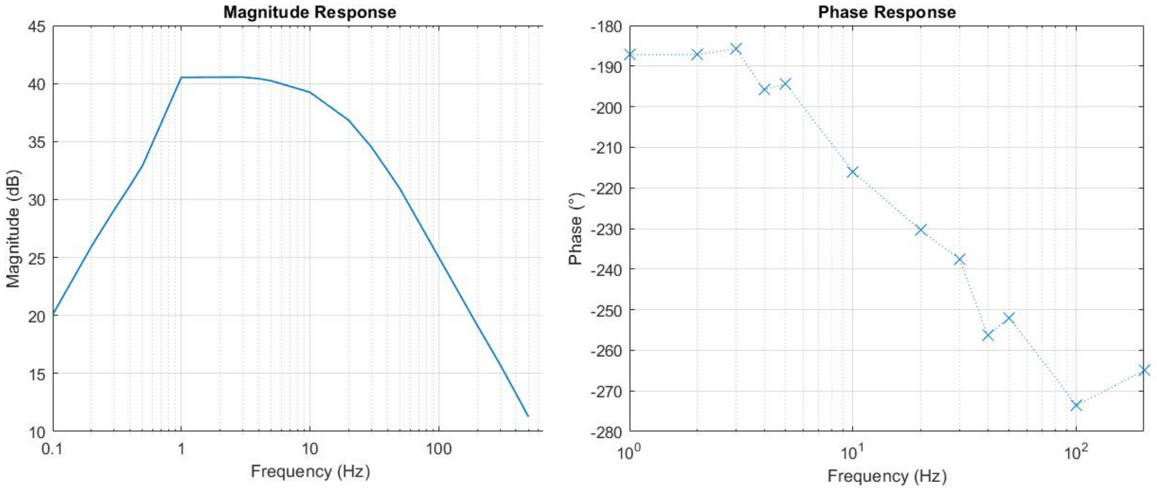


Figure 48: Filter circuit AC magnitude and phase response.

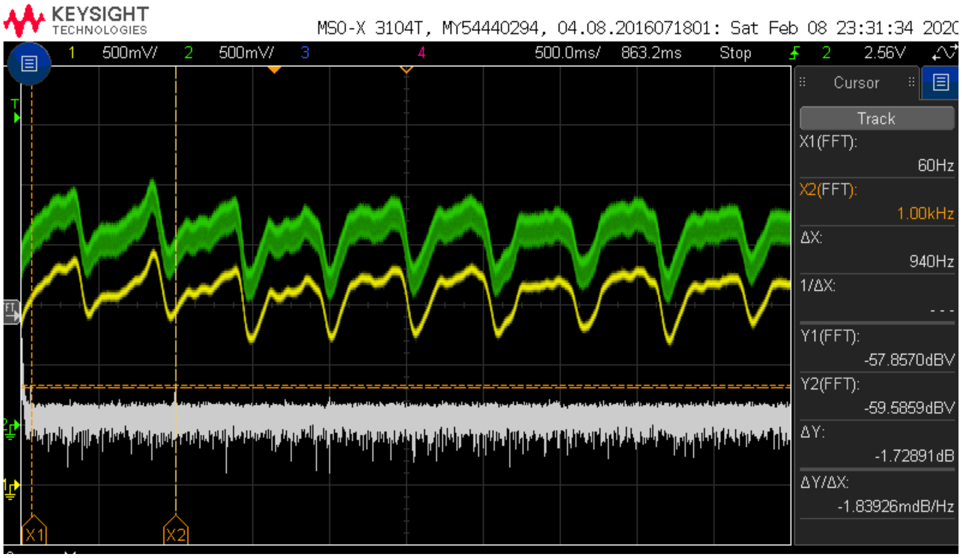
Taking this into account, the relative behavior of the magnitude response and phase response mirrors closely to the simulated filter response found in Figure 23.

5.3. AFE and EPM-4001 Photodiode

This subsection tests the complete analog front end system along with the photodiode and LED driver integration, as illustrated in the AFE circuit in Figure 21. The test procedure of the section is as follows:

1. Construct the test circuit according to LTspice circuit depicted in Figure 47.
2. Replace the U1 element of the system with the transimpedance amplifier.
3. Insert the cathode of the photodiode into the inverting terminal of U1 and the anode to the ground reference.
4. Construct the voltage reference circuit according to Figure 21.

The following figure represents the corresponding red LED AC pulsatile PPG waveform.



MSO-X 3104T, MY54440294, 04.08.2016071801: Sat Feb 08 23:31:34 2020

1 500mV/ 2 500mV/ 3 4 500.0ms/ 863.2ms Stop 2 2.56V

X1(FFT): 60Hz

X2(FFT): 1.00kHz

ΔX: 940Hz

1/ΔX: - - -

Y1(FFT): -57.8570dBV

Y2(FFT): -59.5859dBV

ΔY: -1.72891dB

ΔY/ΔX: -1.83926mdB/Hz

Figure 49: AC pulsatile of PPG signal with FFT overlay.

Based on the figure, the green waveform represents the raw AC signal of the waveform generated at the second stage of the amplifier. The yellow waveform is the same signal with an additional 15 Hz analog low pass filter, which significantly reduced the noise observed in the green waveform. The approximate amplitude of the PPG signal is 700 mV while riding on a mid-supply voltage reference (1.65 V). Along with the waveforms, an FFT operation was performed and overlayed, as indicated in the white graph, which suggests much of the signal's power is near low frequencies. Furthermore, there are peaks where there is an indication of LED switching noise coupling in the PPG signal. For instance, cursor X2 from Figure 49 on the FFT graph shows a peak at 1 kHz (x-position indicated in the cursor information on the right).

5.4. Photodiode

Testing the photodiode as a separate block was difficult because of the low expected output currents. Therefore, we first be characterized a transimpedance amplifier with a feedback resistor of 200 kΩ. The photodiode was then placed at the input of the amplifier and build a small circuit to drive an LED with its max operating voltage and current. Both the photodiode set-up and LED circuit will go into a light-free test chamber. Here we will shine the LED on the photodiode. Because we know the max operating irradiance from the LED, we can calculate the expected current from the photodiode. We will confirm if the output of the transimpedance amplifier agrees with our expected photodiode current. The procedure is listed below

1. Build the transimpedance amplifier in the figure below:

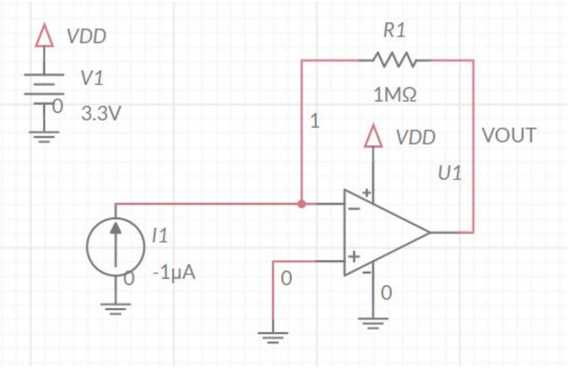


Figure 50: Current-supply testbench circuit.

2. Connect a current source to the input of the amplifier and set the current to be -1 µA.

3. Connect an oscilloscope to the output of the amplifier.
4. Turn on the power supply for the amplifier and the current input signal.
5. Divide the output voltage by the input current to get the actual R_f value.
6. Turn all the supplies and sources off and replace the current source with the photodiode (See below).

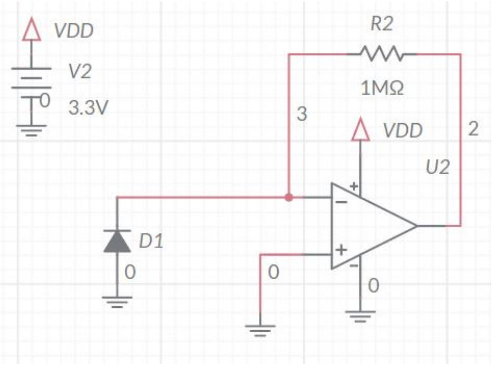


Figure 51: Photodiode testbench circuit.

7. Place the testing setup in an environment with as low ambient light as possible.
8. Create an LED circuit to drive a red or IR LED at its maximum operating range. Check their datasheets to calculate their max irradiance.
9. Power the LED circuit and place it inside the light-free chamber so that the LED is directly facing the photodiode.
10. Measure the output voltage of the amplifier and divide the measured voltage by the gain that was noted in step 5.

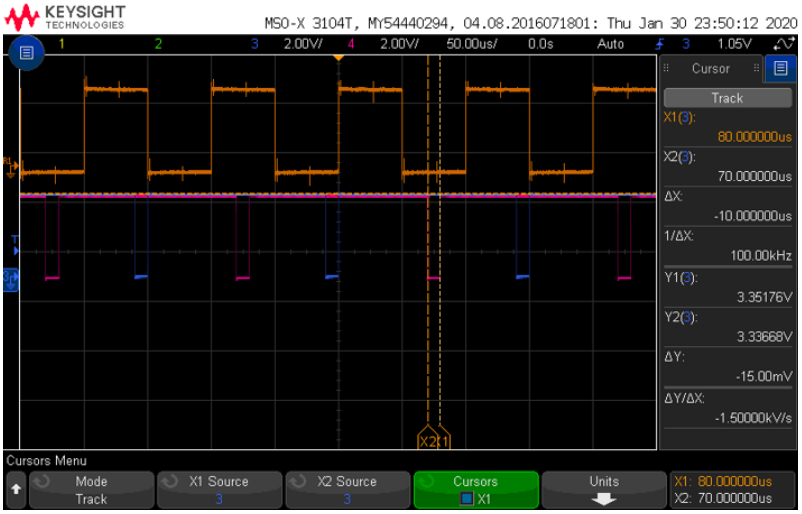
5.5. LED Driver and LEDs

The focus of this functional test is on the LED drivers, in order to vary the light intensity of the LED and, in turn, to control the incoming input DC of the signal. The first step of the functional test is verifying the LED characteristics with a simple voltage, current-limiting resistor, and required LEDs. The following step is constructing the LED drivers using NPN BJTs in conjunction with the desired LEDs. Figure 20 presents the complete test circuit of the LEDs and their respective drivers.

The procedure for this test is as follows:

1. Build and verify the behavior of the red and IR LEDs.
2. Assemble one portion of an LED driver with NPN and resistors.
3. Use a voltage power supply at the base of the transistor to vary the base current and observe any changes with light intensity of the LED (measure current).
4. Repeat this for the IR LED.
5. Utilize a DAC in place of a voltage power supply to vary the light intensities of the LEDs.
6. Implement a switching mechanism to alternately turn on the LEDs.

Figure 52 below shows the LED driver switching the red and IR LEDs ON/OFF. The orange waveform above represents a 10 kHz square wave form which serves as a reference waveform to control the LED control signal. In this case, the PMOS switches turn on an active low, therefore during one period the red LED turns on for 10 µs likewise for IR LED. This amounts to an on-time of 20 µs and equivalent to a 20% duty cycle.



MSO-X 3104T, MY54440294, 04.08.2016071801: Thu Jan 30 23:50:12 2020

1 2 3 2.00V/ 4 2.00V/ 50.00us/ 0.0s Auto 3 1.05V

Cursor

Track

X1(3): 80.000000us

X2(3): 70.000000us

ΔX: -10.000000us

1/ΔX: 100.00kHz

Y1(3): 3.35176V

Y2(3): 3.33668V

ΔY: -15.00mV

ΔY/ΔX: -1.50000kV/s

Cursors Menu

Mode: Track

X1 Source: 3

X2 Source: 3

Cursors: X1

X1: 80.000000us
X2: 70.000000us

Figure 52: Probing various nodes of the LED driver.

5.6. Low Dropout Voltage Regulator

The focus of this functional test is to ensure that the proper output voltage is given from the selected LT1763 LDO micropower regulator.

5.6.1. Line Regulation

The first functional test looks at how varying the input voltage to the regulator affects the output. In this scenario, there is only a 3.3 µF capacitor at the output under the LDO's datasheet recommendation. In this scenario, the input voltage was varied from 3.7 V to 4.2V. 4.2 V supply was in reference to a fully charged LiPo battery. Likewise, the identical test was performed with a 3.7 V supply in order to simulated a nearly discharged battery. The following figure illustrates the output voltages given for these two different input parameters.

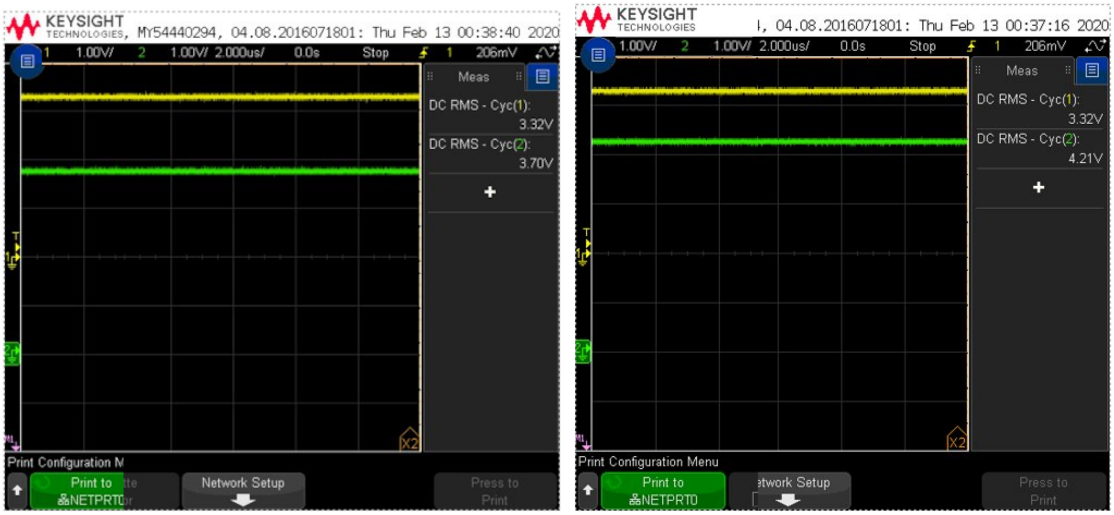
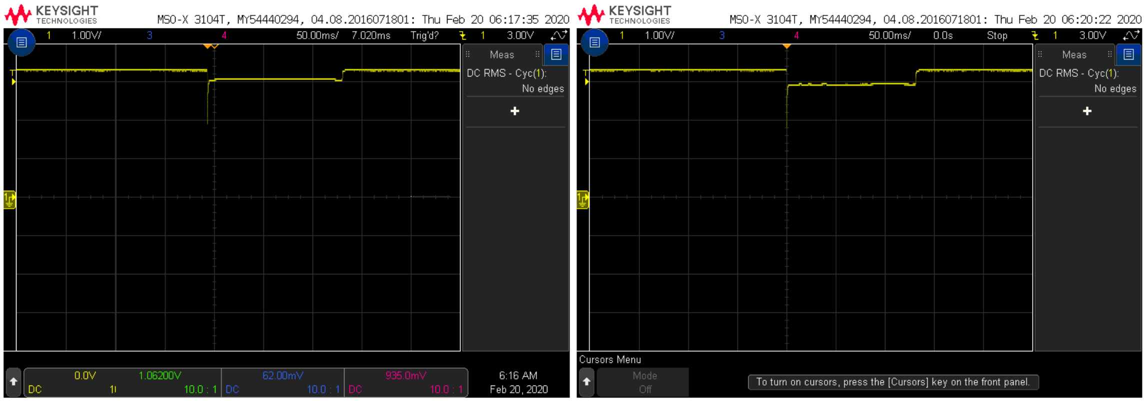

KEYSIGHT TECHNOLOGIES, MY54440294, 04.08.2016071801: Thu Feb 13 00:38:40 2020
1 1.00V/ 2 1.00V/ 2.000us/ 0.0s Stop 1 206mV
Meas
DC RMS - Cyc(1): 3.32V
DC RMS - Cyc(2): 3.70V
Print Configuration M
Print to NETPRT0
Network Setup
Press to Print
KEYSIGHT TECHNOLOGIES I, 04.08.2016071801: Thu Feb 13 00:37:16 2020
1.00V/ 2 1.00V/ 2.000us/ 0.0s Stop 1 206mV
Meas
DC RMS - Cyc(1): 3.32V
DC RMS - Cyc(2): 4.21V
Print Configuration Menu
Print to NETPRT0
etwork Setup
Press to Print

Figure 53: LDO response: 4.2 V input voltage on the left, 3.7 V input voltage on the right.

From these figures, it can be seen that the regulator successfully maintains a constant 3.3 V supply necessary for the pulse oximeter design.

5.6.2. Load Regulation

The next test looks at load regulation with the LDO, specifically to test regulator's ability to source a 250 mA current to the connected load. To do this we used a momentary switch attached to a resistor connected to ground. The other connection of the switch was connected to the output of the LDO. We set up the scope to trigger on the falling edge of the LDO's ouput voltage. The resulting scope pictures are in the figures below:



To turn on cursors, press the [Cursors] key on the front panel.

Figure 54: LDO response; 4.2 V load regulation response, 3.7 V load regulation response.

Similar conclusions can be drawn from these tests, which show the the output of the LDO drop abruptly when the momentary switch was pushed and then recover while the switch is pressed. Similar to the line regulation test we used both the expected fully charged, and fully discharged battery voltage of a LiPo battery. This illustrates the robustness of the regulator.

5.7. Breadboard, Printed Circuit Board, and Flexible Printed Circuit Board

There were three iterations of the pulse oximetry system. The first iteration was constructed on a breadboard to ensure the proper functionality of the system block diagrams outlined in Chapter 3. With the first iteration, all the necessary components were soldered onto a breakout board that was then placed on a breadboard.

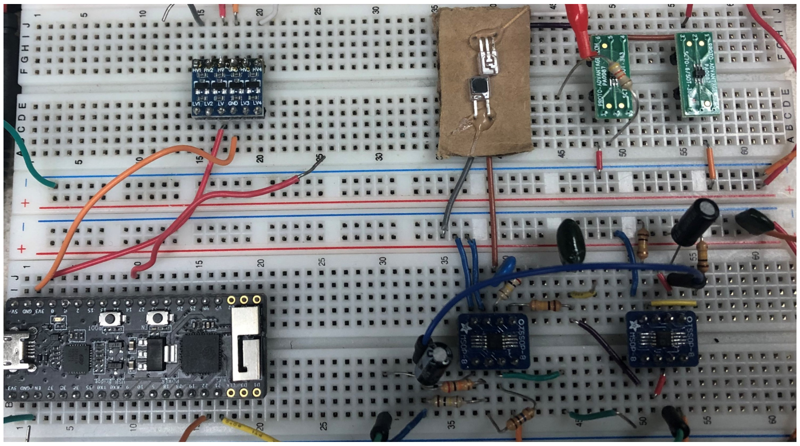


Figure 55: Breadboard implementation.

The circuitry in the top right (represented in the green breakout boards) constitutes the circuits involved with the LED drivers that control the LEDs to the left of the drivers. The photodiode's output is connected to one of four ADA4500 amplifiers in the bottom right, which makes up the AFE of the proposed pulse oximeter device. The bottom left is the microcontroller reading from the output of the amplifier while also controlling LED timing.

The second iteration of our circuit is the rigid PCB as illustrated in Figure 56and focuses on transferring the entire system to a smaller form factor. Many of the chosen parts were surface mount components, which reduced the overall size of the system and weight. In addition, they increased the portability of the pulse oximeter. Furthermore the second iteration had LTC2067 amplifiers to reduce the overall power consumption of the system.

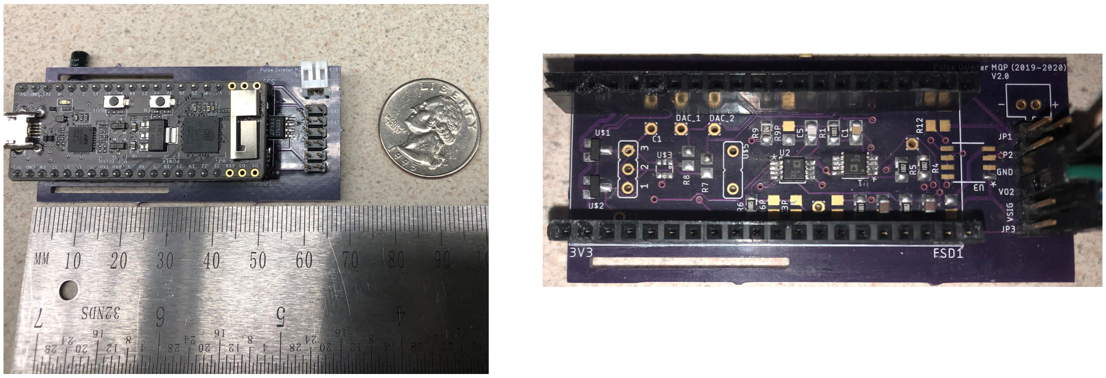


Figure 56: Rigid printed circuit board implementation top view: microcontroller on the left, without the microcontroller on the right.

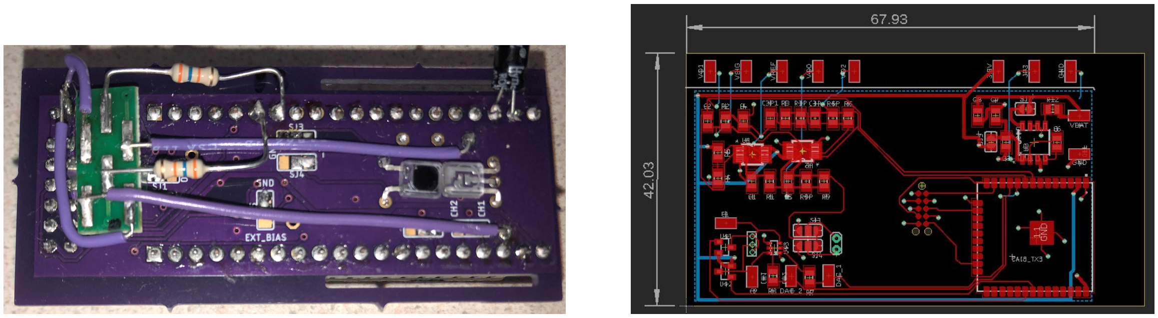


Figure 57: Bottom: Rigid printed circuit board implementation on the left, Proposed flexible PCB layout on the right.

The rigid PCB significantly reduced the overall size of the system from the original 17820 mm^2 to 2010 mm^2. This was an 88.7% reduction in size, which allowed for portability. The rigid PCB was designed in such a way that the ESP32 could be installed in a similar manner as the shields for Arduino, as shown in Figure 56. Conversely, the LEDs and photodiodes were placed on the underside of the PCB, as depicted in Figure 57 (left). The additional wiring underneath the board is due to a layout error with an incorrect footprint for our BJT package. A makeshift BJT breakout board was implemented to ensure proper operation. Lastly, Figure 56 highlights the remaining circuitry, such as the AFE, power management, and LED drivers of the system. Additional information about the rigid PCB schematic and top layer view is shown in Appendix E. Due to unforeseen circumstances, a flexible PCB variant was not created. However, the purpose of this final iteration is to achieve flexibility and portability where it is intended to have the device conform at the placement site. Figure 57 (right) is the proposed layout for the flexible system. The sized increased to 2858 mm^2 placed on the same plane as the rest of the circuit.

6. System Integration and Data Analysis

This section discusses the software involved with the pulse oximeter, which includes data extraction and analysis, in order to return the appropriate vital signature. This includes the various software function tests to document the results of the algorithms. The first section details the procedure for the system level tests with human testing and provides the corresponding results. The section closes with the data analysis of the results gather from human testing and comparison of other devices to validate the wearable sensor's performance.

6.1. Human Test Implementation

This section discusses the human testing to document and validate the performance of the flexible, wearable pulse oximeter. This validation procedure was achieved with the Rad-8 Masimo Pulse Oximeter indicated in Figure 58. The test examined the test subject under normal blood oxygen saturation levels (future works could potential look into artificially reducing the blood oxygen saturation to test under low ranges of SpO_2). The Masimo pulse oximeter is able to measure SpO_2 from 1 - 100%, pulse rate from 25 - 240, and finally a perfusion index, is the ratio of the pulsatile blood flow to the non-pulsatile or static blood in peripheral tissue, in the range of 0.02 - 20%.

The following Figure 58 demonstrates the process to characterize the performance of our device. The measurement was done with both devices collecting data simultaneously.

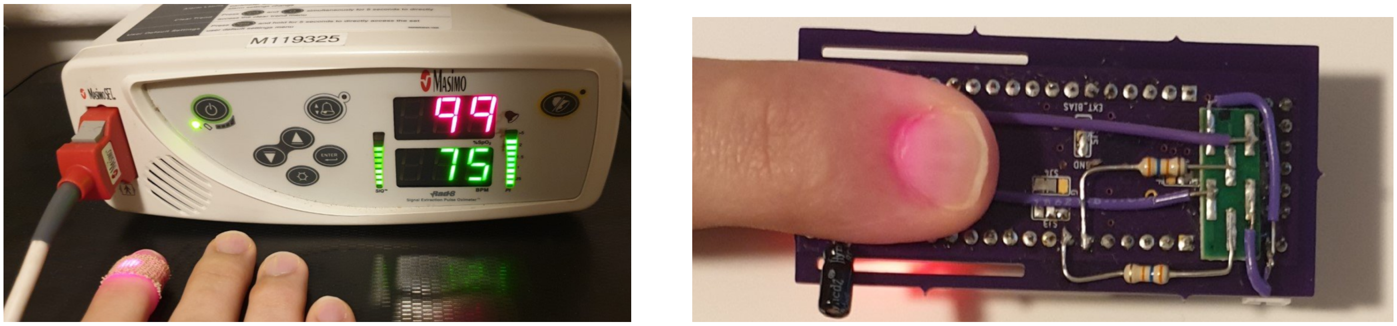


Figure 58: Masimo Pulse Oximeter setup on the left, MQP device setup on the right.

6.2. Digital Signal Processing

Throughout the testing of the device, the output of the ADC was plotted to visualize the data coming from the analog front-end circuitry. The signal contained a lot of noise and a few different filters were implemented to remove unwanted signals. The first digital filter used was a 2$^{\text{nd}}$ order butterworth filter that had been z-transformed from the continuous to the discrete domain. The purpose of this approach was to create a filter in the $j\omega$ plane, convert the filter into the discrete domain, and extract the coefficients of the discrete filter to use them to implement the filter on the microcontroller. The output of this filter for just the IR PPG data is in Figure 59 below:

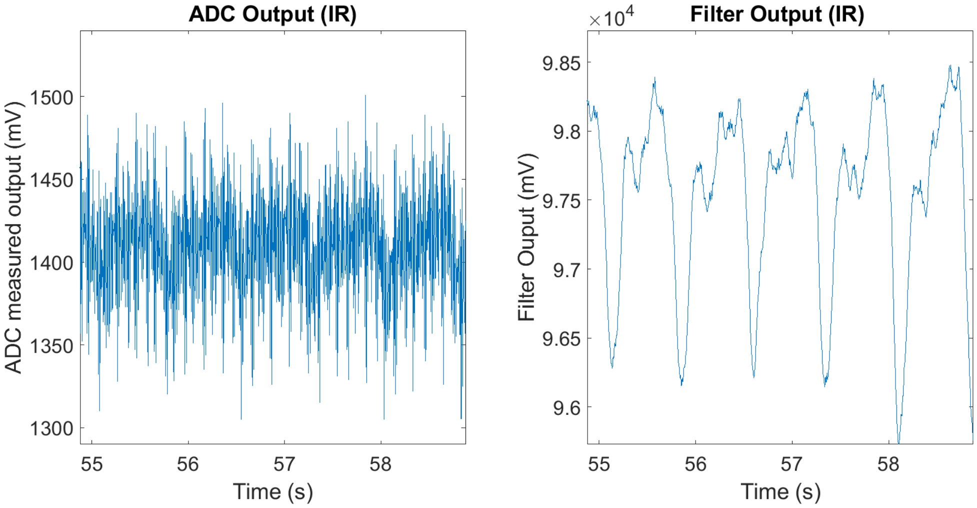


Figure 59: 2nd order butterworth low-pass operation at 5Hz for red LED reading.

In Figure 59 it can be seen that there are high-frequency artifacts in the signal, which requires additional filtering. This is resolved by increasing the order of the filter, as described in Section 3.3. Eventually, a 6^{th} order lowpass Butterworth filter was implemented using built-in Matlab functions.

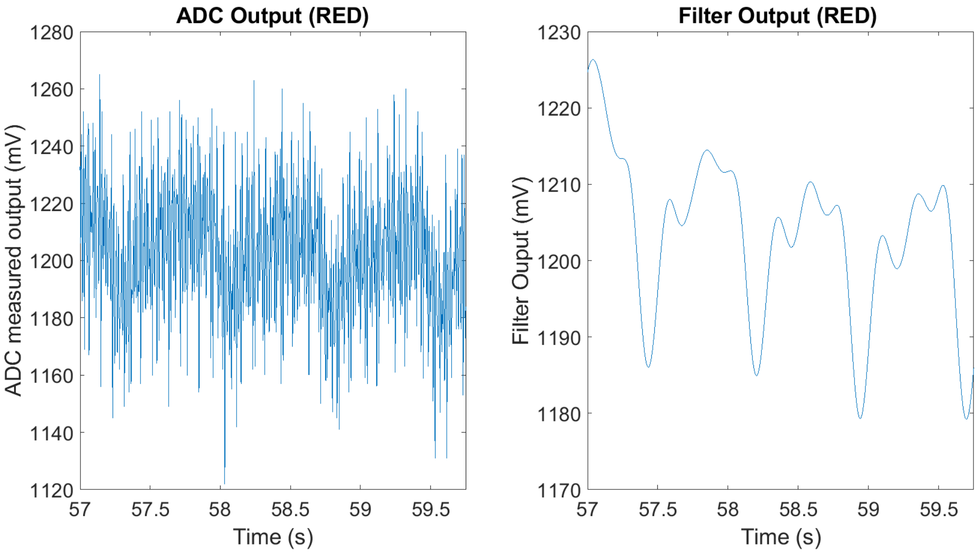


Figure 60: 6th order low-pass butterworth (5 Hz) (red)

You can see from Figures 60 and 61 the output of the filter is much better compared to the raw ADC readings or our original 2nd order filter. We hoped to implement the digital filter on the phone application using the filter coefficients extracted from the Matlab filter. Unfortunately, when we created the filter in the app using a difference equation causes the instability in he filter and the data became unusable. Because of this instability issue, we resorted to importing the open-source library "An efficient IIR filter library written in JAVA" and using the filter's implementation. After this was done, the filter's output was verified to be stable and gave us proper values.

During the digital filter implementation and testing, the importance of reducing 60 Hz light from the photodiode became apparent. Performing the FFT on the raw ADC data showed that the 60 Hz noise was an order of magnitude larger than the signals in the desired frequency range. To mitigate this issue, we placed our device in an environment with little to no ambient light to reduce the 60 Hz noise. With the filtering implemented and the reduction of ambient light while testing our device is able to sample the user's PPG waveform reliably. The next sections will detail the results of extracting the vital sign measurements.

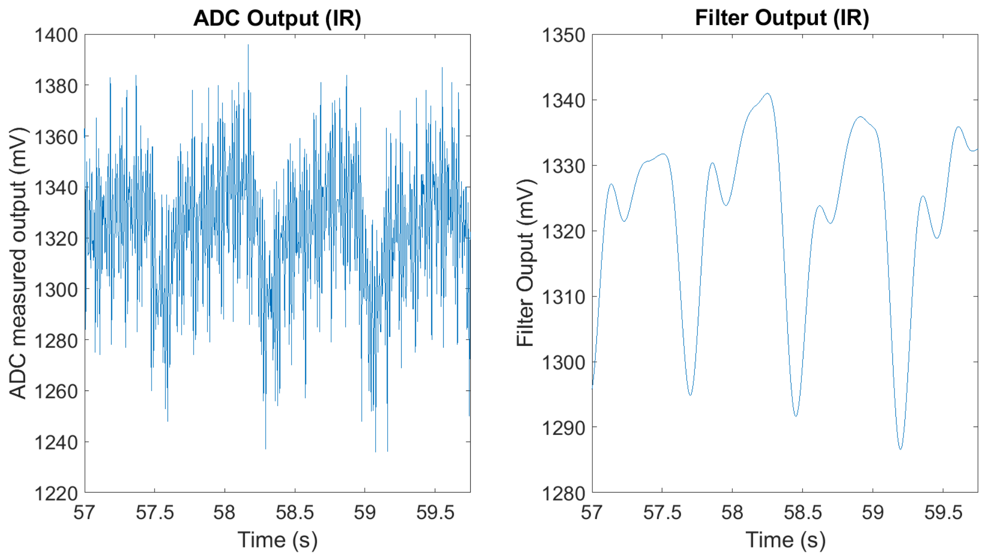


Figure 61: 6th Order Low-Pass Butterworth (5 Hz) (IR)

6.3. SpO_2 and Heart Rate

One of the project focal points is the extraction of SpO_2 and heart rate values from the corresponding PPG waveform. The values generated by the algorithm are displayed for the patient through the mobile app as well as the Firebase database. The results from Section 6.1 are shown in the following Table 5 and Figure 62.

Table 5: Numerical results for SpO_2 and Heart Rate

	Masimo Rad-8 Pulse Oximeter [67]		This work		Percent Error (%)	
Test #	SpO_2 (%)	HR (bpm)	SpO_2 (%)	HR (bpm)	SpO_2 (%)	HR (bpm)
Test 1	97.96	68.38	97.92	68.66	-0.03	0.41
Test 2	99.78	76.08	100	76.90	0.21	1.07
Test 3	97.21	83.67	97.65	82.39	0.45	-1.52
Test 4	95.46	73.22	94.5	76.90	-1.00	5.024

The chart above indicates that four sets of measurements were performed in the evaluation. A key consideration regarding these results is that our device provides one SpO_2 value for one test, (roughly 2 minute period of data collection). Conversely, values from the commercial device were averaged over that same period. From Table 5, a comparison graph is also generated to visualize how our device performed relative to the commercial-grade pulse oximeter, as illustrated in Figure 62.

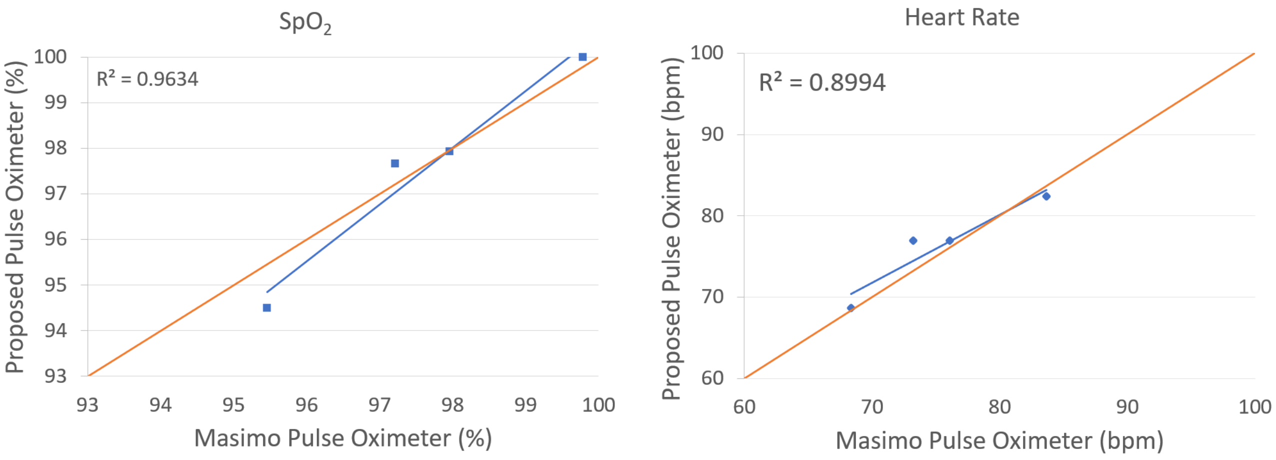


Figure 62: SpO$_2$ (left) and heart rate (right) comparison chart between Masimo and MQP pulse oximeter.

The x-axis plots the SpO$_2$ or heart rate measurement calculated by the Masimo device [67] with our device's results plotted along the y-axis. Ideally, we should expect a one-to-one matching between these two devices, as indicated by the orange line. Realistically, there are variations between the two. The relationship between the two can be expressed by a statistical merit, coefficient of determination, or R^2 as given by the following equation

$$R^2 = 1 - \frac{SS_{regression}}{SS_{total}}, \tag{19}$$

where SS$_{regression}$ is the sum of squares due to regression (explained sum of squares) and SS$_{total}$ is the total sum of squares [68]. This value ranges from 0 to 1 and provides insight on how well the data fit with a regression model, with 1 indicating an excellent fit to the model. From Figure 62 and Table 5, the values of SpO$_2$ and heart rate are consistent with the nominal adult range for SpO$_2$ and heart rate. Furthermore, the calculated R^2 is 0.9634 for SpO$_2$ and .8994 for the heart rate measurements.

The app displays with the corresponding measurements are demonstrated in Figure 63. The x-axis above represents the 24-hour clock, which shows four points (measurement) over time. Similarly, the y-axis contains their respective measurements for SpO$_2$ and heart rate.

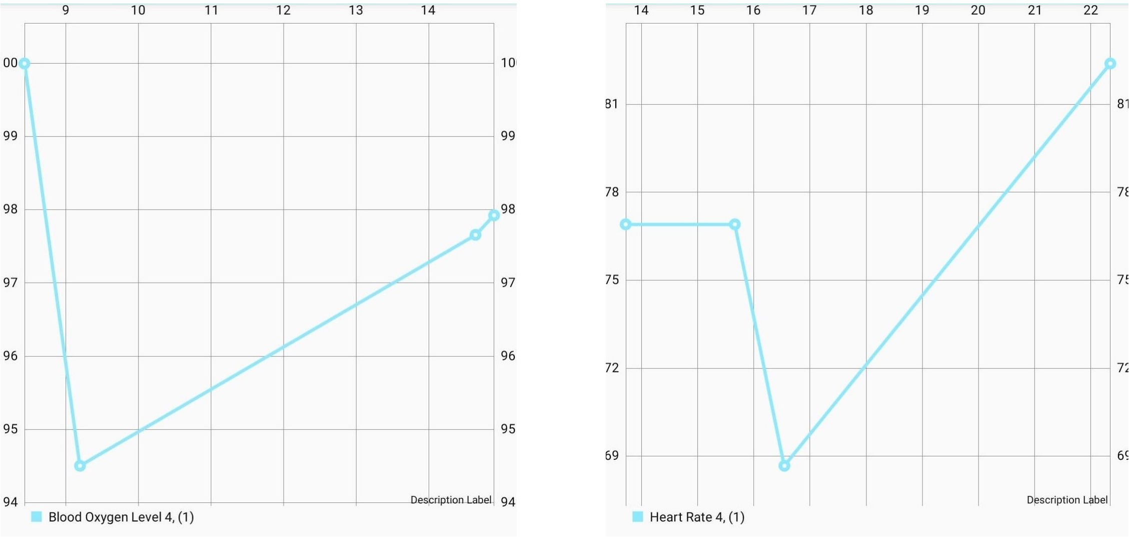


Figure 63: SpO_2 (left) and heart rate (right) app display.

6.4. Heart Rate Variability

We gathered the expected values for the different HRV indices from previously published papers[26],[69]. Table 6 details the expected values of interest to us from these papers (Note: N/R stands for 'Not Reported').

Table 6: Expected HRV values from other works.

Author and Date	Number of Participants	SDNN (ms)	COV	SDSD	RMSSD (ms)	NN50	pNN50 (%)
Pikkujamsa (2001) [69]	392	13-168	N/R	N/R	N/R	N/R	N/R
Johnston (2006) [26]	36	20-120	.04-.12	20-100	20-80	80-120	15-30

We can then compare the expected HRV values from Table 6 to our gathered results in Table 7. In this way we can compare how well our device extracts the necessary information. It can be seen from comparing our extracted values to the expected values, that our measurements seem a bit skewed in the positive direction. Measurements for COV, SDSD, RMSSD, and pNN50 are all greater than their expected counterparts. This is most likely due to our peak detection algorithm. It can be seen from the RMSSD value that the error found from comparing successive NN intervals is greater than twice the expected value. This means our algorithms

Table 7: Numerical SHRV values.

	HRV Indices					
Test #	SDNN	COV	SDSD	RMSSD	NN50	pNN50 (%)
1	145.795	0.2526	127.612	208.473	89	80
2	148.204	0.25877	134.682	215.112	92	82
3	157.765	0.2627	145.81	228.87	89	83
4	158.5958	0.26372	139.015	221.3869	87	81

may be having a hard time finding the peaks of our signal and over calculating the time between peaks. This conclusion is further supported by the fact that the pNN50 measurement is greater than the expected by roughly 50%. This means that the proportion of NN intervals that are greater than 50 ms is approximately 80% rather than the expected 30%. Improvements can be made for all these measurements by improving the input signal for the ADC, optimizing our digital filtering, and testing more robust peak detection algorithms. Figures 64, 65, and 66 display the phone application graphs for these various indices that the user of the wearable sensor would see.

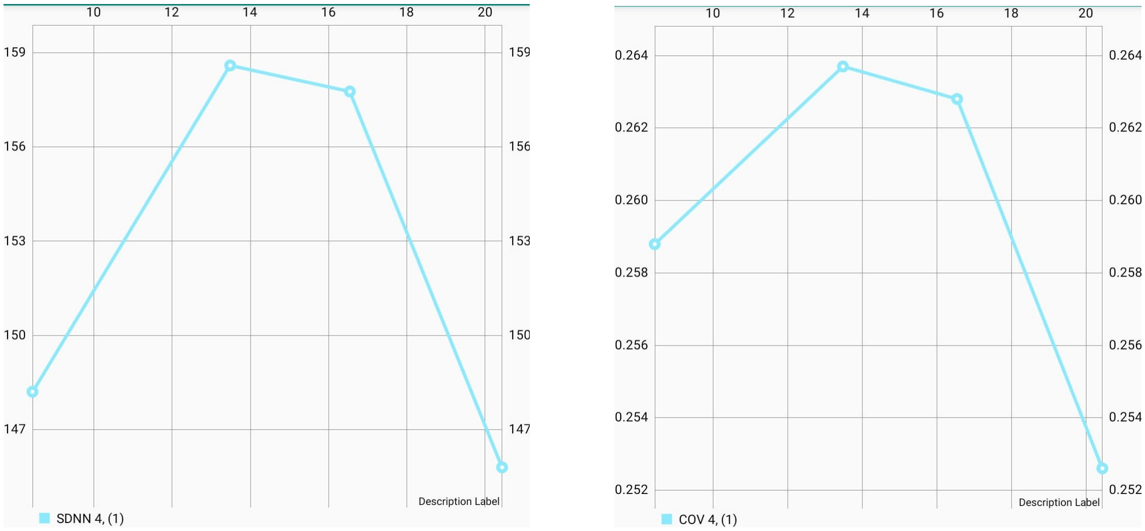


Figure 64: App display: SDNN (left) and COV (right).

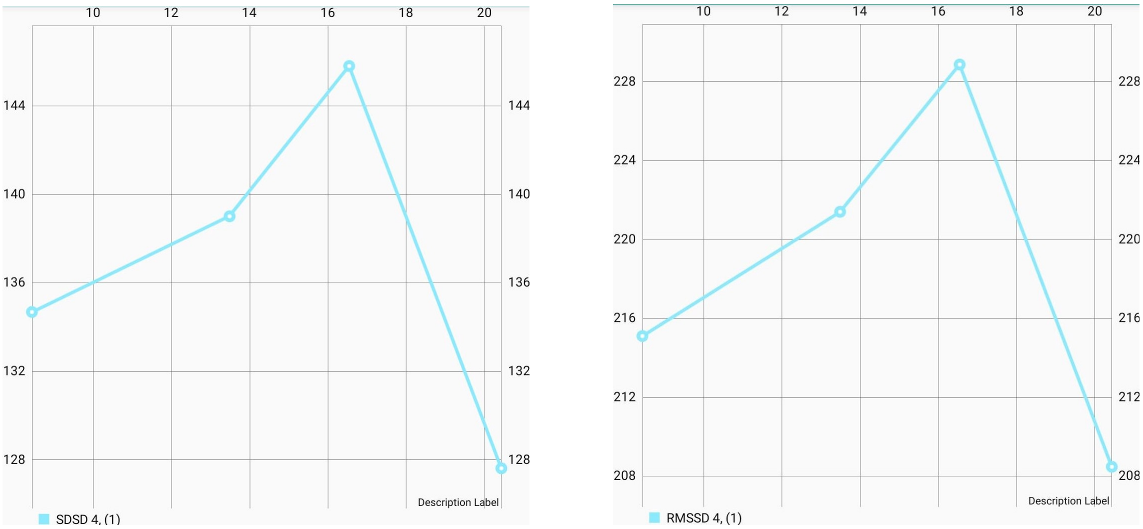


Figure 65: App display: SDSD (left) and RMSSD (right).

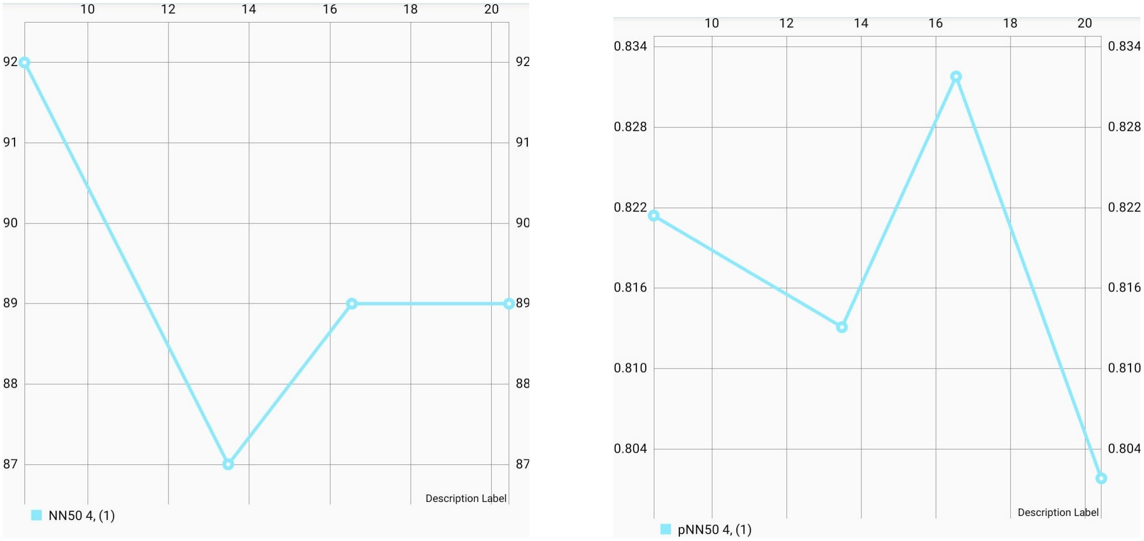


Figure 66: App display: NN50 (left) and pNN50 (right).

7. Recommendations

In this section, we detail the various improvements to the pulse oximeter device. An important consideration would be to optimize the power consumption of the device. Based on the initial power calculations detailed in Section 4.8, the two power-intensive blocks are the LED drivers and the microcontrollers. In terms of the LED drivers, the duty cycle can be modified such that the LEDs are not on 50% of the time, thereby effectively reducing the LED average current. With the current design, the pulse oximeter device communicates in a single direction, as shown in Figure 67a. Improvements can be made to implement a bi-directional communication between the mobile device and the pulse oximeter, as shown in Figure 67b. This permits flexibility for the end-user to adjust the device according to their needs.

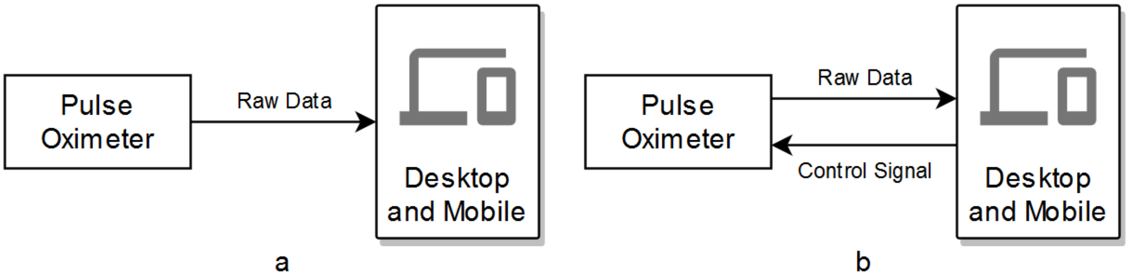


Figure 67: Device communication scheme.

Another significant block to consider in terms of power consumption is the microcontroller. This is especially the case when the microcontroller is transmitting data from the portable device to a mobile device. Possible future work could focus on reducing this power consumption to improve the sensor's lifetime on a battery. The microcontroller was operating at a full clock speed of 80 MHz. This clock could be reduced to reduce power while making sure that overall functionality is maintained.

The initial project looks at the engineering design process to develop a pulse oximeter monitoring device. Therefore, another recommendation is designing the sensor to ensure proper compliance with any potential codes or standards for medical devices. Some of the possible codes to design are the following:

- IEEE 11073, which is a family of standards related to medical and health device communication standards, focuses explicitly on the "interoperability in personal health devices by allowing the secure exchange of information among the device, patient, and physician." The specific standard that might be applicable for our device is 11073-10404-2020, which is the standard that defines "communication between personal telehealth pulse oximeter devices and computes engines."

- Another code to consider is the Ingress Protection code (IPX code), which is concerned with whether the device is designed with solid particle protection, liquid ingress protection, and mechanical impact resistance to properly classify the environments that the device tolerates.
- A final consideration is the Food and Drug Administration (FDA), an agency that "monitors the ongoing safety and efficacy of regulated marketed devices" [70]. For medical devices, the FDA classifies devices under three classes: "Class I includes devices with the lowest risk, and Class III includes those with the greatest risk." [71]. Adhering to these considerations will ensure the proper compliance of the wearable medical monitoring devices to mitigate potential risks for the device consumers.

When looking at the output of our digital filters, one can see that the total voltage swing of the PPG waveform is no greater than 50 mV peak to peak. One possible solution is to replace one of our analog filters with an active filter stage. This would accomplish filtering the signal while also improving the swing of the desired PPG waveform. On a similar note, doing the FFT on our raw ADC data, we noticed that the 60 Hz component of the data was extremely large due to the ambient light. Future designs should implement either a digital or analog notch filter at the 60 Hz frequency to improve the overall signal integrity.

It is also important to emphasize that the ADC sampling frequency should be twice the frequency of the highest signal seen at the ADC; this includes the highest possible frequency of the noise. This means if an application's low pass filter has a -20 dB roll-off after the corner frequency, depending on the resolution of the ADC, the frequencies between the corner and the -40dB point of the filter will most likely be sampled by the ADC. If the ADC is not sampling at a high enough frequency, aliasing will occur between the noise frequencies and the desired signal frequencies, leading to useless data. For this reason, future designs absolutely must incorporate higher-order low-pass filters to reduce the unwanted frequencies seen by the ADC and therefore reduce the overall sampling frequency.

Another recommendation is the implementation of multispectral based pulse oximetry. For the conventional red and IR LEDs, which have a longer wavelength of the spectra "allowing for deeper tissue penetration," the issue arises with coupled motion artifacts. In contrast, the alternative LEDs such as blue and green are relatively invariant to these artifacts. The usage of alternative LEDs can present a method to improve PPG signal quality.

Regarding the algorithms, one recommendation would be to perform the analysis for heart rate in the time-domain and find the periodicity by using autocorrelation. Detecting heart rate by performing analysis in the frequency domain is a reasonable estimate and yielded satisfactory results. However, an optimized analysis in the time domain could potentially offer even higher accuracy. Furthermore, for short-term HRV our results indicated that our peak detection algorithm was not locating all the systolic and/or diastolic peaks, which led to slightly greater NN intervals than expected. An optimized peak detection algorithm would help calculate heart rate, but it would also assist with short-term HRV measurements.

A possible future direction of work is implementing the algorithm relating to arterial stiffness in terms of hardware-related design. Implementation with a flexible PCB version can further reduce the weight of the sensor. Additionally, more tests should be performed to validate the performance of outside the nominal ranges for each vital sign. This would involve additional IRB authorization to conduct these special circumstances safely. Lastly, future teams should look at the system on chip (SoC) to create a miniaturized pulse oximeter system.

8. Conclusion

The goal of this project was to design a flexible wearable sensor using the engineering design process. The wearable sensor was successfully implemented with the hardware and part of the algorithms. With our device, we can extract SpO_2, heart rate, short-term heart rate variability. However, due to the time constraints of the project, only a rigid PCB implementation was developed. However, the transition from a breadboard prototype to a PCB significantly improved the portability of the device. An improvement from the rigid PCB would consider a flexible PCB variant to better adjust to the measurement site.

Acknowledgment

We would like to thank our MQP advisor Professor Ulkuhan Guler for her guidance and patience. Furthermore, we would like to thank Professor Yitzhak Mendelson for his valuable input regarding the algorithms. We would also like to express our gratitude towards Ian Costanzo and Devdip Sen for their constructive feedback throughout this project. Finally, we would like to thank our friends and family for their continuous support and patience.

References

[1] R. Ramani", “Remote monitoring telemedicine,” 2020. [Online]. Available: https://www.upmc.com/healthcare-professionals/physicians/telemedicine/services/remote-monitoring

[2] A. H. Association", “Using remote patient monitoring technologies for better cardiovascular disease outcomes guidance,” 2019. [Online]. Available: https://www.heart.org/-/media/files/about-us/policy-research/policy-positions/clinical-care/remote-patient-monitoring-guidance-2019.pdf?la=en&hash=A98793D5A043AB9940424B8FB91D2E8D5A5B6BEB

[3] M. Clinic", “Hypoxemia,” 2018. [Online]. Available: https://www.mayoclinic.org/symptoms/hypoxemia/basics/definition/sym-20050930

[4] “Using pulse oximeters,” 2011. [Online]. Available: https://www.who.int/patientsafety/safesurgery/pulse_oximetry/who_ps_pulse_oxymetry_tutorial1_the_basics_en.pdf

[5] E. Engl and D. Attwell, “Nonsignalling energy use in the brain,” *The Journal of Physiology*, vol. 593, no. 16, pp. 3417–3429, Aug 15, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.2014.282517

[6] “Mld support economic model,” 10-14 2015. [Online]. Available: http://gpontin.com/2015/10/mld-support-economic-model/

[7] D. F. Treacher and R. M. Leach, “Oxygen transport 1 basic principle,” pp. 1302–1306, -11-7 1998. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1114207/

[8] E. A. Huebner and S. M. Strittmatter, “Axon regeneration in the peripheral and central nervous systems,” *Results and problems in cell differentiation*, vol. 48, pp. 339–351, 2009. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/19582408

[9] R. Myatt, “Pulse oximetry: what the nurse needs to know,” *Nursing standard (Royal College of Nursing (Great Britain) : 1987)*, vol. 31, no. 31, pp. 42–45, Mar 29, 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/28351240

[10] “Pulse oximetry training manual,” 2011. [Online]. Available: https://www.who.int/patientsafety/safesurgery/pulse_oximetry/who_ps_pulse_oxymetry_training_manual_en.pdf

[11] R. N. Pittman, *Oxygen Transport in Normal and Pathological Situations: Defects and Compensations.* Morgan and Claypool Life Sciences, 2011. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK54113/

[12] "mediologiest", "Hypoxia: definition, symptoms, treatments," 2018. [Online]. Available: https://www.mediologiest.com/hypoxia/

[13] I. yasaei", "Hypoxia and hyperventilation," 2018. [Online]. Available: https://www.slideserve.com/diallo/hypoxia-and-hyperventilation

[14] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, Z. Cheng, T. Yu, J. Xia, Y. Wei, W. Wu, X. Xie, W. Yin, H. Li, M. Liu, Y. Xiao, H. Gao, L. Guo, J. Xie, G. Wang, R. Jiang, Z. Gao, Q. Jin, J. Wang, and B. Cao, "Clinical features of patients infected with 2019 novel coronavirus in wuhan, china," *The Lancet*, vol. 395, no. 10223, pp. 497 – 506, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/downloads/Huang-2019-nCoV-clinical-features-Lancet-1-24-2020.pdf

[15] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, Y. Zhao, Y. Li, X. Wang, and Z. Peng, "Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus-infected pneumonia in wuhan, china," *JAMA*, vol. 323, 02 2020.

[16] I. "David C. Dugdale, "Coronavirus disease 2019 (covid19)," 2020. [Online]. Available: https://www.pennmedicine.org/for-patients-and-visitors/patient-information/conditions-treated-a-to-z/coronavirus-disease-2019-covid19

[17] H. A. Rothan and S. N. Byrareddy, "The epidemiology and pathogenesis of coronavirus disease (covid-19) outbreak," *Journal of Autoimmunity*, vol. 109, p. 102433, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0896841120300469

[18] S. LaMotte", "Silent hypoxia: Covid-19 patients who should be gasping for air but aren't," 2020. [Online]. Available: https://www.cnn.com/2020/05/06/health/happy-hypoxia-pulse-oximeter-trnd-wellness/index.html

[19] K. Gaines, "Know your abgs - arterial blood gases explained." [Online]. Available: https://nurse.org/articles/arterial-blood-gas-test/

[20] Radiometer, "The principles of transcutaneous monitoring." [Online]. Available: https://www.radiometeramerica.com/en-us/products/transcutaneous-monitoring/the-principles-of-transcutaneous-monitoring

[21] J. Goodman, "Arterial blood gases, digital pulse oximetry, and routine blood work." [Online]. Available: http://www.oxyview.com/Portals/0/PDF/Arterial\%20Blood\%20Gases,\%20Digital\%20Pulse\%20Oximetry,\%20and\%20Routine\%20Blood\%20Work.pdf

[22] N. Domain", "Transcutaneous oximetry," 2009. [Online]. Available: https://www.cheyenneregional.org/location/heart-vascular-surgical-services/patient-resources/tests/transcutaneous-oximetry/

[23] X. Ge, P. Adangwa, J. Y. Lim, Y. Kostov, L. Tolosa, R. Pierson, D. Herr, and G. Rao, "Development and characterization of a point-of care rate-based transcutaneous respiratory status monitor," *Medical Engineering & Physics*, vol. 56, pp. 36 – 41, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350453318300511

[24] J. Farkas, "Top 10 reasons pulse oximetry beats abg for assessing oxygenation," 2016. [Online]. Available: https://emcrit.org/pulmcrit/pulse-oximetry/

[25] E. Niggemeyer, "Corience: Pulse oximetry<br>," 2013. [Online]. Available: http://www.corience.org/diagnosis-and-treatment/diagnosis-of-congenital-heart-defects/pulse-oximetry/

[26] W. Johnston, "Development of a signal processing library for extraction of spo2, hr, hrv, and rr from photoplethysmographic waveforms," *Masters Theses (All Theses, All Years)*, -07-31 2006. [Online]. Available: https://digitalcommons.wpi.edu/etd-theses/919

[27] P. P. S. Authority, "Skin integrity issues associated with pulse oximetry," 2005. [Online]. Available: http://patientsafety.pa.gov/ADVISORIES/Documents/200506_25.pdf

[28] "Spo2 vs. sao2?" [Online]. Available: https://allnurses.com/spo-vs-sao-t191378/

[29] nthu, "Beer-lambert law." [Online]. Available: http://life.nthu.edu.tw/~labcjw/BioPhyChem/Spectroscopy/beerslaw.htm

[30] S. Bhuckory, J. C. Kays, and A. M. Dennis, "In vivo biosensing using resonance energy transfer," *Biosensors*, vol. 9, no. 2, 2019. [Online]. Available: https://www.mdpi.com/2079-6374/9/2/76

[31] Scharf, "Green light pulse oximeter," November 03, 1998. [Online]. Available: https://patents.google.com/patent/US5830137A/en

[32] P. Aroul, "Miniaturized pulse oximeter reference design," *Healthtech*, 2014. [Online]. Available: http://www.ti.com/lit/ug/tidu542/tidu542.pdf?ts=1588534192131

[33] "Reflective ppg integrated sensor - apmkorea." [Online]. Available: https://hansangh2.en.ec21.com/Reflective-PPG-Integrated-Sensor--10605888.html?view_type=Gallery\&pageNo=1

[34] J. Kalinowski, *Organic light-emitting diodes Principles, characteristics, and processes.* New York: Marcel Dekker, 2005, includes bibliographical references (p. 427-462) and index.; ID: alma9936761562704746.

[35] "Light measurement handbook," 2000. [Online]. Available: http://www.dfisica.ubi.pt/~hgil/fotometria/HandBook/ch07.html

[36] B. E. Khoo, "Forensic bloodstain imaging: A digital method for stain enhancement and background reduction," 2014. [Online]. Available: https://www.researchgate.net/figure/LED-diagram-size-d5mm-viewing-angle-2th15_fig2_272017494

[37] R. Sharpeshkar, *Low-power pulse oximetry*, 1st ed., ser. Ultra Low Power Bioelectronics. Cambridge University Press, 2010, pp. 595–601.

[38] ——, *Low-power pulse oximetry*, 1st ed., ser. Ultra Low Power Bioelectronics. Cambridge University Press, 2010, pp. 275–280.

[39] K. W. Busch and M. A. Busch, *Multielement Detection Systems for Spectrochemical Analysis.* Wiley-Interscience, 1990.

[40] L. Orozco, "Optimizing precision photodiode sensor circuit design," 2014. [Online]. Available: https://www.analog.com/en/technical-articles/optimizing-precision-photodiode-sensor-circuit-design.html

[41] J. G. Webster, *Design of pulse oximeters.* Milton Park: Taylor & Francis, 1997. [Online]. Available: http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&local_base=BVB01&doc_number=025161193&sequence=000002&line_number=0001&func_code=DB_RECORDS&service_type=MEDIA

[42] L. Butler", "Amplifier noise," 1985. [Online]. Available: http://users.tpg.com.au/ldbutler/AmpNoise.htm

[43] J. Vega", "Source resistance and noise considerations in amplifiers," 2012. [Online]. Available: http://www.ti.com/lit/an/slyt470/slyt470.pdf

[44] A. Devices, "Oxygen saturation (spo2) measurement." [Online]. Available: https://www.analog.com/en/applications/markets/healthcare-pavilion-home/vital-signs-measurement/spo2.html

[45] J. J. Carr, "Secrets of rf circuit design," 2001. [Online]. Available: http://accessengineeringlibrary.com/browse/secrets-of-rf-circuit-design-third-edition

[46] "Introduction to frequency filters - analog and digital filters," -09-13T19:34:35+00:00 2017. [Online]. Available: https://911electronic.com/analog-digital-filters/

[47] Siemens, "Introduction to filters: Fir versus iir," 2019. [Online]. Available: https://community.sw.siemens.com/s/article/introduction-to-filters-fir-versus-iir

[48] A. Faizan, "Difference between analog and digital filter." [Online]. Available: http://electricalacademia.com/electrical-comparisons/analog-filter-vs-digital-filter-difference-between-analog-and-digital-filter/

[49] M. Computing, "Analog to digital conversion," 2010. [Online]. Available: https://www.mccdaq.com/PDFs/specs/Analog-to-Digital.pdf

[50] "industrial electronics", "Analog-to-digital conversion," 2008. [Online]. Available: http://www.industrial-electronics.com/DAQ/IOT-2-ADC.html

[51] K. Macalanda, "Fundamentals to automotive led driver circuits," 2019. [Online]. Available: http://www.ti.com/lit/wp/slyy163/slyy163.pdf

[52] "Driving leds on higher voltage," 2020. [Online]. Available: https://lednique.com/driving-leds-on-higher-voltage/

[53] W. Jung, “References and low dropout linear regulators,” 1998. [Online]. Available: https://www.analog.com/media/en/training-seminars/design-handbooks/Practical-Design-Techniques-Power-Thermal/Section2.pdf

[54] T. Instruments, “Low dropout regulators quick reference guide,” 2018. [Online]. Available: http://www.ti.com/lit/sg/slyt228p/slyt228p.pdf

[55] B. S. Lee, “Understanding the terms and definitions of ldo voltage regualtors,” Texas Instruments, Tech. Rep. SLVA079, October 1999. [Online]. Available: https://www.ti.com/lit/an/slva079/slva079.pdf

[56] J. Patoux, “Ask the applications engineer 37: Low dropout regulators,” 2007. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/low-dropout-regulators.html

[57] B. Sun, C. Wang, X. Chen, Y. Zhang, and H. Shao, “Ppg signal motion artifacts correction algorithm based on feature estimation,” *Optik*, vol. 176, pp. 337 – 349, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0030402618313883

[58] B. Carter, “A single-supply op-amp circuit collection,” 2000. [Online]. Available: https://mil.ufl.edu/4924/docs/TI_SingleSupply_OpAmp.pdf

[59] M. Day, “Understanding low drop out (ldo) regulators,” 2007. [Online]. Available: http://www.ti.com/download/trng/docs/seminar/Topic\%209\%20-\%20Understanding\%20LDO\%20dropout.pdf

[60] N. Sani, W. Mansor, Y. K. Lee, N. Zainudin, and S. A. Mahrim, “Determination of heart rate from photoplethysmogram using fast fourier transform,” 05 2015, pp. 168–170.

[61] C. C and B. A., “Physiology, respiratory rate.” 01 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK537306/

[62] (2017, 12/31/ 09:46:33 PM UTC) Ppg signal analysis wiki. [Online]. Available: https://everipedia.org/wiki/lang_en/ppg-signal-analysis

[63] D. J. Ewing, J. M. Neilson, and P. Travis, “New method for assessing cardiac parasympathetic activity using 24 hour electrocardiograms.” *Heart*, vol. 52, no. 4, pp. 396–402, oct 1984. [Online]. Available: https://doi.org/10.1136\%2Fhrt.52.4.396

[64] J. Bigger, R. E. Kleiger, J. L. Fleiss, L. M. Rolnitzky, R. C. Steinman, and J. Miller, "Components of heart rate variability measured during healing of acute myocardial infarction," *The American Journal of Cardiology*, vol. 61, no. 4, pp. 208 – 215, 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0002914988909174

[65] J. E. Mietus, C.-K. Peng, I. Henry, R. L. Goldsmith, and A. L. Goldberger, "The pnnx files: re-examining a widely used heart rate variability measure," *Heart*, vol. 88, no. 4, pp. 378–380, 2002. [Online]. Available: https://heart.bmj.com/content/88/4/378

[66] "Insight into esp32 sleep modes & their power consumption," -12-23T14:35:08+00:00 2018. [Online]. Available: https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/

[67] (2020) Rad-8. [Online]. Available: https://www.masimo.com/products/continuous/rad8/

[68] C. Education", "What is r-squared?" 2015. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/other/r-squared/

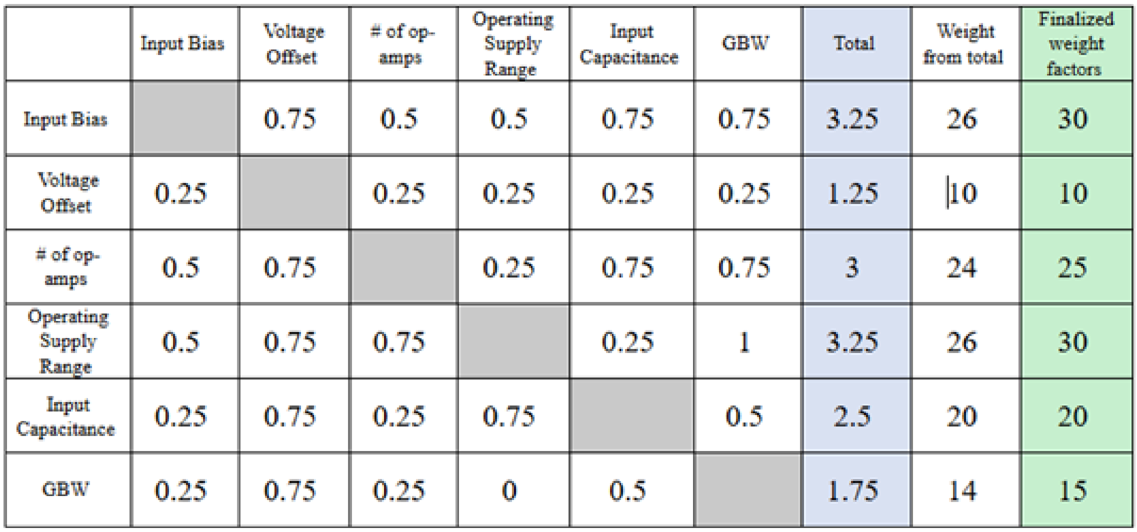
[69] D. NUNAN, G. R. H. SANDERCOCK, and D. A. BRODIE, "A quantitative systematic review of normal values for short-term heart rate variability in healthy adults," *Pacing and Clinical Electrophysiology*, vol. 33, no. 11, pp. 1407–1417, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-8159.2010.02841.x

[70] "Fda's role in regulating medical devices," 2018. [Online]. Available: https://www.fda.gov/medical-devices/home-use-devices/fdas-role-regulating-medical-devices

[71] "Classify your medical device," 2020. [Online]. Available: https://www.fda.gov/medical-devices/overview-device-regulation/classify-your-medical-device

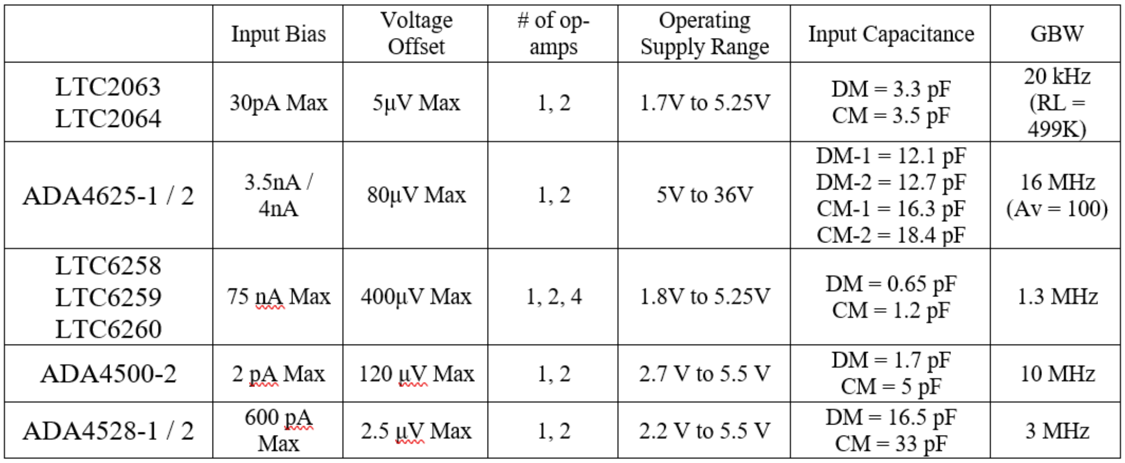
A. Amplifier Decision Matrix

Table A.1: Weight Factors for Amplifier Selection



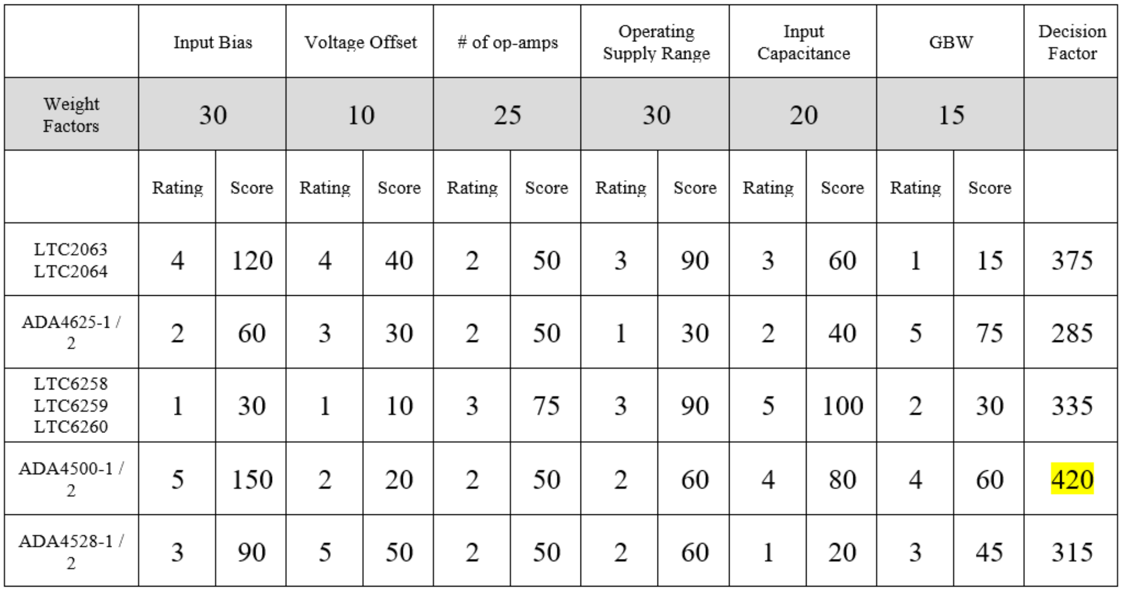
	Input Bias	Voltage Offset	# of op-amps	Operating Supply Range	Input Capacitance	GBW	Total	Weight from total	Finalized weight factors
Input Bias		0.75	0.5	0.5	0.75	0.75	3.25	26	30
Voltage Offset	0.25		0.25	0.25	0.25	0.25	1.25	10	10
# of op-amps	0.5	0.75		0.25	0.75	0.75	3	24	25
Operating Supply Range	0.5	0.75	0.75		0.25	1	3.25	26	30
Input Capacitance	0.25	0.75	0.25	0.75		0.5	2.5	20	20
GBW	0.25	0.75	0.25	0	0.5		1.75	14	15

Table A.2: Amplifier Datasheet Extraction



	Input Bias	Voltage Offset	# of op-amps	Operating Supply Range	Input Capacitance	GBW
LTC2063 LTC2064	30pA Max	5µV Max	1, 2	1.7V to 5.25V	DM = 3.3 pF CM = 3.5 pF	20 kHz (RL = 499K)
ADA4625-1 / 2	3.5nA / 4nA	80µV Max	1, 2	5V to 36V	DM-1 = 12.1 pF DM-2 = 12.7 pF CM-1 = 16.3 pF CM-2 = 18.4 pF	16 MHz (Av = 100)
LTC6258 LTC6259 LTC6260	75 nA Max	400µV Max	1, 2, 4	1.8V to 5.25V	DM = 0.65 pF CM = 1.2 pF	1.3 MHz
ADA4500-2	2 pA Max	120 µV Max	1, 2	2.7 V to 5.5 V	DM = 1.7 pF CM = 5 pF	10 MHz
ADA4528-1 / 2	600 pA Max	2.5 µV Max	1, 2	2.2 V to 5.5 V	DM = 16.5 pF CM = 33 pF	3 MHz

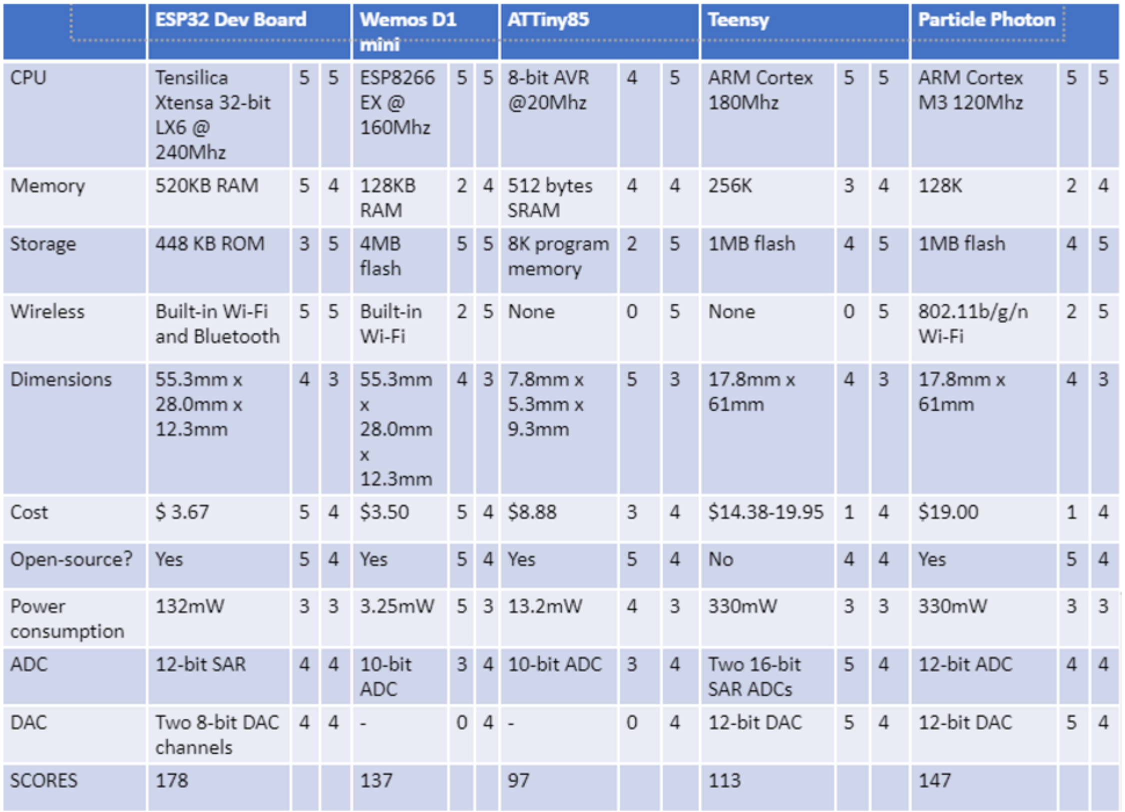
Table A.3: Value Analysis for Amplifiers



	Input Bias		Voltage Offset		# of op-amps		Operating Supply Range		Input Capacitance		GBW		Decision Factor
Weight Factors	30		10		25		30		20		15		
	Rating	Score	Rating	Score	Rating	Score	Rating	Score	Rating	Score	Rating	Score	
LTC2063 LTC2064	4	120	4	40	2	50	3	90	3	60	1	15	375
ADA4625-1 / 2	2	60	3	30	2	50	1	30	2	40	5	75	285
LTC6258 LTC6259 LTC6260	1	30	1	10	3	75	3	90	5	100	2	30	335
ADA4500-1 / 2	5	150	2	20	2	50	2	60	4	80	4	60	420
ADA4528-1 / 2	3	90	5	50	2	50	2	60	1	20	3	45	315

B. Microcontrollers Decision Matrix

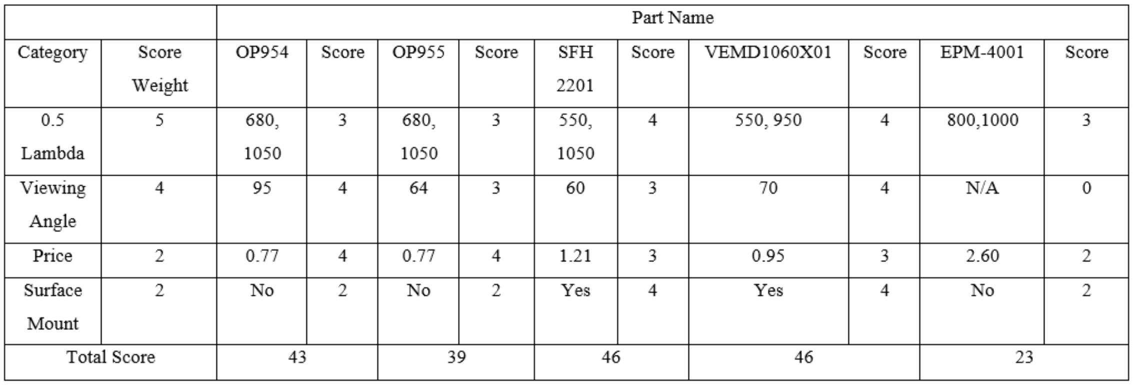
Table B.1: Value Analysis for Microcontrollers



	ESP32 Dev Board			Wemos D1 mini			ATTiny85			Teensy			Particle Photon		
CPU	Tensilica Xtensa 32-bit LX6 @ 240Mhz	5	5	ESP8266 EX @ 160Mhz	5	5	8-bit AVR @20Mhz	4	5	ARM Cortex 180Mhz	5	5	ARM Cortex M3 120Mhz	5	5
Memory	520KB RAM	5	4	128KB RAM	2	4	512 bytes SRAM	4	4	256K	3	4	128K	2	4
Storage	448 KB ROM	3	5	4MB flash	5	5	8K program memory	2	5	1MB flash	4	5	1MB flash	4	5
Wireless	Built-in Wi-Fi and Bluetooth	5	5	Built-in Wi-Fi	2	5	None	0	5	None	0	5	802.11b/g/n Wi-Fi	2	5
Dimensions	55.3mm x 28.0mm x 12.3mm	4	3	55.3mm x 28.0mm x 12.3mm	4	3	7.8mm x 5.3mm x 9.3mm	5	3	17.8mm x 61mm	4	3	17.8mm x 61mm	4	3
Cost	$ 3.67	5	4	$3.50	5	4	$8.88	3	4	$14.38-19.95	1	4	$19.00	1	4
Open-source?	Yes	5	4	Yes	5	4	Yes	5	4	No	4	4	Yes	5	4
Power consumption	132mW	3	3	3.25mW	5	3	13.2mW	4	3	330mW	3	3	330mW	3	3
ADC	12-bit SAR	4	4	10-bit ADC	3	4	10-bit ADC	3	4	Two 16-bit SAR ADCs	5	4	12-bit ADC	4	4
DAC	Two 8-bit DAC channels	4	4	-	0	4	-	0	4	12-bit DAC	5	4	12-bit DAC	5	4
SCORES	178			137			97			113			147		

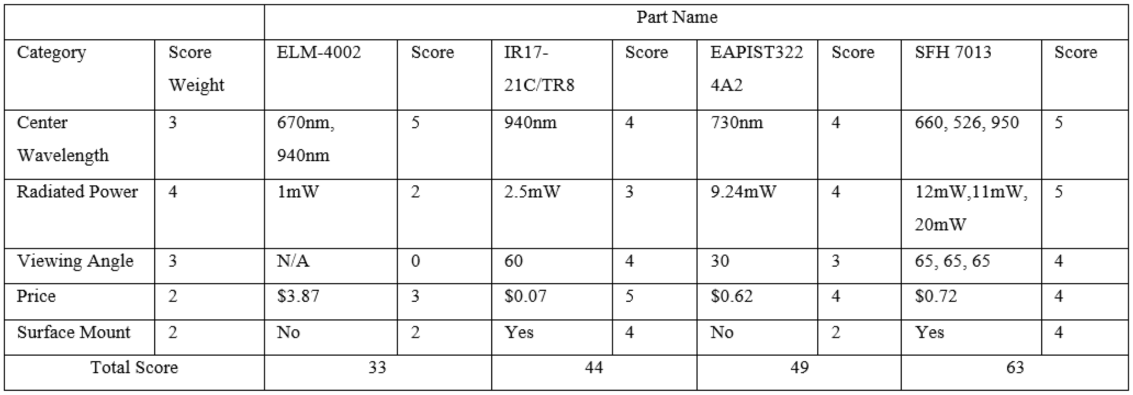
C. LEDS and Photodiode Decision Matrix

Table C.1: Value Analysis for Photodiodes



		Part Name									
Category	Score Weight	OP954	Score	OP955	Score	SFH 2201	Score	VEMD1060X01	Score	EPM-4001	Score
0.5 Lambda	5	680, 1050	3	680, 1050	3	550, 1050	4	550, 950	4	800,1000	3
Viewing Angle	4	95	4	64	3	60	3	70	4	N/A	0
Price	2	0.77	4	0.77	4	1.21	3	0.95	3	2.60	2
Surface Mount	2	No	2	No	2	Yes	4	Yes	4	No	2
Total Score		43		39		46		46		23	

Table C.2: Value Analysis for LEDs



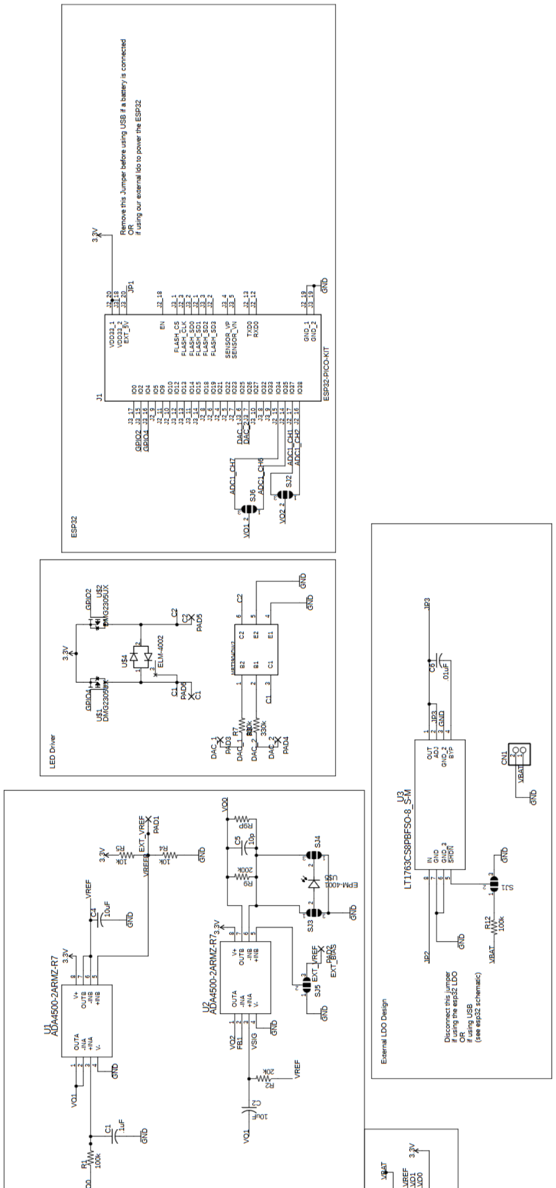
		Part Name							
Category	Score Weight	ELM-4002	Score	IR17-21C/TR8	Score	EAPIST3224A2	Score	SFH 7013	Score
Center Wavelength	3	670nm, 940nm	5	940nm	4	730nm	4	660, 526, 950	5
Radiated Power	4	1mW	2	2.5mW	3	9.24mW	4	12mW,11mW, 20mW	5
Viewing Angle	3	N/A	0	60	4	30	3	65, 65, 65	4
Price	2	$3.87	3	$0.07	5	$0.62	4	$0.72	4
Surface Mount	2	No	2	Yes	4	No	2	Yes	4
Total Score		33		44		49		63	

D. Measured AC Response of the Analog Filter

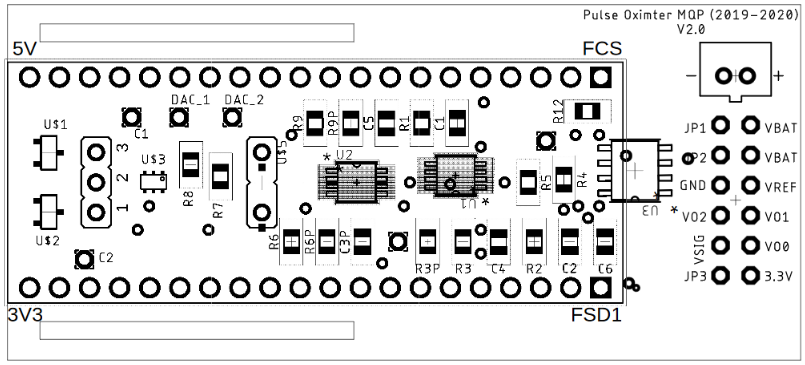
Table 8: Data Recorded of the Analog Filter

Frequency	Vin(V)	Vout(V)	Gain(V/V)	Gain (dB)	Period(s)	Delay(s)	Phase
0.1	3.0E-02	3.0E-01	1.0E1	2.0E1	1.0E1	7.2E0	-2.6E2
0.2	2.9E-02	5.9E-01	1.9E2	2.5E2	5.0E1	3.5E1	-2.5E3
0.3	2.9E-02	8.4E-01	2.8E3	2.9E3	3.3E2	2.3E2	-2.4E4
0.4	3.0E-02	1.0E3	3.6E4	3.1E4	2.5E3	1.0E3	-1.4E5
0.5	3.0E-02	1.3E4	4.4E5	3.2E5	2.0E4	9.0E-01	-1.6E6
1	2.0E-02	2.1E5	1.0E7	4.0E6	1.0E5	5.2E-01	-1.8E7
2	2.5E-02	2.6E6	1.0E8	4.0E7	5.0E-01	2.6E-01	-1.8E8
3	2.7E-02	2.8E7	1.0E9	4.0E8	3.3E-01	1.7E-01	-1.8E9
4	2.7E-02	2.9E8	1.0E10	4.0E9	2.5E-01	1.3E-01	-1.9E10
5	2.8E-02	2.9E9	1.0E11	4.0E10	2.0E-01	1.0E-01	-1.9E11
10	2.8E-02	2.61E10	9.1E11	3.9E11	1.0E-01	6.0E-02	-2.1E12
20	2.9E-02	2.01E11	6.9E12	3.6E12	5.0E-02	3.2E-02	-2.3E13
30	2.8E-02	1.53E12	5.3E13	3.4E13	3.3E-02	2.2E-02	-2.3E14
40	2.9E-02	1.23E13	4.2E14	3.2E14	2.5E-02	1.7E-02	-2.5E15
50	2.8E-02	1.0E14	3.5E15	3.0E15	2.0E-02	1.4E-02	-2.5E16
100	2.9E-02	5.2E-01	1.7E16	2.5E16	1.0E-02	7.6E-03	-2.7E17
200	2.9E-02	2.6E-01	9.0E16	1.9E17	5.0E-03	3.6E-03	-2.6E18
300	2.9E-02	1.8E-01	6.1E17	1.5E18	3.3E-03	8.7E-04	-9.4E18
400	2.9E-02	1.3E-01	4.6E18	1.3E19	2.5E-03	6.9E-04	-9.9E19
500	3.0E-02	1.1E-01	3.6E19	1.1E20	2.0E-03	4.7E-04	-8.4E20

E. Rigid PCB Schematic



PCB Schematic of the Wearable Sensor



Top Level of AFE, LED, Power Management Circuit

F. Bill of Materials

Table 9: BOM

Part	Value	Device	Package	Description	PRICE
C1	.1uF	C-USC0805	C0805	CAPACITOR,	0.19
C2	10uF	C-USC0805	C0805	CAPACITOR,	0.83
C4	10uF	C-USC0805	C0805	CAPACITOR,	0.83
C5	10pF	C-USC0805	C0805	CAPACITOR,	0.11
CN1		JST_2PIN-THM	JST-PH-2-THM	JST 2-Pin Connectors	0.17
J1	PICO	ESP32-PICO-KIT	PICO-KIT	Wifi Bluetooth Eval Board	None
JP6		JP6Q	JP6Q	JUMPER	
R1	100k	R-US_R0805	R0805	RESISTOR,	0.1
R2	20k	R-US_R0805	R0805	RESISTOR,	0.1
R3	100k	R-US_R0805	R0805	RESISTOR,	0.1
R4	10k	R-US_R0805	R0805	RESISTOR,	0.1
R5	10k	R-US_R0805	R0805	RESISTOR,	0.1
R6	18M	R-US_R0805	R0805	RESISTOR,	0.1
R7	330k	R-US_R0805	R0805	RESISTOR,	0.1
R8	330k	R-US_R0805	R0805	RESISTOR,	0.1
R9	200k	R-US_R0805	R0805	RESISTOR,	0.1
R12	100k	R-US_R0805	R0805	RESISTOR,	0.1
U1	-	DMG2305UX	SOT23	PMOSFET to enable LEDs	0.38
U3	-	MBT3904DW2	SOT363	BJT current source for LE	0.2
U4	-	ELM-4002	ELM-4002	Red and IR LED	3.87
U5	-	EPM-4001	EPM-4001	Photodiode	2.63
U6	-	ADA4500-2	RM_8	Op Amps in current design	2.81
U7	-	LT1763_S-M	SO-8_S-M	Fixed 3.3V LDO	5.02

G. Microcontroller and Algorithm Code

G.1. Embedded microcontroller code

```
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>

BLECharacteristic *pCharacteristic;
bool deviceConnected = false;

#define SERVICE_UUID        "4fafc201-1fb5-459e-8fcc-c5c9c331914b" // UART
    service UUID
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"

class MyServerCallbacks: public BLEServerCallbacks {
    void onConnect(BLEServer* pServer) {
      deviceConnected = true;
    };

    void onDisconnect(BLEServer* pServer) {
      deviceConnected = false;
    }
};

float ppg_array[20000] = {0}; // 16384
int red_counter = 0;
int ir_counter = 0; // 8192 + 525
long red_DC = 0;
long ir_DC = 0;
int state = 0;
int i = 0;
long sum_raw_data = 0;


volatile int interruptCounter;

hw_timer_t * timer = NULL;
```

```
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;

void IRAM_ATTR onTimer() {
  portENTER_CRITICAL_ISR(&timerMux);
  interruptCounter++;
  //ADCValue = analogRead(GPIO_38);
  portEXIT_CRITICAL_ISR(&timerMux);

}

void setup() {

  Serial.begin(115200);// 115200

  // Creating the BLE Device
  BLEDevice::init("MQP");
  // Create the BLE Server
  BLEServer *pServer = BLEDevice::createServer();
  pServer->setCallbacks(new MyServerCallbacks());
  // Create the BLE Service
  BLEService *pService = pServer->createService(SERVICE_UUID);

  // Create a BLE Characteristic
  pCharacteristic = pService->createCharacteristic(
                                      CHARACTERISTIC_UUID,
                                      BLECharacteristic::PROPERTY_READ |
                                      BLECharacteristic::PROPERTY_WRITE
                                    );

  pCharacteristic->addDescriptor(new BLE2902());

  // Start the service
  pService->start();

  // Start advertising
  pServer->getAdvertising()->start();


  timer = timerBegin(0, 80, true);
  timerAttachInterrupt(timer, &onTimer, true);
  timerAlarmWrite(timer, 2000, true); // 500 hz interrupt
  timerAlarmEnable(timer);

  // ADC setup (voltage ref)
  analogSetAttenuation(ADC_11db);

  pinMode(4, OUTPUT);
  pinMode(2, OUTPUT);
  digitalWrite(4, HIGH);
  digitalWrite(2, HIGH);
  dacWrite(26, 0); // RED
  dacWrite(25, 0); // IR - 112

}

void loop() {

  if (interruptCounter > 0) {

      portENTER_CRITICAL(&timerMux);
      interruptCounter--; // critical section
      portEXIT_CRITICAL(&timerMux);


      switch (state) {
        case 0:
          {
            digitalWrite(4, LOW);
            dacWrite(26, 70); // 80, 70 RED - was previously 75, 92
            for (int raw_cnt=0; raw_cnt<8; raw_cnt++) {
              sum_raw_data += analogRead(38)*3300/4095;
            }
            red_DC += analogRead(34)*3300/4095;

            state = 1;
          }
          break;
        case 1:
          {
              digitalWrite(4, HIGH);
              dacWrite(26, 0); // RED
              ppg_array[red_counter] = sum_raw_data/8;
```

```
       red_counter++;
       sum_raw_data = 0;
       state = 0;
      if (red_counter == 20000) { // 8192 + (525 to transmit values...)
       red_DC = red_DC/20000;
       state = 4;
       red_counter = 0;
      }
     }
     break;
    case 2:
     {
      digitalWrite(2, LOW);
      dacWrite(25, 70); // 80, 70 RED - was previously 75, 92
      for (int raw_cnt=0; raw_cnt<8; raw_cnt++) {
       sum_raw_data += analogRead(38)*3300/4095;
      }

      ir_DC += analogRead(34)*3300/4095;

      state = 3;
     }
     break;
    case 3:
     {
       digitalWrite(2, HIGH);
       dacWrite(25, 0); // RED
       ppg_array[ir_counter] = sum_raw_data/8;
       ir_counter++;
       sum_raw_data = 0;
       state = 2;
      if (ir_counter == 20000) { // 8192 + (525 to transmit values...)
       ir_DC = ir_DC/20000;
       state = 5;
       ir_counter = 0;
      }
     }
     break;
    case 4:
     {
      if (deviceConnected) {
        char txString[8];
        dtostrf(ppg_array[i], 7, 2, txString);
        pCharacteristic->setValue(txString);
        pCharacteristic->notify();
        i++;
        delay(100);
      }


    if (i==20000) { // 16384
        char txString[8];
        dtostrf(red_DC, 7, 2, txString);
        pCharacteristic->setValue(txString);
        pCharacteristic->notify();
        state = 2;
        i = 0;
    }
  }
  break;
case 5:
  {
    if (deviceConnected) {

        char txString[8];
        dtostrf(ppg_array[i], 7, 2, txString);
        pCharacteristic->setValue(txString);
        pCharacteristic->notify();
        i++;
        delay(100);
      }


    if (i==20000) { // 16384
        char txString[8];
        dtostrf(red_DC, 7, 2, txString);
        pCharacteristic->setValue(txString);
        pCharacteristic->notify();
        state = 0;
        i = 0;
    }
```

```
         }
         break;
     }


  }
}
```

G.2. SpO$_2$

```
    public double spo2_calculation(double[] ppg_array_all) {

       // Initialize variables, data structures and split input data into
           red and ir arrays
       Complex[] red_ppg_arr = new Complex[16384]; // 8192
       Complex[] ir_ppg_arr = new Complex[16384]; // 8192
       double red_DC_val = -1;
       double ir_DC_val = -1;
       double red_AC_val = 0;
       double ir_AC_val = 0;
       double SpO2_ratio = 0;
       double local_max_red = 0;
       double local_max_ir = 0;
       int red_local_max_index = 0;
       int ir_local_max_index = 0;
       double Fs = 250; // 50 sampling frequency
 //      int Fs = 1000;
       int array_length = 16384;
       double[] freq_values = new double[8193]; // 4097

       // DC values of RED and IR are transmitted over BLE
       red_DC_val = ppg_array_all[16384];
       ir_DC_val = ppg_array_all[32769];

       // Copy RED and IR values received from our pulse oximeter to their
           corresponding
       // arrays to perform FFT
       for (int i = 0; i < 16384; i++) { // 8192
    red_ppg_arr[i] = new Complex(ppg_array_all[i], 0);
    ir_ppg_arr[i] = new Complex(ppg_array_all[i+16385], 0); // in
        position 16384 there is a DC values
}
// Perform FFT to RED and IR waveforms
fft(red_ppg_arr);
fft(ir_ppg_arr);
// Compute two-sided spectrum of RED and IR by calculating absolute
    value of each point and
// dividing by length of FFT. Furthermore, we calculate frequency
    values associated with
// each one-sided spectrum value for RED and IR respectively
double[] abs_red_ppg_arr = new double[8193];
double[] abs_ir_ppg_arr = new double[8193];
for (int i = 0; i < 8193; i++) {
    abs_red_ppg_arr[i] = red_ppg_arr[i].abs()/array_length;
    abs_ir_ppg_arr[i] = ir_ppg_arr[i].abs()/array_length;
    freq_values[i] = (Fs * i)/array_length; // frequency values
        associated with each manipulated PPG waveform value
}
// Complete computation of one-sided spectrum by multiplying first
    half of the FFT points
// by 2, excluding the edges.
for (int i = 1; i < 8192; i++) {
    abs_red_ppg_arr[i] = 2 * abs_red_ppg_arr[i];
    abs_ir_ppg_arr[i] = 2 * abs_ir_ppg_arr[i];
}
// Assign DC component of RED and IR waveforms by assigning the 0th
    point of
// the one-sded spectrum respectively (amplitude at 0 Hz)
// Calculate AC component of RED and IR waveforms by finding local
    max in the range of
// 0.5 to 5 Hz
double freq = 0;
for (int i = 1; i < 8192; i++) {
    freq = freq_values[i];
    if ((freq >= 0.5) && (freq <= 5)) {
        if (local_max_red < abs_red_ppg_arr[i]) {
            local_max_red = abs_red_ppg_arr[i];
            red_local_max_index = i;
        }
```

```
            if (local_max_ir < abs_ir_ppg_arr[i]) {
                local_max_ir = abs_ir_ppg_arr[i];
                ir_local_max_index = i;
            }
        }
    }
    // assigning resulting local maxes to the corresponding AC
        component variables
    red_AC_val = local_max_red;
    ir_AC_val = local_max_ir;
    // Compute SpO2 ratio by dividing the normalized RED and IR AC
        components
    //(ACred/DCred)/(ACir/DCir)
    SpO2_ratio = (red_AC_val/red_DC_val)/(ir_AC_val/ir_DC_val);
    // Calculate SpO2 percentage by applying our empirical formula to
        the resulting ratio
    //(A - B * Ratio)
    double SpO2_percentage = 115 - SpO2_ratio * 25; // maybe 117 or
        112.75, will see
    if (SpO2_percentage > 100) {
        SpO2_percentage = 100;
    }
    return (SpO2_percentage);
}
```

G.3. Heart Rate

```
public double hr_calculation(double[] ppg_array_all) {
    Complex[] red_ppg_arr = new Complex[16384];
    double Fs = 250; // sampling frequency
    //int Fs = 1000;
    int array_length = 16384;
    double[] freq_values = new double[8193];
    double local_max_red = 0;
    int red_local_max_index = 0;
    double heart_rate = 0;
    for (int i = 0; i < 16384; i++) {
        red_ppg_arr[i] = new Complex(ppg_array_all[i], 0);
    }
    // FFT, find dominant frequency between 1 and 1.7 Hz (60 to 100
        bpm), multiply dominant frequency with 60
    // Perform fft
    fft(red_ppg_arr);
    double[] abs_red_ppg_arr = new double[8193];
    for (int i = 0; i < 8193; i++) {
        abs_red_ppg_arr[i] = red_ppg_arr[i].abs()/array_length;
        freq_values[i] = (Fs * i)/array_length; // frequency values
           associated with each manipulated PPG waveform value
    }

    // double values since one-sided...
    for (int i = 1; i < 8192; i++) {
        abs_red_ppg_arr[i] = 2 * abs_red_ppg_arr[i];
    }

    double freq = 0;

    for (int i = 1; i < 8192; i++) {
        freq = freq_values[i];
        if ((freq >= 1) && (freq <= 4)) { // 1.7
           if (local_max_red < abs_red_ppg_arr[i]) {
               local_max_red = abs_red_ppg_arr[i];
               red_local_max_index = i;
           }
        }
    }
    heart_rate = freq_values[red_local_max_index] * 60;
    return (heart_rate);
}
```

G.4. Respiration Rate

```
public double rr_calculation(double[] ppg_array_all) {

        Complex[] red_ppg_arr = new Complex[16384];
```

```
double Fs = 250; // sampling frequency
//int Fs = 1000;
int array_length = 16384;
double[] freq_values = new double[8193];
double local_max_red = 0;
int red_local_max_index = 0;
double respiratory_rate = 0;

for (int i = 0; i < 16384; i++) { // 0 to 8192
    red_ppg_arr[i] = new Complex(ppg_array_all[i], 0); // i
}

// FFT, find dominant frequency between 0.05 and 0.7 Hz (3 to 42
    breaths per minute), multiply dominant frequency with 60

// Perform fft
fft(red_ppg_arr);


double[] abs_red_ppg_arr = new double[8193];
for (int i = 0; i < 8193; i++) {
    abs_red_ppg_arr[i] = red_ppg_arr[i].abs()/array_length;
    freq_values[i] = (Fs * i)/array_length; // frequency values
        associated with each manipulated PPG waveform value

}


// double values since one-sided...
for (int i = 1; i < 8192; i++) {
    abs_red_ppg_arr[i] = 2 * abs_red_ppg_arr[i];
}

double freq = 0;

for (int i = 1; i < 8192; i++) {
    freq = freq_values[i];
    if ((freq >= 0.2) && (freq <= 0.34)) { // 0.05 to 0.7
        if (local_max_red < abs_red_ppg_arr[i]) {
            local_max_red = abs_red_ppg_arr[i];
            red_local_max_index = i;
```

```
              }
           }
       }

       respiratory_rate = freq_values[red_local_max_index] * 60;

       return (respiratory_rate);
    }
```

G.5. Short-term Heart Rate Variability

```
public double[] short_hrv_calculation(double[] ppg_array_all, double hr) {

       int minor_peak = 0;
       boolean found_first_peak = false;
       int peak_1_index = 0;
       int peak_2_index = 0;
       double[] peak_lengths = new double[16383]; // 8191
       int array_pointer = 0;
       double Fs = 250;
       double sequential_point_length = 1/Fs;

//       double hr_period = 60/hr;
//       double search_period = hr_period *1.5*Fs;
//       int num_element_search = (int) (250*search_period);

       double SDNN = 0;
       double COV = 0;
       double SDSD = 0;
       double RMSSD = 0;
       double NN50 = 0;
       double pNN50 = 0;


       // Will use either one (whichever is more precise)
       // Find length of NN Intervals (time) by locating peaks (local
          maxima) and
       // using sampling frequency and array indices, calculate time
```

```
            period between intervals
        for (int i = 5; i < 16379; i++) { // OR 8192 , <16384
            if (array_pointer < 16383) {

                if ((minor_peak == 1) && (!found_first_peak)) {
                    if ((ppg_array_all[i] > ppg_array_all[i - 1]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 1]) &&
                        (ppg_array_all[i] > ppg_array_all[i - 2]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 2]) &&
                        (ppg_array_all[i] > ppg_array_all[i - 3]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 3]) &&
                        (ppg_array_all[i] > ppg_array_all[i - 4]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 4])&&
                        (ppg_array_all[i] > ppg_array_all[i - 5]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 5])) { // &&
                        (ppg_array_all[i] > ppg_array_all[i - 2]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 2]) &&
                        (ppg_array_all[i] > ppg_array_all[i - 3]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 3]
                        peak_1_index = i;
                        found_first_peak = true;
                    }

                } else if (minor_peak == 2) {
                    if ((ppg_array_all[i] > ppg_array_all[i - 1]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 1]) &&
                        (ppg_array_all[i] > ppg_array_all[i - 2]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 2]) &&
                        (ppg_array_all[i] > ppg_array_all[i - 3]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 3]) &&
                        (ppg_array_all[i] > ppg_array_all[i - 4]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 4])&&
                        (ppg_array_all[i] > ppg_array_all[i - 5]) &&
                        (ppg_array_all[i] > ppg_array_all[i + 5])) {
                        peak_2_index = i;
                        // new array value
                        peak_lengths[array_pointer] = (peak_2_index -
                            peak_1_index) * sequential_point_length;
                        array_pointer++;
                        peak_1_index = peak_2_index;
//                        peak_2_index = 0;
```

```
              minor_peak = 1;
           }

       } else if ((ppg_array_all[i] > ppg_array_all[i-1]) &&
           (ppg_array_all[i] > ppg_array_all[i+1])) {
           minor_peak++;
       }
    }
}

// Calculate SDNN
// TODO: Make helper function for Standard Deviation.
// first find mean
int counter = 0;
while (peak_lengths[counter] != 0) {
    counter++;
}

for (int f = 0; f < peak_lengths.length; f++)
{
   peak_lengths[f] = peak_lengths[f]*1000; //turn to milliseconds
}

double total = 0;
for (int i = 0; i < peak_lengths.length; i++)
{
    total += peak_lengths[i];
}
double mean = total/counter;
// now calculate standard deviation
double sum = 0;
for (int i = 0; i < counter; i++)
{
    sum += Math.pow((peak_lengths[i] - mean), 2);
}

SDNN = Math.sqrt(sum/(counter - 1)); // Bessel’s correction

// Calculate COV - normalized SDNN
COV = SDNN/mean;
```

```
// Calculate SDSD
double[] successive_NN_diffs = new double[counter-1]; // 8190

for (int i=1; i<counter; i++) {
    successive_NN_diffs[i-1] = Math.abs(peak_lengths[i-1] -
        peak_lengths[i]);
}

// first find mean
double total_for_diff = 0;
for (int i = 0; i < successive_NN_diffs.length; i++)
{
    total_for_diff += successive_NN_diffs[i];
}
double mean_for_diff = total_for_diff/(successive_NN_diffs.length);
// now calculate standard deviation
double sum_for_std = 0;
for (int i = 0; i < successive_NN_diffs.length; i++)
{
    sum_for_std += Math.pow((successive_NN_diffs[i] -
        mean_for_diff), 2);
}

SDSD = Math.sqrt(sum_for_std/(successive_NN_diffs.length - 1));

// Calculate RMSSD
double summation_squares = 0;
for (int i=0; i<successive_NN_diffs.length;i++) {
    summation_squares += Math.pow(successive_NN_diffs[i], 2);
}

RMSSD = Math.sqrt(summation_squares/successive_NN_diffs.length);

// Calculate NN50
for (int i=0;i<successive_NN_diffs.length;i++) {
    if (successive_NN_diffs[i] > 50) {
        NN50++;
    }
}
// Calculate pNN50
pNN50 = NN50/successive_NN_diffs.length;
```

```
    if ((pNN50 < 0.01) && (pNN50 > 0)) {
        pNN50 = 0;
    }



    double[] indices_array = {SDNN, COV, SDSD, RMSSD, NN50, pNN50};
    return indices_array;
}
```

G.6. Long-term Heart Rate Variability

```
public double long_term_hrv(Complex[] half_ppg_array_all_24) {
    int N = half_ppg_array_all_24.length;
    int Fs = 250;
    double psd_term= (double) 1/(Fs * N);
    double freq_term = (double) Fs/N;
    // Perform FFT
    fft(half_ppg_array_all_24);
    // Find PSD
    double[] abs_ppg_arr = new double[(N/2)+1];
    double[] corr_frequencies = new double[(N/2)+1];
    for (int i=0; i<=(N/2); i++) {
        abs_ppg_arr[i] = psd_term * half_ppg_array_all_24[i].abs();

        if ((i!=0) || (i!=(N/2))) {
            abs_ppg_arr[i] = 2*abs_ppg_arr[i];
        }
        // 0:Fs/N:Fs/2
        corr_frequencies[i] = i * freq_term;
    }

    double local_max_sym = 0; // for sympathetic activity
    int sym_local_max_index = 0;
    double local_max_para = 0; // for parasympathetic activity
    int para_local_max_index = 0;

    for (int i = 1; i < (N/2); i++) {
            // Sympathetic actiity
            if ((corr_frequencies[i] >= 0.01) && (corr_frequencies[i] <=
                0.15)) {
                if (local_max_sym < abs_ppg_arr[i]) {
                    local_max_sym = abs_ppg_arr[i];
                    sym_local_max_index = i;
                }
            }

            // Parasympathetic actiity
            if ((corr_frequencies[i] >= 0.15) && (corr_frequencies[i] <=
                0.5)) {
                if (local_max_para < abs_ppg_arr[i]) {
                    local_max_para = abs_ppg_arr[i];
                    para_local_max_index = i;
                }
            }
        }

        double hrv_pwr_ratio = local_max_sym/local_max_para; // LF/HF
        return hrv_pwr_ratio;
    }
```

H. Informed Consent Agreement

Informed Consent Agreement for Participation in a Research Study
Principal Investigator:
Ulkuhan Guler, uguler@wpi.edu
Student Investigators:
Franco Baudino, fabaudino@wpi.edu
Bill Chieng, bchieng@wpi.edu
Fivos Kavassalis, fikavassalis@wpi.edu

Introduction
You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the study The purpose of this study is to evaluate the functionality of our prototype flexible wearable pulse oximeter. This includes extracting biometric information such as blood oxygen level, heart rate, respiratory rate, and heart rate variability. The measurements from our prototype device will be compared to the measurements from a commercial oxygen level and heart rate sensor and an Apple watch. Our prototype device will be placed on your left index finger and right forearm. The commercial device will only be placed on your right index finger and on your right wrist. Both devices use light to measure your blood oxygen level and heart rate. Therefore, you should feel no discomfort or warming of the skin at all, please let the test administrator know if you are feeling any kind of discomfort. The test should take no longer than 25 minutes.

Procedures to be followed

1. Each volunteer will be asked to wipe the commercial device (Masimo Rad-8), the Apple watch, both index fingers and their right wrist with an alcohol wipe to sanitize the area.

2. The commercial device (Masimo Rad-8) will be placed on the right index finger, the Apple watch will be placed on the right wrist and our prototype device will be placed on the left index finger.

3. The data extracted from the sensors will be stored in an excel file within a digital folder that will be secure stored in our MQP's private OneDrive folder. Each collection will be labelled as volunteer # to minimize any personal identifier.

4. After enough usable data is extracted from the index finger (5 minutes), the right forearm of the subject will be alcohol wiped.
5. The prototype will then be placed on the subject's forearm.
6. After enough data has been collected (5 minutes) both the commercial and prototype devices will be removed from the subject.

Record keeping and confidentiality
The expected reported information will include the following reflected light intensity, blood oxygen saturation, respiration rate, heart rate, and heart variability. This information is being stored into an online data base that only the investigators of this study have access to. Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or it's designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury
We do not expect serious injuries to be caused by our testing. There is a minimal risk of skin reactions for subjects who have a history of skin conditions. If injury or harm during the period of trial, the investigators conducting the study will contact WPI EMS.
You do not give up any of your legal rights by signing this statement.

Cost/Payment
There is no cost or payment for this testing
For more information about this research or about the rights of research participants, or in case of research-related injury, contact:

Principal Investigator:
Ulkuhan Guler uguler@wpi.edu
Student Investigators:
Franco Baudino, fabaudino@wpi.edu
Bill Chieng, bchieng@wpi.edu
Fivos Kavassalis, fikavassalis@wpi.edu
IRB Manager:
Ruth McKeogh
Tel: 508-831-6699

Email: irb@wpi.edu
Human Protection Administrator:
Gabriel Johnson
Tel: 508-831-4989 Email: gjohnson@wpi.edu
Your participation in this research is voluntary.

Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.
X:
Date:

I. MQP Test Participant Advertisement

TEST SUBJECTS WANTED FOR FLEXIBLE WEARABLE SENSOR MQP
TIME REQUIREMENT: < 25min
WHERE: AK 319
WHEN: All Throughout D-term
Advisor: Professor Ulkuhan Guler
CONTACT ANYONE OF THE STUDENT RESEARCHERS FOR MORE INFORMATION

We are a group of MQP students who need test subjects for our MQP project. We are creating a non-invasive wearable blood oxygen sensor for our MQP. Our sensor uses light to take non-invasive measurements. We have reached the phase in our project where we need to compare the accuracy of our measurements to those commercial sensors.

A commercial sensor (Masimo Rad-8) and an Apple watch, shown in the figures below, will be placed on your right index finger and wrist respectively. Our prototype will be placed on your left index finger and right forearm. The test will be finished after we record some measurements. The entire test should take no more than 25 minutes.