EXTENDED-USE ECG MONITOR

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Biomedical Engineering

by
Daniel Aaron Soski
June 2018
COMMITTEE MEMBERSHIP

TITLE: Extended-Use ECG Monitor

AUTHOR: Daniel Aaron Soski

DATE SUBMITTED: June 2018

COMMITTEE CHAIR: Robert Szlavik, Ph.D.
Professor of Biomedical Engineering

COMMITTEE MEMBER: Tina Smilkstein, Ph.D.
Associate Professor of Electrical Engineering

COMMITTEE MEMBER: David Clague, Ph.D.
BMED Department Graduate Coordinator
Professor of Biomedical Engineering
ABSTRACT

Extended-Use ECG Monitor

Daniel Aaron Soski

In this thesis, a prototype ECG monitor was developed that is integrated into an elastic shirt and takes a 3-lead ECG for over 5 days. The high-quality measurements can be used to identify markers indicative of various detrimental heart conditions. Measurements recorded by the device are encrypted and stored onto a micro-SD card. Current Holter monitors are expensive and have functional lives less than 48 hours; however, extended duration monitoring has been proven more useful in diagnosis. The device designed demonstrates that ECG measurements can be taken over longer durations without sacrificing quality, comfort, or device cost.
ACKNOWLEDGMENTS

I owe a huge thanks to my, advisor Dr. Robert Szlavik, for his guidance throughout my long journey completing this thesis. He has remained extremely patient and flexible throughout the entire process. The many forks in the road and backtracks that he coached me through has given me a much greater understanding of myself and how to best help others.

I am also grateful to both of my committee members, Dr. Tina Smilkstein and Dr. David Clague. Dr. Smilkstein’s contagious enthusiasm for electronics and biomedical technology kept me excited with my classwork and helped me stay on this path when things got tough. Dr. Clague’s curiosity and passion for everything biomedical caused me to be interested in many aspects that I previously would never have found interest in.

I would also like to thank CPConnect for providing me funding for this project. The funding allowed me to pay for all of the prototype components and manufacturing services that I used. This turned the theoretical design into a functional prototype.

A big thanks goes to all my professors, lecturers, lab instructors, and administrators for believing in Cal Poly’s “Learn by Doing” motto and encouraging my classmates and me to take it to heart. It has truly changed the way I solve new problems and work with others.

Last but not least, I would like to thank my family and friends for their forever continual support and encouragement especially while working on this thesis.
TABLE OF CONTENTS

LIST OF TABLES .......................................................................................................................... viii
LIST OF FIGURES ......................................................................................................................... ix

CHAPTER
1. INTRODUCTION .......................................................................................................................... 1
  1.1 Objective ............................................................................................................................... 1
  1.2 Need .................................................................................................................................... 1
  1.3 Background .......................................................................................................................... 2
  1.3.1 The Electrocardiogram (ECG) ....................................................................................... 2
  1.3.2 Glasgow Algorithm ......................................................................................................... 5
  1.3.3 Electrical Model of Skin and Electrodes ....................................................................... 5
  1.3.4 Signal-to-Noise Ratio .................................................................................................... 6
  1.3.5 Microcontrollers and Sampling ...................................................................................... 7
  1.3.6 Weighted Average Power Consumption ...................................................................... 8
  1.4 Current Solutions ............................................................................................................... 9
  1.4.1 Classical Holter Monitors .............................................................................................. 9
  1.4.2 Modern ECG Monitors ............................................................................................... 10
  1.5 Markets of Interest ........................................................................................................... 12
  1.5.1 Market Entry Criteria .................................................................................................. 12
  1.5.2 Consumer Market ........................................................................................................ 12
  1.5.3 Healthcare Market ....................................................................................................... 13
2. METHODS ................................................................................................................................... 14
  2.1 Circuit Simulation ............................................................................................................... 14
  2.2 PCB Design ....................................................................................................................... 14
  2.3 PCB Fabrication and Assembly ......................................................................................... 14
  2.4 Improvement Criteria ....................................................................................................... 14
    2.4.1 Measurement Quality ................................................................................................. 14
    2.4.2 Data Transfer .............................................................................................................. 15
    2.4.3 Functional Life ........................................................................................................... 16
      2.4.3.1 Comfort ................................................................................................................ 16
      2.4.3.2 Hardware Size ..................................................................................................... 17
      2.4.3.3 Battery Life ......................................................................................................... 17
    2.4.4 Device Cost ................................................................................................................ 17
    2.4.5 User Interaction .......................................................................................................... 18
    2.4.6 Device Requirements Summary ............................................................................... 18
3. RESULTS ................................................................................................................................... 19
  3.1 Device .................................................................................................................................. 19
  3.2 Measurement Quality ....................................................................................................... 22
    3.2.1 Instrumentation Amplifier ......................................................................................... 23
    3.2.2 Hardware Filters ....................................................................................................... 24
    3.2.3 Microcontroller Sampling ......................................................................................... 31
    3.2.4 Software Filters ....................................................................................................... 32
    3.2.5 Waveform Feature Comparison ............................................................................... 33
  3.3 Data Transfer ..................................................................................................................... 34
    3.3.1 Transfer Method Assessment ..................................................................................... 34
    3.3.2 WiFi ............................................................................................................................ 34
    3.3.3 BLE ............................................................................................................................ 35
    3.3.4 BT-EDR ..................................................................................................................... 36
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Functional Comparison of Classical and Modern Holter Monitors [3,8,10]</td>
<td>11</td>
</tr>
<tr>
<td>2. Production Device Requirements</td>
<td>18</td>
</tr>
<tr>
<td>3. Front-End Frequency Response Results</td>
<td>31</td>
</tr>
<tr>
<td>4. Weighted Average Microcontroller Current Draw by State</td>
<td>41</td>
</tr>
<tr>
<td>5. Device Requirements vs. Implementation</td>
<td>45</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>1. Electrode placement for (a) 3-lead and (b) 12-lead ECGs</td>
<td>3</td>
</tr>
<tr>
<td>2. Typical ECG signal [7]</td>
<td>4</td>
</tr>
<tr>
<td>3. Electrical Models of Wet and Dry Electrodes on Skin</td>
<td>6</td>
</tr>
<tr>
<td>4. A signal with a) small SNR and b) large SNR</td>
<td>7</td>
</tr>
<tr>
<td>6. Finger Placement using AliveCor Monitor</td>
<td>10</td>
</tr>
<tr>
<td>7. The final prototype device design – 3-Lead ECG with Electronics and Battery</td>
<td>20</td>
</tr>
<tr>
<td>8. Connected PCBs</td>
<td>21</td>
</tr>
<tr>
<td>9. Block Diagram of the 3-Lead ECG</td>
<td>21</td>
</tr>
<tr>
<td>10. Microcontroller, ADC, and Micro-SD Card PCB</td>
<td>22</td>
</tr>
<tr>
<td>11. Analog Front-End PCB</td>
<td>22</td>
</tr>
<tr>
<td>12. Block diagram of electrode analog front-end</td>
<td>23</td>
</tr>
<tr>
<td>13. Electrode-Amplifier Model</td>
<td>24</td>
</tr>
<tr>
<td>14. Instrumentation Amplifier and High-Pass Feedback Loop</td>
<td>25</td>
</tr>
<tr>
<td>16. Front-End Second Stage Sallen-Key Filter</td>
<td>26</td>
</tr>
<tr>
<td>17. Sallen-Key Filter Design Frequency Response</td>
<td>27</td>
</tr>
<tr>
<td>18. Front-End Third Stage Amplifier</td>
<td>28</td>
</tr>
<tr>
<td>19. Front-End Fourth Stage High-Pass Amplifier</td>
<td>28</td>
</tr>
<tr>
<td>20. Schottky Diode Clamp</td>
<td>29</td>
</tr>
<tr>
<td>21. Simulated Frequency Response of the Analog Front-End</td>
<td>29</td>
</tr>
<tr>
<td>22. Front-End Measurement Test Setup</td>
<td>30</td>
</tr>
<tr>
<td>23. Electrode Front-End Frequency Response</td>
<td>30</td>
</tr>
<tr>
<td>24. Sampling Period Variation Histogram</td>
<td>32</td>
</tr>
<tr>
<td>25. Results of MATLAB Filtering in (a) Time Domain and (b) Frequency Domain</td>
<td>33</td>
</tr>
<tr>
<td>26. ECG waveforms recorded by (a) the device and (b) a Philips PageWriter TC70</td>
<td>34</td>
</tr>
<tr>
<td>27. SD Card wiring diagram</td>
<td>37</td>
</tr>
<tr>
<td>28. Flow Diagram of Microcontroller Operation</td>
<td>38</td>
</tr>
<tr>
<td>29. Microcontroller Current Draw</td>
<td>40</td>
</tr>
<tr>
<td>30. Microcontroller Current Draw (During Sampling ADC/Sleep Period)</td>
<td>40</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

1.1 Objective
The goal of this project is to create a Holter monitor that improves upon current designs by increasing functional life and usability without sacrificing measurement quality. The motivation for this project stems from the failure of current Holter monitors to be more widely useful and accepted in screening and diagnosing disease which requires longer duration measurements.

1.2 Need
Heart disease is the leading cause of death in the United States and accounts for nearly 16% of worldwide deaths each year [1, 2] even though heart disease can be easily prevented and even treated in early stages with diet changes, vitamin and mineral supplements, medications, and lifestyle changes [2]. However, in later stages it is only treatable with surgery making it extremely beneficial to identify at-risk individuals early so that their heart condition may be monitored and managed [3].

The at-risk identification process currently employs two methods: risk factor assessment and electrocardiogram (or ECG) screening. The former relies on metrics: age, sex, blood pressure, smoking, lipid levels, diabetes, obesity, lifestyle, and family history. The U.S. Preventive Services Task Force (USPSTF) has determined that this strategy is currently the best method of assessment, noting that it is unclear if ECG screening has an added net benefit [2].

In the same investigation, the USPSTF agreed that measuring the electrical activity of the heart with an ECG can be used to identify increased risk for serious heart conditions, but recommend against it because of the cost and stress placed on patients involved in ECG screenings. An abnormal ECG reading in a healthy individual can be due to the “potential adverse effects of invasive tests.” Apprehension of measurements is shown to affect medical results and can do more harm than good for patients [2]. It is important to note that the ECG screening in question was done in a clinical environment as a single event and where the individual may have had anxiety about the test in an unfamiliar or uncomfortable environment.
The potential false positives that are caused by this type of screening can be due to events such as “an irregular heart rhythm that 'comes and goes', and the recording can be normal between episodes.” [4] Dr. Tim Kenny notes that specialized ECG recordings can overcome these issues. One such type of recording is the Ambulatory ECG. “This is where you wear a small monitor which constantly records your heart rhythm. This test records the electrical activity of your heart when you are walking about (ambulatory) and doing your normal activities. It aims to detect abnormal heart rhythms that may 'come and go'. The electrical activity is usually recorded for 24 to 48 hours” [2, 5]. By extending the duration of the ECG recording, the results can compensate for anomalies.

This ambulatory ECG is currently performed using Holter monitors; however, their use is limited because they suffer from issues such as high price, poor quality recordings, insufficient number of recordings, and uncomfortable apparatuses. These drawbacks have limited the ability of heart monitoring to very special cases and for short time periods. By developing a heart monitor that is comfortable enough for daily use while holding the quality of recordings constant, widespread use of ECG monitors may become more useful in detecting debilitating heart conditions such as heart disease.

1.3 Background

1.3.1 The Electrocardiogram (ECG)

The electrical activity of the human body can be measured at the surface of the skin with electrodes. Major sources of electrical activity include the brain, muscles, and the heart. The measurement taken of the electrical activity of the heart is called an electrocardiogram or ECG. The electrical activity of the heart is not symmetrical so the ECG electrical signal looks different depending on where the electrodes are placed on the skin. The different placements of electrodes give different "views" of the heart by looking at the electrical activity in a different direction. ECGs consist of 1, 3, 6, or 12 different views or 'leads'. The electrode placement for each lead configuration is shown in Figure 1.
Taking a 3-lead measurement requires a minimum of 4 electrodes. The four electrodes on the chest are labeled with right (R), left (L), arm (A), and leg (L) resulting in four labels right arm (RA), left arm (LA), right leg (RL), and left leg (LL). The RL electrode is used as the common node for the other electrode’s voltage measurement. Each lead is a difference of two electrodes:

- **Lead 1** = LA – RA
- **Lead 2** = LL – RA
- **Lead 3** = LL – LA

From these 3 equations it can be shown that Lead 3 is a combination of Lead 1 and Lead 2 by the equation:

\[
\text{Lead 3} = \text{Lead 2} - \text{Lead 1} = (\text{LL} - \text{RA}) - (\text{LA} - \text{RA}) = \text{LL} - \text{RA} - \text{LA} + \text{RA} = \text{LL} - \text{LA}
\]

Therefore, if Lead 1 and Lead 2 are measured, Lead 3 may be calculated.
The signal that is recorded during an ECG from each lead is divided into different parts, as shown in Figure 2. Each part corresponds to a specific action or set of specific actions of the heart [6]:

- **P wave**: Atrial depolarization
- **PR Segment**: Conduction through AV node and AV bundle
- **QRS Complex**: Ventricular depolarization
- **ST Segment**: Ventricles contract
- **T wave**: Ventricular repolarization

![Figure 2. Typical ECG signal](image)

Analyzing these different parts of the ECG recording can indicate how the heart is functioning [2]. An extended P wave can indicate atrial enlargement. Excessively short PR intervals suggest Wolf-Parkinson-White syndrome while long PR intervals can indicate atrioventricular block, both problems with the AV node. Excessively tall or short QRS complexes can indicate problems with the walls of the ventricles where the muscles are too thick or too stiff, known as ventricular
hypertrophy and restrictive cardiomyopathy. Elevated or depressed ST segments and inverted T waves can indicate coronary artery disease which restricts blood flow to the heart [8].

Every signal can be broken down into sinusoids at varying frequencies. ECG signals approximately occupy the frequency range of 0.5Hz – 150Hz [9].

**1.3.2 Glasgow Algorithm**

Dr. Peter Macfarlane and his team developed an algorithm to automatically detect abnormalities in a resting ECG. This algorithm has been improved over time and “is capable of diagnosing all commonly recognized ECG abnormalities such as myocardial infarction (MI), including acute MI, ventricular hypertrophy, ST-T abnormalities and common abnormalities of rhythm. Conduction defects and other abnormalities such as prolonged QT interval are also reported” [10]. Many other algorithms or variations of algorithms are used in clinical settings depending on the machine [11]. Each of these has varying features and accuracy.

**1.3.3 Electrical Model of Skin and Electrodes**

The electrical model of the skin-electrode interface creates trade-offs that affect how the electrodes are interfaced with the electronics. Measuring the ECG requires converting the biological signal to an electrical signal using electrodes. Electrical potential is generated in the electrode due to ionic current flow at the interface between the electrode and the skin, where the metallic surface of the electrode is the chemical electrode and body fluids are the electrolyte [12]. When the electrode is brought into contact with the electrolyte they exchange ions and electrons. This constant exchange allows for current to flow and creates a static potential difference between the electrode and the electrolyte, known as the half-cell potential.

Unfortunately, the top layer of skin, the stratum corneum consists mainly of dead skin cells and is mostly void of body fluids making it an electrical insulator. This is approximated electrically as a 1MΩ or greater resistor in parallel with a 10nF capacitor [13]. Beneath the stratum corneum is the stratum germinativum and the dermis, which are both full of fluids and very conductive to biopotential signals. These are represented as the body resistance which is typically negligible.
There are two methods of skin preparation typically used to improve the conductivity of the stratum corneum: abrasion and gel. Abrasion involves removing some of the top layers of skin to reduce the thickness of the stratum corneum or using electrodes with spikes to penetrate into the skin, bypassing some of the stratum corneum. Abrasion is typically used with dry electrodes and can be represented by reducing the resistance of the stratum corneum in varying amounts down to 100kΩ.

Wet electrodes, typically made of Ag-AgCl, rely on conductive gels. The gel diffuses into the stratum corneum and increases its conductivity, approximated electrically as a stratum corneum with a resistance of 100kΩ. The gel itself is represented electrically as a small resistance in parallel with a small capacitance, both negligible in comparison to the impedance of the stratum corneum. The electrical models of dry and wet electrodes are shown in Figure 3 [13].

![Electrical Models of Wet and Dry Electrodes on Skin](image)

**Figure 3. Electrical Models of Wet and Dry Electrodes on Skin**

1.3.4 Signal-to-Noise Ratio
The amplitude of the electrical signal generated by the electrodes are very weak, even when using the wet electrodes. In the initial state, the weak signal is hard to distinguish from random noise. External noise sources such as lights and portable electronics radiate electricity near the frequency range of the ECG signal, which can be coupled into the electrodes and other circuitry. Internal sources, such as components on the device itself, generate a small amount of noise because of thermal variations. To make the signal usable it must be amplified and filtered.
Signal quality is measured by signal-to-noise ratio (SNR), which is the relative measure of signal amplitude to noise amplitude. A signal with a large SNR is very easy to make out of the noise and a signal with a small SNR is hard to spot in the noise, shown in Figure 4.

![Figure 4](image.png)

**Figure 4. A signal with a) small SNR and b) large SNR**

The SNR is improved by increasing the amplitude of the signal more than the amplitude of the noise. The Noise Figure (NF) of an electronic system is a measure of the effect on SNR; defined as $\text{NF} = \text{SNR}_{\text{in}} - \text{SNR}_{\text{out}}$. Signals are typically cleaned up (improving SNR) using several stages of amplifiers and filters.

The Friis formula defines the effects of sequential stages on SNR [14]. The formula indicates that secondary stages are less sensitive to noise because of the gain of previous stages. Therefore, putting a low noise, high gain stage first reduces the necessity of further stages being low noise.

### 1.3.5 Microcontrollers and Sampling

Microcontrollers (MCUs) are digital devices so an analog to digital converter (ADC) is required to convert the analog ECG signal into a usable digital format. The ADC takes the analog value and
converts the voltage amplitude into a sampled digital value. Measuring the value of the analog signal at periodic instances in time to approximate the signal is called sampling. The frequency at which the signal is sampled is the sampling rate. The sampling rate has a huge effect on the quality of the approximated signal. The Nyquist Theorem says that the sampling rate must be 2x the highest frequency component of the signal [15]. If a lower sampling frequency is used, then the higher frequency components will be aliased onto the target signal, corrupting it. Filtering stages are used to limit the signal to the expected frequencies. For an ECG signal, who’s highest frequency component is 150Hz, the minimum sampling rate should be 300Hz.

One common type of ADC is the successive approximation register (SAR) ADC. Various ADC settings can be selected that affect the speed and quality of the measurement. The first two settings for the ADC are the number of digital bits that the analog value will be represented by and the reference voltage which is the maximum input value. A 12-bit number can represent numbers up to $2^{12}=4096$. For a 12-bit ADC with a 3.3V reference voltage, the analog signal will be approximated within $3.3V/4096=805.6\mu V$ per bit.

SAR ADCs use a sample-and-hold circuit to prevent the input voltage from changing while the conversion is taking place. This works by connecting the input signal to a capacitor for a period of time called the sample time and then disconnecting the input while the conversion takes place. The sample time allows the input signal to charge up the sample capacitor. When disconnected, the capacitor holds the voltage of the input signal at that time even if the input signal changes while the conversion is taking place. The ADC has a setting called sample speed which sets the sample time. A low impedance input signal will be able to charge the sample capacitor quickly, allowing for a shorter sample time. The conversion speed of the ADC is controlled directly by the ADC clock frequency. A faster clock frequency results in faster conversions but higher power consumption.

1.3.6 Weighted Average Power Consumption
Accurately calculating the power consumption of a complex system over long periods of time can be challenging and time consuming. Electrical systems consume differing amounts of power
depending on input signals, microcontroller state, and communication frequency which may vary over time. A time-weighted average can be employed to simplify this problem while still providing a first-order approximation of power consumption. This assumes that the system functionality is periodic or can be approximated as periodic and works by dividing the system functionality into a number of discrete states where the power consumption in constant. Each state’s fractional-duration and power consumption are calculated or measured. Fractional duration is the percent of time the state is active out of the entire period of the system. For periodic systems this is the state duration divided by the system period, or all of the state durations added together. For non-periodic systems, this is the state duration times the probability that the event will occur within one period divided by the system period.

1.4 Current Solutions

1.4.1 Classical Holter Monitors
Current Holter monitors are about the size of a small camera and must be worn on a belt or around the neck of the individual for the duration of the test, as shown in Figure 5. Wires connect the monitor to the electrodes that are adhered to the individual’s chest. A doctor or qualified ECG technician must be present to place the electrodes on the wearer as well as present to retrieve the recorded data from the Holter monitor. Finally, Holter monitors can cost between $300 and $10,000 depending on function and quality.

![Figure 5. A Classical Holter Monitor](image)
1.4.2 Modern ECG Monitors

There are several companies attempting to bring the Holter monitor into the 21st century, most notably AliveCor, ECG Check, and HeartCheck. These companies’ products, which are focused on increasing convenience and reducing cost, are limited to taking only a 1-lead ECG. They function by having the user place their index fingers on electrodes as shown in Figure 6.

![Figure 6. Finger Placement using AliveCor Monitor](image)

Using the index fingers limits the ECG to taking 1-lead measurements, making it difficult to verify the cause of any abnormalities that may appear in the ECG. Variable finger placement on the device can also increase variability in the measurements and makes it hard to compare results from different times. Additionally, each of these products requires the user to initiate a recording, whereas classical Holter monitors consistently record data.

The form factor, interface, and supported platforms of these three products define the user’s experience. The HeartCheck PEN is a standalone handheld device that must be connected to a Windows XP/7 PC to download and view the measurements. This is very inconvenient for the user in two ways: the user must carry a unique device with them at all times and the user must wait until they are with their computer to view the information that was recorded. Being required to carry a separate device makes the user reluctant to use the product and can result in missed recordings if the product is forgotten or left at home.

Additionally, if a very serious condition is recorded by the device while away from a computer then the event may go unnoticed until the user gets back to their computer and remembers to upload
the data from the product. Both AliveCor and ECG Check have solved these problems by creating
products that act as cases for smartphones and can sync directly with the smartphone. By
incorporating the device into a case, the user only has to carry around their smartphone. The case
does make the phone bulkier than most cases and could become an annoyance to the user. The
products also sync with the phones so that recordings can be viewed using the user’s phone and
uploaded when the user has internet connection.

All three products have FDA approval for over-the-counter (OTC) sales as well as prescription (Rx)
sales. For ECG Check, some features of the product, such as direct view and analysis of the signal,
are limited to physicians and prescribed sales. These restrictions may be a requirement of the FDA
approval since separate approvals are required for each sales case.

The prices of these devices range from approximately $100 to $300. This is a fine price point when
targeting prescription sales which are likely subsidized by health insurance companies; however,
when targeting over-the-counter sales, this price may be expensive enough to deter anyone other
than fitness fanatics from purchasing the product.

The comparison of functions and specifications of classical and modern Holter monitors are shown
in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Classical Holter Monitor</th>
<th>Modern Holter Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong># of Leads</strong></td>
<td>3, 6, 12</td>
<td>1</td>
</tr>
<tr>
<td><strong>Recording Duration</strong></td>
<td>24 hours continuous</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>Buttons, LCD</td>
<td>Smartphone</td>
</tr>
<tr>
<td><strong>Sample Resolution (bits)</strong></td>
<td>8 - 12</td>
<td>12-16</td>
</tr>
<tr>
<td><strong>Sample Rate (sps)</strong></td>
<td>200 to 1000</td>
<td>300</td>
</tr>
<tr>
<td><strong>Weight (with battery)</strong></td>
<td>10 oz.</td>
<td>0.5-1.5 oz.</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>$300 to $10,000</td>
<td>~$100-300</td>
</tr>
</tbody>
</table>
1.5 Markets of Interest

The development of these solutions is shaped by the healthcare and consumer markets. Physicians could prescribe the use of the device to patients and consumers could buy the device for fitness and health interests.

1.5.1 Market Entry Criteria

Due to the nature of these devices and their impact on user health, there is significant regulation by the FDA and FCC regardless of the intended market. The first step to entering any market is achieving FDA approval which can take several years and significant amounts of money depending on how many studies must be performed and what documentation the FDA determines is necessary. This is required for not only the physical device, but for any other software or algorithms that are released.

1.5.2 Consumer Market

The consumer market for fitness monitors(trackers is currently smaller than the healthcare industry, but it is growing rapidly. According to the Consumer Electronics Association, “Consumer interest in purchasing dedicated wearable fitness devices … quadrupled to 13 percent in 2013, from just three percent in 2012.” And in 2013, “three quarters of online U.S. consumers (75 percent) now say they own a fitness technology product.” [17] The Consumer Electronics Association also conservatively projected that the fitness technology market will surpass $1 billion in revenues in 2014. While only a portion of this market is applicable to heart monitors, the interest in heart monitoring is clear by the number of heart rate monitors available and the increasing number of products adding heart rate measurements as a feature, such as the Apple watch.

The benefit of the consumer market is that it is much easier to enter due to the low number and low popularity of current alternatives. While the three devices mentioned earlier are the main products involved in the consumer ECG monitoring market, really only AliveCor has made any significant impact. AliveCor is a young company (2 years old) that has a design that is very focused at the consumer and is finally beginning to make news with it’s products and service offerings. Since
consumer knowledge of AliveCor is still so low, it would be relatively easy to make a superior product and use the rising popularity of fitness monitoring to attract more customers. After FDA approval, a consumer product could be easily marketed to customers as well as in brick-and-mortar stores.

1.5.3 Healthcare Market
The healthcare market is filled with classical Holter monitors. This is because the current daily-use ECG monitors do not have the necessary capabilities for practical medical use, mainly 3-lead and greater measurements. Competing with classical Holter monitors is also difficult due to the current habitual practices that have been formed by doctors in the industry.

The healthcare based global heart monitoring market is enormous. According to research done by MarketsandMarkets, it was a $18.4 billion industry in 2012 and is estimated to be a $23.3 billion industry in 2017. This is based on an estimated 2.2 million patients using remote monitoring in the US in 2010 and is estimated to reach 6 million patients by 2016 [18]. The current increase in prevalence of heart diseases each year increases the number of patients that would directly benefit from heart monitoring. A superior product that achieves only a mere 10% market share would have $2.3 billion in revenues in 2017. This emphasizes the importance of creating a product that is up to the healthcare industry’s standards for cardiac monitoring.

A downside to the healthcare industry is that the barriers to enter the market are very high. After FDA approval, the product would have to be marketed to healthcare providers as a replacement or alternative to their current monitors. Selling to healthcare providers could easily be accomplished by enlisting the help of a sales organization experienced with medical devices; however, doing so would lower the product’s profit margin. Marketing the product to healthcare providers directly would increase the profit margin of the final product but would initially take much more money and industry connections.
2. METHODS

2.1 Circuit Simulation

The analog electronics were designed and validated using LTSpice, an electronics simulation software tool developed by Linear Technologies. The software provides an accurate time-domain and frequency-domain response of the simulated circuit for determining gain, current draw, and noise characteristics [19]. Spice models of the integrated circuits used in this project were obtained from the associated manufacturers and used in the simulation tool [Appendix A, B].

2.2 PCB Design

The PCB design of this project utilized two ECAD software tools. Initially, EagleCAD Light Edition (for educational use) was used; however, it’s license limited designs to 2-layer PCBs. Subsequent designs used open-source software KiCAD which had no license limitations allowing for 4-layer designs.

2.3 PCB Fabrication and Assembly

Prototyping house, OSH Park, was used for PCB fabrication. They fabricate high quality, lead-free boards in the US [20]. OSH Park is used by professionals and hobbyists to build low-volume, low-cost, high-quality PCBs. All assembly was done by hand using soldering irons, heat guns, and reflow ovens. Solder paste stencils were made by OSH Stencils.

2.4 Improvement Criteria

2.4.1 Measurement Quality

To be an effective solution for identifying early signs of heart conditions as well as fulfilling the healthcare market requirements, the device must provide quality ECG measurements. Measuring one lead does not provide enough information or data about the function of the heart to accurately diagnose heart conditions. For medical conditions that are identified by anomalies in the QRS complex, three leads is the minimum number of leads necessary for a physician or algorithm to make an accurate diagnosis [21, 22]. A 3-lead ECG can indicate other possible medical conditions, but requires a 12-lead ECG follow-up to make the diagnosis. Taking a 3-lead measurement
requires a minimum of 4 electrodes that must be placed on the body. Measurement resolution, sample rate, filtering, and electrode placement have a significant effect on the quality of the measurement [23]. The measurement quality is evaluated in two ways. First empirically, by measuring the frequency-domain transfer function of the electrode front end to validate the signal passed undistorted. Distortion occurs when frequencies pass through filters at different rates. The distortion is simulated using LTSpice and then validated by injecting known signals into the hardware and measuring the distortion. Second, measurement quality is validated by comparing the device measurements to medically acceptable measurements based on noise levels and identifiable physiological markers. Medically acceptable measurements are taken on the same subject using a medical-grade ECG machine, the Philips PageWriter TC70 cardiograph. The sampling rate is verified by measuring the frequency of a trigger output from the microcontroller while it measured the electrode signals.

2.4.2 Data Transfer
The method used to record and display the ECG data is critical to the usefulness of the device. In many cases the patient is required to return the device to the prescribing doctor in order for the data to be reviewed. If the device is used for longer durations, then the doctor may desire to check the measurements periodically before the end of the test period. In order to allow for this flexibility, it is advantageous to create a method to transfer device data without having to bring the device to the doctor. This method could be a wireless or wired connection. A wireless connection would not require the user to physically connect the device to a PC, but would likely require more complicated components and power for the radio. The effectiveness of the data transfer method is measured by the data transfer rate and the level of user interaction. Faster data rates reduce the time required for the device to upload data to the doctor. Certain data transfer methods have a high burst data rate that is only achieved for short periods of time. However, when transferring large sets of data the sustained throughput speed of the transfer method is critical as it determines the total data transfer time.
Less user interaction reduces the possibility of incorrectly handling the data as well as increasing the convenience for the user. User interaction was evaluated based on the number of initial or repeated steps the user must perform to complete data transfer. Initial steps include wirelessly pairing devices or setting up accounts. A repeated step would be starting data transfers manually. As a rule-of-thumb, less than 3 clicks are required for the user to initiate data transfers.

Health data is very sensitive and the user's privacy must be of highest concern. The data transfer method is also evaluated on level of security. Data storage and transfer encryption methods will be compared to the NSA Suite B Cryptography List, a highly trusted list of encryption methods suggested by the NSA to protect Top Secret information [24].

2.4.3 Functional Life
The functional life of the device is dependent on two factors, battery life and device comfort. Improving battery life beyond the current limits of 48 hours gives the device the ability to have a longer functional life. However, if the ergonomics of the device are not improved, then the usable device life will not change. For example, current devices require a doctor to place the electrodes onto the patient's skin and they cannot be removed during the measurement time. For functional durations longer than 48 hours, the patient will likely desire to shower and the adhesive will likely fail. Additionally, for patients to accept longer duration use, the device must be more comfortable to use during their daily activities. As such, the design of the electrode interface with the skin must be improved and the size of the hardware must be decreased once the battery life is improved in order to improve overall functional device life.

2.4.3.1 Comfort
For this project, comfort is defined as minimal interference with usual daily activities of the user. It is measured by the number of daily activities that must be modified to account for using the device. This characteristic is a direct result of device design and is independent of an individual’s preferences. Design decisions about device’s waterproofing, battery life, and data transfer method are most likely to affect showering, dressing, and sleeping.
Waterproofing affects if the device can be washed. It will need to be washed after being worn for more than a day. Battery life and data transfer method determine when the device will be out of service and for what duration. If the user must recharge the device they will likely not be able to wear it. Depending on the method, the user might not be able to use the device during data transfer.

2.4.3.2 Hardware Size
Hardware size was measured by the device weight and physical extrusion beyond normal daily-wear. The largest physical extrusion will be the electronics and be a function of various factors. The complexity of the design will drive the number of components needed on the PCB and the selection of those components will drive the size of each one. However, both of these aspects are limited by the manufacturing abilities used to produce the device. Certain extremely small component packages are unusable without expensive manufacturing equipment available for larger, higher quality manufacturing. Device weight will be largely driven by the size of the PCB and the size of the battery.

2.4.3.3 Battery Life
Battery life is dependent on the electronics and firmware design. The most common trade-off in battery life is power consumption for performance. Battery life was calculated by dividing the battery capacity by the power usage of the device. The power usage of the device was first calculated by the theoretical current draw of the electronics and then measured by instrumenting the device during each of its functional states and then applying P=IV. Once the device is optimized for power consumption, the battery (and associated capacity) can be chosen to achieve the desired battery-life; however, the upper limit on battery life is determined by the physical size and weight of the battery needed.

2.4.4 Device Cost
For insurance companies to more widely administer Holter monitors to patients, they will need to have a positive cost-benefit ratio. Reducing the hardware unit cost of each device improves the likelihood that the device will be adopted. Prototype costs are inevitably higher than the final production cost of the device, so cost was evaluated on the component and assembly cost at the
scale of producing 1000 devices. This results in an optimistic estimate of total device cost since this excludes more complicated costs such as total device assembly, packaging, and support; however, it is still useful for predicting a final sale price. A common rule-of-thumb is that final sale price is 4 to 5 times the material cost of the product.

2.4.5 User Interaction
A successful device must be easy for the patient and doctor to use and comfortable for the patient. Simple and intuitive designs are more likely to be used than complex ones. Ease of use is measured by the number of steps and interfaces that the user and doctor must learn and interact with in order to correctly use the device.

2.4.6 Device Requirements Summary
The prototype device developed in this project is different than a final production device because of different assembly processes and large quantity benefits. The measurable requirements for a production device are summarized in Table 2.

<table>
<thead>
<tr>
<th>Measurement Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Leads</td>
</tr>
<tr>
<td>Sample Resolution</td>
</tr>
<tr>
<td>Sample Rate Per Lead</td>
</tr>
<tr>
<td>Undistorted Signal Spectrum</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer Rate</td>
</tr>
<tr>
<td>Number of User Interactions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Life</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Daily Activities Affected</td>
</tr>
<tr>
<td>Device Size (including batteries)</td>
</tr>
<tr>
<td>Device Weight (including batteries)</td>
</tr>
<tr>
<td>Battery Life</td>
</tr>
<tr>
<td>Device Cost</td>
</tr>
</tbody>
</table>

Table 2. Production Device Requirements
3. RESULTS

3.1 Device

The prototype device design consists of an elastic undershirt with sewn-in non-adhesive electrodes wired to a set of small PCBs and battery at the base of the shirt, as shown in Figure 7. Once the device is programmed with patient information, the patient can wear the shirt. The prototype device is able to record a 3-lead ECG for approximately 6 days without being connected (wired or wireless) to any external device. Lead 1 and Lead 2 are measured directly and Lead 3 is calculated from the data of Leads 1 and 2.

To the patient, the device is single use; however, the lithium-ion battery within the device may be recharged by the doctor allowing for reuse. The measurements are stored on a micro SD-card that can be removed from the device and inserted into a computer. Computer software could be developed to read data from the SD-card and upload it to a database where the patient’s doctor could examine it. Uploading data throughout the 6 days of use would allow a doctor to review some of the data earlier.

Minimal filtering and no analysis is performed on the device, reducing the cost and complexity of the hardware. However, further software could be developed for the database to filter and analyze the data, providing high-level insights such as diagnosis suggestions. Keeping the filters and analysis in software allows for their design to be more flexible and capable.
Figure 7. The final prototype device design – 3-Lead ECG with Electronics and Battery
The four electrodes can be seen embedded within the shirt with wires connecting them to the interconnected PCBs in the bottom right that house the device electronics, as shown in Figure 8.

![Figure 8. Connected PCBs](image)

The electronics are broken down into several components as shown in Figure 9. The analog front-end PCB, shown in Figure 10, that the 4 electrodes plug into contains the analog front-end, which amplifies, filters, and level-shifts the ECG signal. The processed ECG signal is connected to the microcontroller PCB, shown in Figure 11, through the header pins, which also connects power between the two PCBs. On the microcontroller PCB, the signal is sampled by the ADC in a microcontroller and then stored on the micro-SD card.

![Figure 9. Block Diagram of the 3-Lead ECG](image)
3.2 Measurement Quality

Measurement quality is dependent on the electrodes, electrode analog front-end, the microcontroller ADC, and sampling accuracy. The quality of the electrodes provides an upper boundary on the quality of the measurement. Non-adhesive electrodes provide comfort to the wearer — allowing the device to be taken off and put back on by the user without the presence of
a doctor — while sacrificing measurement quality. Without a doctor placing the electrodes, placement variability increases. The device is designed with the electrodes integrated so that they are placed consistently when a user puts the device on; however, due to the elasticity of the wearable material, there is still some variability between uses. Users of differing body size will need differing placement of the electrodes. This can be accommodated by an initial placement of the electrodes by a doctor onto the wearable and by the elasticity of the wearable. As the wearable stretches for a different user the electrodes scale to fit the new body size. Unfortunately, non-adhesive electrodes are inherently noisier than their adhesive counterparts because the measured signal is smaller (due to the body-electrode interface being higher impedance) and more susceptible to motion artifacts [13]. Non-adhesive electrodes with measurement quality comparable to adhesive electrodes are currently being developed but are outside of the scope of this project [12, 25].

3.2.1 Instrumentation Amplifier
The analog front-end consists of several stages with the instrumentation amplifier first, as shown in Figure 12. Two identical front-ends are implemented, one for Lead 1 and one for Lead 2. Lead 3 is calculated afterwards from Lead 1 and Lead 2 data.

![Figure 12. Block diagram of electrode analog front-end](image-url)
The instrumentation amplifier, AD8426, has the necessary low-noise performance and low input bias current. The low input bias current preserves the signal integrity from the high impedance electrode [26]. As the input bias current increases, the voltage amplitude of the signal into the amplifier is reduced because the voltage is dropped over the internal impedance of the electrode, Ze, as shown in the simplified electrode and amplifier model in Figure 13.

![Figure 13. Electrode-Amplifier Model](image)

The input noise voltage is the noise that is generated by the amplifier at the inputs. The AD8426 has a 27nA maximum input bias current and a 27nV/√Hz maximum input voltage noise. The gain of the AD8426 can be set anywhere between 1 and 1000V/V. The selected gain, 20V/V, provides significant amplification in the first stage to improve the noise figure, but not enough to cause the gain bandwidth of the amplifier to filter the signal.

### 3.2.2 Hardware Filters

The differential signal is filtered from 0.5Hz to 200Hz because the frequency range for ECG signals are approximately 0.5Hz to 150Hz; however, a wider range rejects enough noise to prevent saturation of the electronics but prevents loss of any signal data. Further digital signal processing (DSP) can be performed to increase signal clarity further, which is discussed later.

The first filtering stage in Figure 12 is a feedback loop that uses the reference pin of the instrumentation amplifier to create a high-pass filter and remove the output offset of the instrumentation amplifier shown in Figure 14.
This high-pass topology has the advantage over other topologies in that it does not need any components in series with the forward signal path [27]. Similar to the effect of input bias current, additional components in series will attenuate the electrode signal. The filter works by putting the output of the instrumentation amplifier through and inverting integrator which amplifies and inverts the DC component. This is then fed back into the reference pin of the instrumentation amplifier shifting the DC offset at the output towards zero. This negative feedback technique is commonly referred to as servo feedback [27]. This system can be analyzed as a negative feedback loop, as shown in Figure 15.

Figure 14. Instrumentation Amplifier and High-Pass Feedback Loop

Figure 15. Negative Feedback Loop in the Frequency Domain [15]
The forward gain, \( G(s) \), is the gain of the instrumentation amplifier, \( G_{IA} \), and the feedback gain, \( H(s) \), is the negative of the gain of the integrator. The gain of an integrator in the frequency domain (s-plane) is \( 1/sRC \) [15]. The transfer function of a negative feedback loop of this kind is as shown in Equation 1 [15].

\[
G = G_{IA}; \quad H = \frac{1}{sRC}
\]

\[
TF(s) = \frac{G(s)}{1 + G(s)H(s)}
\]

(Equation 1)

\[
TF(s) = \frac{G_{IA}}{1 + \frac{G_{IA}}{sRC}} = \frac{sRC + G_{IA}}{sRC + G_{IA}} = \frac{sRC + G_{IA}}{\frac{1}{sRC} + G_{IA}}
\]

(Equation 2)

\[
G_{DC} = \lim_{s \to 0} TF(s) = \frac{G_{IA}}{0 + G_{IA}} = 0
\]

(Equation 3)

\[
G_{HF} = \lim_{s \to \infty} TF(s) = \frac{sRC + G_{IA}}{sRC} = G_{IA}
\]

(Equation 4)

The corner frequency of this stage’s transfer function in Equation 2 is \( 1/RC \). The implementation uses \( R = 200k\Omega \) and \( C = 20uF \), so the expected cutoff frequency is 0.25Hz. The high-pass nature of the filter design is verified by Equations 3 and 4; the low frequency (DC) gain, \( G_{DC} = 0 \) and the high frequency gain, \( G_{HF} = G_{IA} \).

The second stage low-pass filter in Figure 16 uses a second order Sallen-Key topology, for its sharper frequency response roll-off, shown in Figure 17 [28].

Figure 16. Front-End Second Stage Sallen-Key Filter
The filter’s designed corner frequency and quality factor are $f_c = 183.7\,\text{Hz}$, and $Q = 0.8$, as shown in Figure 17. The low Q value prevents response ripple that could distort the signal by having varying gains and high group delay variation. The ripple is less than $0.25\,\text{dB}$. The maximum group delay measures $1.6\,\text{ms}$ in simulation. Since this is almost $1/100^{th}$ of the feature size of the ECG waveform, it should not cause any noticeable distortion.

![Figure 17. Sallen-Key Filter Design Frequency Response](image)

Implementing a higher gain in the first stage helps reduce the noise figure of the overall system and reduces the impact of noise in further stages. As such, the second, third, and fourth stage amplifiers focus on low input/output offset and low power consumption. Low input/output offset is a major design consideration because the system has a large gain, greater than $10\,\text{V/V}$. All input offsets are amplified to the next stage and can cause the signal offset to reach the voltage limits of the amplifier, truncating the signal. The signal offset of an amplifier can be modeled as $V_{out} = G \times V_{in} + V_{offset}$ where each voltage is the DC component. For example, assume the output offset of an amplifier is $0.2\,\text{V}$. If the following stage has a gain of 25, with its own offset of $0.2\,\text{V}$ then the output offset would be $25 \times 0.2\,\text{V} + 0.2 = 5.2\,\text{V}$. For a $3.3\,\text{V}$ system, as in this case, the signal would be truncated at $3.3\,\text{V}$ and all of the signal content would be lost. Aside from choosing amplifiers with
low output offset, this effect is mitigated by distributing the gain across several stages and allocating the fair portion of the gain to the last high-pass level-shifter stage. The high-pass nature of the final stage attenuates all offset from the previous stages so the output offset is only the innate offset of that stage.

The third gain stage is shown in Figure 18.

![Figure 18. Front-End Third Stage Amplifier](image)

The fourth amplification stage in Figure 12 sets the DC offset of the signal to half supply. This gives the most dynamic range for the microcontroller’s ADC to sample the signal, as shown in Figure 19.

![Figure 19. Front-End Fourth Stage High-Pass Amplifier](image)
Lastly, since the amplification stages uses dual supplies, a schottky diode clamp is used to protect the microcontroller from a negative voltage below the rated minimum pin voltage, as shown in Figure 20.

![Schottky Diode Clamp](image)

**Figure 20. Schottky Diode Clamp**

The selected diode has a typical forward voltage of 0.28V in this configuration which is less than the pin tolerance of 0.3V [29].

The LTSpice simulation of the entire analog front-end is shown in Appendix A. The results of the AC simulation is the frequency response shown in Figure 21. The maximum gain is 54.2dB, with 3dB cutoff frequencies of 0.998Hz and 200.9Hz.

![Simulated Frequency Response of the Analog Front-End](image)

**Figure 21. Simulated Frequency Response of the Analog Front-End**
Using the setup shown in Figure 22 the frequency response of the assembled front-end was measured. The waveform generator output was a 4mVp-p sinusoid to prevent saturation of the output of the analog front-end. The signal was fed through a 1MΩ resistor in parallel with a 10nF capacitor to simulate the impedance of dry electrodes. The results are shown in Figure 23.

Figure 22. Front-End Measurement Test Setup

Figure 23. Electrode Front-End Frequency Response
The measured maximum gain is 53.8dB and the 3dB cutoff frequencies of the front-end are 1.1Hz and 205.0Hz. The variation between measured and simulated values, shown in Table 3, are likely due to passive component parameter variations and are still within acceptable device tolerances.

### Table 3. Front-End Frequency Response Results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Simulated Value</th>
<th>Measured Value</th>
<th>Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passband Gain (dB)</td>
<td>54.2</td>
<td>53.8</td>
<td>0.4dB</td>
</tr>
<tr>
<td>Lower Cutoff Frequency (Hz)</td>
<td>0.998</td>
<td>1.1</td>
<td>10.2%</td>
</tr>
<tr>
<td>Upper Cutoff Frequency (Hz)</td>
<td>200.9</td>
<td>205.0</td>
<td>2.0%</td>
</tr>
</tbody>
</table>

**3.2.3 Microcontroller Sampling**

The device design uses the Freescale MK20DX256VLH7 microcontroller (on a Teensy 3.2 development board) for data sampling and transfer. The high clock frequency, 72MHz, allows for the on-board 12-bit ADC to be sampled at 500Hz. The data must be sampled at a rate that is greater than 2 times the highest frequency component [15]. Since the filtering stops at 200Hz, the sample rate must be greater than 400Hz. For a more visually accurate time-domain representation of the signal, the sample rate must be even faster. 500Hz was initially chose however; during testing it was found to be around the upper limit due to processing limitations of the microcontroller.

The integrated ADC is configured to use 12 bits, 0 averages, high-speed sampling, and high-speed conversion. The ADC is able to be configured for less than 12 bits; however, using all 12 bits allows for the greatest accuracy in reproducing the ECG waveform. Averaging is turned off to reduce the sampling time. Increasing the number of averages to 4 could increase the accuracy of the measurement, but could also potentially have an undesired low-pass filtering effect. High-speed sampling and high-speed conversion are also used to reduce sampling time. High-speed sampling requires a low-impedance signal like the output of the analog front-end.

To validate the accuracy of setting the sampling frequency, sampling period is measured over 4 seconds. The microcontroller toggled a GPIO pin at the start of each ADC sample and the time between each toggle, the sampling period, was measured. A histogram of the sampling period measurements is shown in Figure 24. The maximum, minimum, and mean sampling periods are
2.024ms, 2.008ms, and 2.014ms respectively. And the resulting minimum, maximum, and mean sampling frequencies are 494.1Hz, 498.0Hz, and 496.5Hz, respectively. The mean sampling frequency is off from the expected 500Hz by 0.7% and the minimum and maximum deviate from the mean by 0.5% and 0.3%, respectively.

![Sampling Period Variation Histogram](image)

**Figure 24. Sampling Period Variation Histogram**

### 3.2.4 Software Filters
Once the data is transferred from the device to a computer, a MATLAB script is used to demonstrate the capabilities of DSP to further reduce the signal spectrum as well as create narrowband filters to reject noise from narrowband sources such as 60Hz from grid power [30] [Appendix G]. Designing extremely sharp filters in electronics results in distortion; however, software implemented filters can have infinite sharpness without introducing distortion, called brick-wall filters [31]. Brick-wall filters have complete transmission in the pass-band and complete attenuation in the stop-band. In MATLAB, one filter removes higher frequency noise greater than 150Hz and another filter eliminates 60Hz power noise. This filter can be configurable to remove all signal from that
frequency or to equalize it to the average level of the rest of the signal. The results of using both filters are shown in Figure 25.

![Figure 25. Results of MATLAB Filtering in (a) Time Domain and (b) Frequency Domain](image)

By removing the very low frequency signals, the time domain data is shifted to zero offset and has a flat baseline. Removing the 60Hz and high frequency noise makes the signal clearer.

3.2.5 Waveform Feature Comparison
ECG measurements taken by the device are compared against measurements taken by a Philips PageWriter TC70 cardiograph on the same subject. Visual inspection of these two measurements, shown in Figure 26, shows that the device’s measurements are very close in quality to the TC70. More specifically, the TC70 identified “early repolarization” which is linked to the elevated S-T segment in Figure 26b, see Note 5. This feature which would be used as a marker for diagnosis can also be identified in the measurement taken by this project’s device, in Figure 26a. One negative feature is that the P-wave is not identifiable in the device’s measurement of Lead 1 (see Note 1); however, it is visible in the TC70’s measurement, although it is small. In Lead 2, the P-wave is identifiable in both measurements (see Note 2). Additionally, there are artifacts in the T-wave that do not appear in the TC70 measurement (Note 4). Lastly, there is a notch in the QRS complex that is visible in both measurements (Note 3).
Figure 26. ECG waveforms recorded by (a) the device and (b) a Philips PageWriter TC70

3.3 Data Transfer

3.3.1 Transfer Method Assessment
The viability of various methods of data transfer — WiFi, BLE, BT-EDR, USB, Micro-SD Card — were assessed. Data transfer must be optimized for user experience; however, it is limited by the size of data needing to be transferred. The device records 2 channels of data at 500 samples per second and each measurement is taken by a 12-bit ADC which requires 2 bytes (16 bits). The data creation rate is 2 * 500sps * 16bits = 16kbps = 2kBps. The data throughput of the data transfer method must exceed this value to keep up with new measurements. Additionally, this calculation assumes that only the data and no metadata, such as timestamps, device ID, and patient ID, are sent which slightly increases the required data rate.

3.3.2 WiFi
WiFi would provide sufficient data transfer rates; however, WiFi modules consume significant amounts of power for extended durations. WiFi functions using a router and device network where there is one router connection per device. The device is required to connect to a router to transmit
data (such as in a home network) or to act as router that a secondary device would connect to (such as a smart phone). The first scenario requires the device to store data until it is in a known WiFi network. The second scenario requires the smart phone to connect to the device over WiFi and then connect to a different WiFi network to upload data. Finally, WiFi can be challenging for users to setup correctly.

3.3.3 BLE
Bluetooth Low Energy (BLE) is by design very low power and cheap to implement. The main issue is the data throughput. The data transmitted through BLE is sent at 1MB/s; however, data cannot be sent continuously [32]. Data can only be sent once during each connection followed by a connection interval where both devices sleep. BLE connection interval defaults to 30ms with a minimum of 20ms, although it is limited to 30ms for many devices. Each connection interval allows 1 set of 64 bytes to be sent. This means that the throughput of BLE defaults to 64bytes/30ms = 2.13kBps and is limited to 64/20ms = 3.2kBps. This is barely sufficient to keep up with the data creation rate. In this case, the device would need to continuously transmit data to the computer because if it missed several packets, it would fall behind and would take a long time to catch up to the current measurement.

BLE connections can be easily interrupted by losing line of sight, so it is likely that a user would lose connection if their phone was left in another room in their home [33]. If the BLE connection was lost, the device would need internal data storage to save all the measurements taken while out of range and once back in range, it would take a while to catch up to the current measurement. For example, if a device was out of range for 10 minutes, it would generate 1,200kB of measurements. Assuming a best case of 3.2kBps data rate, only 1.2kBps would be available to send that data because the other 2kBps would be used to send current data. That means it would take 1,200kB/1.2kBps = 1000s =16.7 minutes to catch up.
3.3.4 BT-EDR
BT-EDR was investigated as a potential data transfer method because of its increased data rate compared to BLE; however, it does require more power. A practical data rate for EDR in a system is 1MB/s. This is more than enough to stream the data from the device to a smartphone. Streaming the data would put a significant power drain on the device and the user’s phone. This method is used for Bluetooth enabled speakers, and the large batteries they typically use only last for approximately 5 hours. Even transmitting for a fraction of the time, the battery life of the device would not be acceptable. Additionally, it cannot be expected that the user has their smartphone on them at all times. During a 1-hour period the device will create 2kBps * 60 minutes/second * 60 minutes/hour * 1 hour = 7.2MB. This would take 7.2 seconds to transfer to the user’s phone, making BT-EDR viable as a data transfer method only if the device had internal storage. This approach would still require the user to set up the Bluetooth connection.

3.3.5 USB
USB protocols, 1.1 full-speed, 2.0 high-speed, and 3.0 super-speed could be used for the data transfer but would require the device to be tethered to a laptop or other internet connected device during the transfer period. Realistically, a person would then upload data once a day. Over 24 hours the device would have to store 172.8MB of data. For USB full-speed, at a data rate of 12Mbps, it would take around 1.9 minutes to upload. For USB high-speed, at a data rate of 480Mbps, it would take less than 3 seconds to upload. USB super-speed would make transfer time negligible. Unfortunately, high-speed and super-speed are not commonly supported on low-cost, low-power microcontrollers. USB full-speed would result in a poor user experience, where the user must remain near the upload device during the ~2 minute data transfer. Another option would be for the user to remove the wearable, which would also be a poor experience.

3.3.6 Micro SD Card (Chosen Solution)
Each of the data transfer method options require internal storage on the device as primary storage or backup for loss of wireless communication. Using removable data storage such as a Micro-SD card would result in a fast transfer rate and remove the need for the user to wait during the data
transfer. SD cards use a serial data protocol of either the proprietary SD protocol or Serial Peripheral Interface (SPI). SPI can write data to a FAT formatted SD card at approximately 1Mbps depending on SPI settings and SD card specifications [34]. This is enough to keep up with the measurements however; it is dependent on the SD card itself. Different cards can support various read and write speeds, so it is necessary to pick one with high throughput. To upload the data, the user would remove the SD card from the device and insert it into a laptop. The upload could take up to several minutes, but the user would not be required to wait during that time. For the data transfer rate of 2kBps, a common 8GB micro-SD card could hold 4 million seconds (approximately 46 days) of data.

The schematic of the SPI interface implementation is shown in Figure 27. Writing to the FAT file system requires a significant amount of overhead time to set up during a write cycle. To make the write process power efficient, the microcontroller buffers measurement data to the local RAM and once the memory is full it writes the entire block to the SD card [Appendix F].

![Figure 27. SD Card wiring diagram](image)

### 3.3.7 Data Encryption

Data encryption is implemented using an AES encryption library for microprocessors written in C. The implementation uses chunk-by-chunk encryption with a 256-bit key that would be programmed into the device and saved in a healthcare database at the time of patient registration. The
initialization vector (iv) used for encryption is the first date code associated with the block of data. Data is buffered in the RAM of the microcontroller until a set number of bytes have been saved at which point the data is encrypted and saved onto the SD card. For maximum power efficiency, the largest block size is used, which is limited only by the microcontroller’s RAM size.

3.4 Functional Life

3.4.1 Battery Life
The final device can record ECG comfortably for almost one week. The battery life is the limiting factor of the device’s functional life when compared to data storage capacity. The power consumption of the device was focused in two areas, the analog-front end and microcontroller. The analog-front end draws a steady state 1.8mA. This is due to the quiescent current draw of each IC and the passive components associated with them.

The microcontroller, on the other hand, draws a variable amount of power depending on the operation it is performing. The microcontroller’s operational flow is shown in Figure 28.

![Figure 28. Flow Diagram of Microcontroller Operation](image)

The microcontroller starts in “Setup” by configuring its GPIO pins, real-time clock (RTC), AES encryption key, ADC configuration, SPI pins, low-power interrupt timer, sleep mode, and initializing all global variables such as the ADC reference, measurement data buffer, AES reference and SD-
card file object. After setup is complete, the microcontroller enters low-leakage stop (LLS) sleep mode. In this sleep mode, almost all of the microcontroller circuitry is powered down except for the low-power timer and real-time clock. The real-time clock keeps the time for time-stamping measurements and the low-power timer is used to wake up the microcontroller at the correct time to take measurements. When the low-power timer triggers, the microcontroller wakes up, initiates the ADCs to take measurements from the electrode analog front-end, saves the data to local memory, and then goes back to sleep. This cycle occurs until the local memory is almost half-full at which time the microcontroller, after finishing an ADC measurement, makes a copy of the measurement data and then begins encrypting the data, followed by storing the data on the SD card. Once both steps are done, the microcontroller goes back to sleep. A copy of the measurement data is made so that the microcontroller can keep taking measurements during encryption and storing without overwriting the previous data.

Sampling the ADC takes a tiny amount of power compared to encrypting and storing the data but is done for a much longer period of time. The weighted average current draw is measured in two stages. First, the length of time the device spends in each state is measured by capturing the voltage drop across a current sense resistor, as shown in Figure 29 and 30.
Second, the current draw in each state is measured using a current meter while specialized microcontroller code forces a particular state to be active. The total microcontroller current draw is
calculated as a weighted average of the current draw in each state, as shown in Table 4 and Equations 5 and 6.

Table 4. Weighted Average Microcontroller Current Draw by State

<table>
<thead>
<tr>
<th>Microcontroller State</th>
<th>Time in State (ms)</th>
<th>Current Draw in State (mA)</th>
<th>Weighted Average Current Draw (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling ADC/Sleep</td>
<td>4021.45</td>
<td>1.2</td>
<td>1.1</td>
</tr>
<tr>
<td>Encrypting Data</td>
<td>282.75</td>
<td>24.5</td>
<td>1.6</td>
</tr>
<tr>
<td>Storing Data on SD Card</td>
<td>106.52</td>
<td>68.5</td>
<td>1.7</td>
</tr>
<tr>
<td>Total</td>
<td>4410.72</td>
<td></td>
<td>4.3</td>
</tr>
</tbody>
</table>

Microprocessor Current Draw = \( I_{up} \)

\[
I_{up} = \frac{I_1 \times t_1 + I_2 \times t_2 + I_3 \times t_3}{t_1 + t_2 + t_3} 
\]

\[
I_{up} = \frac{1.2mA \times 4021.45 ms + 24.5 mA \times 282.75 ms + 68.5 mA \times 106.52 ms}{4021.45 ms + 282.75 ms + 106.52 ms} = 4.32 mA \quad \text{(Equation 5)}
\]

\[
l_{device} = I_{up} + I_{front-end} = 4.319 mA + 1.8 mA = 6.12 mA \quad \text{(Equation 6)}
\]

The microcontroller uses significantly more power when running at full clock speed; however, this reduces the amount of time required for the microcontroller to be active before it returns to deep sleep. Other lower-power microcontrollers with slower clock frequencies are unable to sample the ADC at 500Hz while encrypting and saving the data to the SD card. The total current draw of the microcontroller is calculated as the sum of each of the states’ average current draw, resulting in 4.3mA. The total current draw of the device which includes the electrode analog front-end and microcontroller is 6.1mA, which can be powered by the selected 850mAh battery for 139.3 hours (5.8 days).

3.4.2 Hardware Size

The final hardware size is 55mm x 37mm x 27mm, which is determined by the final layout size of the PCB and the size of the battery. Surface mount components were used to keep PCB size down, but lab equipment limitations prevented further miniaturization of the PCB. Component size, proximity, and top/bottom side placement was limited by the capabilities of hand placement. Additionally, components were only placed on one side of the PCB due to reflow oven limitations.
The device thickness is limited by the board-to-board connectors used to integrate the microcontroller with the analog front-end.

The final device weighs 39.42g, with the PCB being 23.23g and the battery being 16.19g. The battery is very simple with no extra electronics, so it’s weight is wholly dependent on the density of lithium ion batteries. The PCB weight is a combination of the PCB size and the component weight, although largely driven by the weight of the PCB and connectors.

The battery is a significant limiting factor of the design since lithium battery density is relatively fixed by currently commercialized technologies [35]. The large space and weight taken by the battery is unavoidable; however, a smaller battery could be used with the same battery life by reducing the power consumption of the device or by reducing functionality.

3.4.3 Device Comfort
Device comfort is achieved by using non-adhesive electrodes and an elastic shirt to hold the electrodes and electronics. The electrodes were selected from current commercially available electrodes. The selected electrodes are Florida Instruments TDE-200. These electrodes are advertised as being usable without conductive paste; however, after testing, they are found to need conductive paste to take consistent measurements. Paste application is a detriment to the comfort and convenience of the device however; the wearer only needs to apply paste in the general region of the electrode and the wearable places the electrode correctly. Several non-adhesive, dry electrodes that are being developed could improve this issue. By making the electrodes active, researchers have shown that they can produce comparable results to wet electrodes [36, 37]. These electrodes could be integrated once they reach commercialization.

3.5 Device Cost
The raw materials used to assemble the device cost a total of $73.38 at quantities of 1000, based on the Bill of Materials (BOM) in Appendix E. Production quantity part prices were approximated using various methods. Battery price was determined by identifying prices for rechargeable lithium ion batteries of approx. 850mAh capacity in small quantities. This price is likely an overestimate.
Wearable material price was determined by comparing prices for bulk quantity compression shirts. Online retailers such as Alibaba and Wish sell generic brand compression shirts for $2-$8 with minimum quantities of 10 pieces [38]. It was approximated that an order of 1000 pieces would receive the lower end of the cost breakdown and still be quality material.

Electrodes and electrode cable prices were determined using bulk prices from the parts used in the prototype and adding some extra cost for custom parts. In bulk the prototype electrodes cost $0.50; however, it was shown that these are not optimal for production [39]. Using a custom electrode would significantly improve device performance and comfort so this custom price was used.

The prototype cables cost ~$7 in medium quantities; however, it is assumed that the price will drop more in larger quantities and with unneeded parts removed as a custom build in production [40].

The PCB fabrication price was determined by using an estimate of $0.50 per square inch of board area for a 4-layer PCB. PCB assembly component prices were determined by either manufacturer or distributor pricing for quantities as close to 1000 pieces as available.

### 3.5.1 High-Cost Part Analysis

The highest part costs are the electrodes, battery, and microcontroller. The microcontroller is inevitably a higher cost item because it must be both high-performance and low-power, specifications which are trade-offs and require more complexity to achieve. Battery cost is limited by the required capacity. A lower battery life requirement would allow for a smaller, cheaper battery. Electrode and electrode cable are the most expensive parts of the device cost and significantly limit how cheap the device can be because the electrode quality determines the best-case measurement quality. Active and non-adhesive electrodes are significantly less common and more complex to design than their adhesive counterparts making them more expensive [13, 37].

### 3.5.2 Additional Non-BOM Costs

The BOM does not include various costs that would be included in a final production product, such as the costs of PCB assembly, wearable/electrode assembly, and device enclosure. These would
likely add $5-$10 each, based on similar assemblies and injection molded enclosures. Including these extra costs (using an average cost of $7.5 each), the total cost of the device would be ~$100. Assuming a 200% markup to account for unknown production expenses, non-recurring expenses (NRE), and profit, the final device price would be approximately $300.

3.6 User Interaction

The device is very easy for a new user to get accustomed to using. The user only has to interact with the hardware at two times: when putting on the wearable and when uploading data. When putting on the wearable, the user does not have to specifically align the electrodes. In the current form, the user must apply conductive paste in the general region of the electrodes. When the user is going to upload data to their doctor, they need to remove the SD card and place it into their computer and then put it back into the device once the upload is complete.
4. DISCUSSION

4.1 Overview of improvements

The device successfully meets expectations for most of its requirements as summarized in Table 5.

<table>
<thead>
<tr>
<th>Table 5. Device Requirements vs. Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement Quality</td>
</tr>
<tr>
<td>Number of Leads</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>Sample Resolution</td>
</tr>
<tr>
<td>Sample Rate</td>
</tr>
<tr>
<td>Undistorted Signal Spectrum</td>
</tr>
<tr>
<td>Data Transfer</td>
</tr>
<tr>
<td>Transfer Rate</td>
</tr>
<tr>
<td>Number of User Interactions</td>
</tr>
<tr>
<td>Functional Life</td>
</tr>
<tr>
<td>Number of Daily Activities Affected</td>
</tr>
<tr>
<td>Device Size (including batteries)</td>
</tr>
<tr>
<td>Device Weight (including batteries)</td>
</tr>
<tr>
<td>Battery Life</td>
</tr>
<tr>
<td>Production Device Cost</td>
</tr>
</tbody>
</table>

*For Prototype Device (Would be improved on Production Device.)

The device meets all requirements for measurement quality except one. The number of leads and sample resolution were rigid requirements that drove the design. The device’s sample rate of 494 – 499 sps falls within the required range. The undistorted signal spectrum is slightly shifted when compared to the initial requirement. The high end of 205Hz is acceptable because it is 2.5% greater than the expected value of 200Hz. The low end of 1.1Hz is unacceptable because it is 120% greater than the target value of 0.5Hz.
The device meets all requirements for data transfer. The micro-SD card has a maximum data throughput of 1MB/s which is much greater than the data creation rate of 16kbps. The data transfer implementation also requires 3 steps for the user to upload data.

The device meets some of the requirements for functional life. The device affects the daily activity of dressing by requiring the user to put on the device and apply conductive paste. The device size does not meet initial requirements. The two PCBs stack is over 3 times larger than the volume requirement and no side is smaller than 0.5in. For the prototype device, PCB miniaturization was limited by the assembly techniques; however, this could be improved for a production design with improved facilities and various design changes. The electronics weight is 79% of the initial weight requirement. The device’s battery life does not meet the initial requirement of greater than 1 week; however, it reaches 81% of that goal at 5.7 days. Battery life is significantly reduced by the power consumption of the microcontroller during data encryption and storage. Aside from the microcontroller, the power consumption of the device is limited by the instrumentation amplifier.

Finally, device meets the requirement for cost. The approximate device production cost is exactly the required maximum of $100.

4.2 Future Work

The device developed for this thesis can be used as a platform to investigate further technical, functional, and production improvements that are complicated and involved enough to constitute additional projects.

4.2.1 Production Quality PCB Design and Assembly
The PCB assembly would be improved to production quality by integrating all components on to one PCB with components on both sides and using a pick and place machine for assembly. Utilizing machine assembly would allow for smaller package PCB components to be used as well as allowing components to be placed on both sides of the board. This would allow consolidating both PCBs into one and reducing the size of that PCB by a third, reducing the device size in half and reducing the weight by about one third.
4.2.2 Improved Encryption and Data Transfer
The current encryption implementation uses a library that is not optimized for the selected microcontroller. The battery power required for encryption could be slightly reduced by designing an AES encryption implementation tailored to the specific microcontroller such as by utilizing 32-bit operations. Another option for significantly reducing power consumption would be selecting a new microcontroller with a hardware AES accelerator. It has been shown that hardware-accelerated encryption can reduce encryption time and thus power consumption by 50-80% [41, 42]. Additionally, using non-removable flash memory as data storage could significantly reduce power consumption by reducing write times with direct memory access (DMA). However, a data upload scheme different than what is implemented would be required, such as Bluetooth EDR or high-speed USB. The data encryption and storage could be optimized further by customizing C datatypes that would better fit the data. Measurement data was stored using 16-bit variables even though the data only uses 12 bits. By customizing the variables or storage method, those extra 4 bits could be used. This would reduce the amount of total data that is processed and stored.

4.2.3 Non-adhesive Active Electrodes
Dry, active electrodes that have comparable measurement quality to wet electrodes are in active research and development [12, 13, 25]. Integration of these new types of electrodes with the current system could significantly improve the measurement quality.

4.2.4 Electrode-Wearable Integration
The wearable must be robust enough to withstand daily use without degrading measurement quality. The electrodes and wiring must be integrated so that they remain in the correct location on the patient throughout constant use. It may also be useful for the design to allow doctors to chose the location of the electrodes on the patient and easily integrate them after production.
REFERENCES


[35] M. M. Thackeray, C. Wolverton, and E. D. Isaacs, “Electrical energy storage for transportation—approaching the limits of, and going beyond, lithium-ion batteries.”
B. Spice Models

B.1 AD8426

* AD8426 SPICE Macro-model
* Description: Amplifier
* Generic Desc: Low Power, R-R Out, Wide Supply Range Dual Instrumentation Amplifier
* Developed by: ADI - LPG
*
* Revision History:
* C(09/2009) - PRB (Previous Rev)
* D(06/2014) - SH (Updated header)
* Copyright 2009, 2014 by Analog Devices
*
* Refer to http://www.analog.com/Analog_Root/static/techSupport/designTools/spiceModels/license/spice_general.html for License Statement. Use of this model
* indicates your acceptance of the terms and provisions in the License Statement.
*
* BEGIN Notes:
*
* Not Modeled:
* Temperature effects
* PSRR
*
* Parameters modeled include:
* Output swing vs Common-mode Voltage and Load
* Supply current vs power supplies
* Gain error, Vos, Ibias
* Noise
* Bandwidth
* Slew rate
* CMRR vs frequency
* Small signal pulse response
*
*
* END Notes
*
* Node assignments
* inverting input
* | RG
* | | RG
* | | | non_inverting input
* | | | | negative supply
* | | | | | ref
* | | | | | | output
* | | | | | | | | positive supply
*
.SUBCKT AD8426 IN- RG- RG+ IN+ -Vs REF VOUT +Vs
** INPUT STAGE
R1 N009 N008 50E3
R2 N008 Inverting_Out 50E3
R3 N013 noninverting_out 50000
R4 REF N013 50k
R5 RG- N003 24700
R6 RG+ N012 24724
D3 N003 P001 D
D4 P002 N003 D
V3 P002 VNEGx 0.84
V4 VPOsx P001 .61
D5 N012 P003 D
D6 P004 N012 D
V5 P004 VNEGx 0.84
V6 VPOsx P003 .61
D7 N005 P005 D
D8 P006 N005 D
V7 P006 VNEGx 0
V8 VPOsx P005 0
D9 N019 P007 D
D10 P008 N019 D
V9 P008 VNEGx 0
V10 VPOsx P007 0
D11 N009 P009 D
D12 P010 N009 D
V11 P010 N016 0.750
V12 N010 P009 0.83
D13 REF P011 D
D14 P012 REF D
V13 P012 VNEGx .3
V14 VPOsx P011 .3
D15 N013 P013 D
D16 P014 N013 D
V15 P014 VNEGx 0.6
V16 VPOsx P013 0.6
E4 Inverting_Out 0 N003 0 1
E5 noninverting_out 0 N012 0 1
Q1 Inv_Fdbk N002 RG- 0 PNP
Q2 Pos_Fdbk N015 RG+ 0 PNP
V1 VBIAS -Vs -10
I1 Pos_Fdbk VBIAS 2E-6
I2 Inv_Fdbk VBIAS 2E-6

C1 N003 Inv_Fdbk 4.035e-12
C2 N012 Pos_Fdbk 4.0e-12
E8 N002 0 N005 0 1
E9 N015 0 N019 0 1
VOSI_Neg N004 IN- 25E-6
VOSI_Pos IN+ N017 24E-6
VOSO VOUT N011 300E-6
C3 RG- 0 .242e-12
C4 RG+ 0 .1635e-12
I23 IN- 0 -22.3E-9
I24 IN+ 0 -22E-9
G1 0 IN+ N020 N021 .7e-9
R13 IN+ N020 10e9
R14 N020 IN- 10e9
R15 +Vs N021 10e9
R16 N021 -Vs 10e9
G2 0 IN- N020 N021 .7e-9
E10 VPOSx 0 +Vs 0 1
I3 +Vs -Vs 300E-6
G3 +Vs -Vs +Vs -Vs 1e-6
E11 VNEGx 0 -Vs 0 1
R17 VBIAS Inv_Fdbk 10e9
R18 Pos_Fdbk VBIAS 10e9
H3 N006 N004 V24 14
V24 N001 0 0
R19 N001 0 .0166
H4 N011 N009 V25 100
V25 N007 0 0
R20 N007 0 .0166
H5 N018 N017 V26 14
V26 N014 0 0
R21 N014 0 .0166
G4 0 N005 N006 N005 1E-3
G5 0 N019 N018 N019 1E-3
G6 0 N003 VBIAS Inv_Fdbk 1
G7 0 N012 VBIAS Pos_Fdbk 1
G8 0 N009 N013 N008 1
R10 N005 0 10e9
R7 N003 0 10E9
R11 N019 0 10E9
R8 N012 0 10E9
R9 N009 0 10E9

H1 VPOSx N010 POLY(1) VOSO 0 0 8000
H2 N016 VNEGx POLY(1) VOSO 0 0 8000

* MODELS USED
*
.model D D
.model PNP PNP (BF=10E5 VAF=20000)
.ENDS AD8426
*$
D. PCB Layout

D.1 Analog Front-End PCB

D.1.1 Top Layer

D.1.2 Inner Layer 1
D.2 Microcontroller and Micro-SD Card PCB

D.2.1 Top Layer

D.2.2 Bottom Layer
## E. Bill of Materials (BOM)

<table>
<thead>
<tr>
<th>Description</th>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Quantity</th>
<th>Price @ 1000</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery</td>
<td></td>
<td></td>
<td>1</td>
<td>$5.00</td>
<td>$5.00</td>
</tr>
<tr>
<td>Wearable Material</td>
<td></td>
<td></td>
<td>1</td>
<td>$2.00</td>
<td>$2.00</td>
</tr>
<tr>
<td>Electrodes</td>
<td></td>
<td></td>
<td>4</td>
<td>$4.00</td>
<td>$16.00</td>
</tr>
<tr>
<td>Electrode Cables</td>
<td></td>
<td></td>
<td>4</td>
<td>$6.00</td>
<td>$24.00</td>
</tr>
<tr>
<td>PCB Fabrication</td>
<td></td>
<td></td>
<td>1</td>
<td>$1.50</td>
<td>$1.50</td>
</tr>
<tr>
<td>ARM® M4 MCU</td>
<td>NXP</td>
<td>MK20DX256VLH7</td>
<td>1</td>
<td>$5.68</td>
<td>$5.68</td>
</tr>
<tr>
<td>Schottky Barrier Diode</td>
<td>Toshiba</td>
<td>CUS520</td>
<td>2</td>
<td>$0.05</td>
<td>$0.10</td>
</tr>
<tr>
<td>Dual Instrumentation Amplifier</td>
<td>Analog</td>
<td>AD8426</td>
<td>1</td>
<td>$3.30</td>
<td>$3.30</td>
</tr>
<tr>
<td>Dual Op Amp</td>
<td>Analog Devices</td>
<td>AD8659ARZ-R7</td>
<td>2</td>
<td>$2.45</td>
<td>$4.90</td>
</tr>
<tr>
<td>3.3V LDO 150mA</td>
<td>Diode</td>
<td>AP7313-33SAG7</td>
<td>1</td>
<td>$0.13</td>
<td>$0.13</td>
</tr>
<tr>
<td>100mA Charge-Pump Inverter</td>
<td>Linear</td>
<td>LTC1983ES6-3</td>
<td>1</td>
<td>$1.91</td>
<td>$1.91</td>
</tr>
<tr>
<td>Rail-Rail Dual Op Amp</td>
<td>Microchip</td>
<td>MIC7111</td>
<td>1</td>
<td>$0.52</td>
<td>$0.52</td>
</tr>
<tr>
<td>Micro SD Card Holder</td>
<td>Amphenol</td>
<td>101-00660-68-6</td>
<td>1</td>
<td>$1.04</td>
<td>$1.04</td>
</tr>
<tr>
<td>4-Position Connector</td>
<td></td>
<td></td>
<td>1</td>
<td>$0.50</td>
<td>$0.50</td>
</tr>
<tr>
<td>Cap 0.1uF</td>
<td></td>
<td></td>
<td>10</td>
<td>$0.100</td>
<td>$1.00</td>
</tr>
<tr>
<td>Cap 10nF</td>
<td></td>
<td></td>
<td>2</td>
<td>$0.060</td>
<td>$0.12</td>
</tr>
<tr>
<td>Cap 10uF</td>
<td></td>
<td></td>
<td>2</td>
<td>$0.500</td>
<td>$1.00</td>
</tr>
<tr>
<td>Cap 1uF</td>
<td></td>
<td></td>
<td>1</td>
<td>$0.008</td>
<td>$0.01</td>
</tr>
<tr>
<td>Cap 20uF</td>
<td></td>
<td></td>
<td>4</td>
<td>$0.500</td>
<td>$2.00</td>
</tr>
<tr>
<td>Cap 4.7uF</td>
<td></td>
<td></td>
<td>1</td>
<td>$0.050</td>
<td>$0.05</td>
</tr>
<tr>
<td>Res 100k</td>
<td></td>
<td></td>
<td>4</td>
<td>$0.003</td>
<td>$0.01</td>
</tr>
<tr>
<td>Res 10k</td>
<td></td>
<td></td>
<td>3</td>
<td>$0.003</td>
<td>$0.01</td>
</tr>
<tr>
<td>Res 1k</td>
<td></td>
<td></td>
<td>2</td>
<td>$0.003</td>
<td>$0.01</td>
</tr>
<tr>
<td>Res 2.61k</td>
<td></td>
<td></td>
<td>2</td>
<td>$0.003</td>
<td>$0.01</td>
</tr>
<tr>
<td>Res 200k</td>
<td></td>
<td></td>
<td>2</td>
<td>$0.003</td>
<td>$0.01</td>
</tr>
<tr>
<td>Res 500k</td>
<td></td>
<td></td>
<td>6</td>
<td>$0.003</td>
<td>$0.02</td>
</tr>
<tr>
<td>Res 50k</td>
<td></td>
<td></td>
<td>3</td>
<td>$0.003</td>
<td>$0.01</td>
</tr>
<tr>
<td>Res 7.5k</td>
<td></td>
<td></td>
<td>2</td>
<td>$0.003</td>
<td>$0.01</td>
</tr>
<tr>
<td>RF Shield Frame</td>
<td></td>
<td></td>
<td>1</td>
<td>$1.70</td>
<td>$1.70</td>
</tr>
<tr>
<td>RF Shield Cover</td>
<td></td>
<td></td>
<td>1</td>
<td>$0.85</td>
<td>$0.85</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>$73.38</strong></td>
</tr>
</tbody>
</table>
// Extended-Use ECG Monitor Code for Teensy 3.2
// Version: 6.0
// Daniel Soski

#include <stdio.h>        // sprintf
#include <ADC.h>          // ADC0 and ADC1
#include <SPI.h>          // SPI interface
#include <SD.h>           // SD Card
#include <TimeLib.h>      // RTC library (Time/TimeTeensy3 example)
#include <Time.h>
#include <DS1307RTC.h>
#include <AES.h>          // AES encryption
#include <kinetis.h>      // K20 register values
#include <Snooze.h>

// Patient Defines
#define PATIENT_ID    0
#define PATIENT_PASS  {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}

// ADC Global Variables
const int adcRes = 12;                                  //////////////////
const int adcAvg = 1;                                   //////////////////
const int adcPin0 = A2;
const int adcPin1 = A3;
ADC *adc = new ADC(); // adc object
ADC::Sync_result result;

// SD Card (SPI) Global Variables
#define ERASE_SD_ON_WAKE 1
const int dataOut = 7;
const int dataln = 8;
const int spiClock = 14;
const int chipSelect = 15;
unsigned int chipDetect = 9;
char LOG_FILE[20]   = "meas_XXXXXXXXXX.txt";
char temp_file[20]  = "meas_XXXXXXXXXX.txt";

// Timer Global Variables
#define SCB_SCR_SLEEPDEEP 0x4u
#define SCB_SCR_SLEEPONEXIT 0x2u
const int sampleFreq = 1000;//500; // Hz                      //////////////////

// RTC
tmElements_t tm;

// Data Storage Global Variables
const unsigned int MAX_BUFFER_SAMPLES = 2000; // Per ADC channel                   //////////////////
const unsigned int TWO_SAMP_CHAR_LEN = 4;
const unsigned int MAX_BUFFER_LEN = MAX_BUFFER_SAMPLES * TWO_SAMP_CHAR_LEN;
volatile unsigned int numSamplesPerChannel = 0;
byte leadBuffer[MAX_BUFFER_LEN];
char result_temp[4];
time_t lastSampleTime;

// AES Encryption Variables
const unsigned int NUM_BUFFER_CHUNKS = MAX_BUFFER_LEN / N_BLOCK;
AES aes;
byte key[2 * N_BLOCK] = PATIENT_PASS;  // 32 bytes / 256 bits long
byte iv[N_BLOCK];
byte cipher[MAX_BUFFER_LEN];
#define PRINT_TIMESTAMP 1
#define ENCRYPT_DATA 1

// Debug Global Variables
#define SLEEP_ENABLE 0
elapsedMicros time;
volatile unsigned long t_record = 0;

// LED Global Variables
const int led = LED_BUILTIN;  // the pin with a LED
//digitalWriteFast(led, !digitalReadFast(led));

// -------------------------
// ------- Setup (runs once) -------
// -------------------------
void setup (void)
{
  // ------- Setup I/O Ports -------
  for (int i = 0; i < 46; i++) {
    pinMode(i, OUTPUT);  // set all pins to output to save power
  }
  pinMode(led, OUTPUT);
  pinMode(adcPin0, INPUT);
  pinMode(adcPin1, INPUT);

  unsigned int negRailSHDN_pin = 19;
  pinMode(negRailSHDN_pin, OUTPUT);
  digitalWriteFast(negRailSHDN_pin, 1);
  // -------------------------

  // ------- Setup Debug Serial -------
  Serial.begin(9600);
  delay(1000);
#if !SLEEP_ENABLE
  //while (!Serial) {}
  delay(1000);
#endif
  Serial.println("Begin Setup");
  // -------------------------

  // ------- RTC -------
  // get the date and time the compiler was run
  if (getDate(__DATE__) && getTime(__TIME__)) {

}}
// and configure the RTC with this info
setTime(makeTime(tm));
Serial.print("RTC configured time=");
Serial.print(_TIME_);
Serial.print(", Date=");
Serial.println(_DATE_);
} // -----------------------------------------

// ------------ Data Buffer / AES ------------
Serial.print("Num Data Chunks: ");
Serial.println(NUM_BUFFER_CHUNKS);
if ( aes.set_key(key, 256) != SUCCESS ) {
    Serial.println("Failed set key");
    while (1);
} else {
    char key_char[32];
    for (int i = 0; i < 32; i++) {
        key_char[i] = (char)key[i];
    }
    Serial.print("Key set: ");
    Serial.println(key_char);
} // -----------------------------------------

// ---------------- Setup ADC -----------------
// Setup ADC0
adc->setReference(ADC_REF_3V3, ADC_0);
adc->setAveraging(adcAvg, ADC_0); // set number of averages
adc->setResolution(adcRes, ADC_0); // set bits of resolution
adc->setSamplingSpeed(ADC_VERY_HIGH_SPEED, ADC_0); // change the sampling speed
adc->setConversionSpeed(ADC_HIGH_SPEED, ADC_0); // change the conversion speed
// Setup ADC1
adc->setReference(ADC_REF_3V3, ADC_1);
adc->setAveraging(adcAvg, ADC_1); // set number of averages
adc->setResolution(adcRes, ADC_1); // set bits of resolution
adc->setSamplingSpeed(ADC_HIGH_SPEED, ADC_1); // change the sampling speed
adc->setConversionSpeed(ADC_HIGH_SPEED, ADC_1); // change the conversion speed
} // -----------------------------------------

// --------- Setup SD Card (SPI) ---------
pinMode(chipDetect, INPUT_PULLUP);
pinMode(MISO, INPUT_PULLUP); // set internal pullup resistor on DIN
SPI.setMOSI(dataOut); // set custom pin
SPI.setMISO(dataIn); // set custom pin
SPI.setSCK(spiClock); // set custom pin
Serial.println("Initializing SD card...");
if (!SD.begin(chipSelect)) { // see if the card is present and can be initialized
    Serial.println("Card failed, or not present");
    digitalWriteFast(led, HIGH);
    for (;;) {}
Serial.println("SD Card Initialized.");

if (ERASE_SD_ON_WAKE) {
    // file name is based upon start time
    sprintf(LOG_FILE, "data.ecg");
    // sprintf(LOG_FILE, "%02d%02d%02d%02d.bin", month(), day(), hour(), minute());
    Serial.println("Initializing log file...");
    if (SD.exists(LOG_FILE)) {
        SD.remove(LOG_FILE);
        Serial.println("Previous log file deleted.");
    }
    File dataFile = SD.open(LOG_FILE, FILE_WRITE);
    if (dataFile) {
        Serial.print("Log file (");
        Serial.print(LOG_FILE);
        Serial.println(") initialized");
        dataFile.close();
    } else {
        Serial.print("Log file (");
        Serial.print(LOG_FILE);
        Serial.println(" creation failed");
        while (1);
    }
}

// -----------------------------------------
// Simulating RTC
// -----------------------------------------

noInterrupts();

// ----------- Low Power Timer -----------
SIM_SCGC5 |= SIM_SCGC5_LPTIMER; // Enable low power timer module
SIM_SCGC6 |= SIM_SCGC6_RTC; // Enable RTC module and interrupts
RTC_CR |= RTC_CR_OSCE | RTC_CR_SC4P | RTC_CR_SC8P; // enable 32kHz osc, 12pF load
SIM_SOPT1 |= SIM_SOPT1_OSC32KSEL(2); // Select ERCLK32 source (0=32kHz system osc, 2=32kHz RTC, 3=1kHz LPO)
LPTMR0_CSR &= ~LPTMR_CSR_TEN; // disable timer
LPTMR0_PSR = LPTMR_PSR_PCS(2) + LPTMR_PSR_PBYP; // PCS=1->1kHz LPO, 2->32kHz Xtal // NoGlitchFilter
LPTMR0_CMR = 32768 / sampleFreq; // f = f_clk/CMR, count to XXX = 2ms
LPTMR0_CSR = LPTMR_CSR_TIE + LPTMR_CSR_TCF; // Enable interrupt, clear other interrupts
LPTMR0_CSR |= LPTMR_CSR_TEN; // enable timer
LLWU_ME = 0x01; // Set LPTMR interrupt as sleep-mode wake-up source

Serial.println("End Setup");
Serial.flush();
delay(1000);

NVIC_ENABLE_IRQ(IRQ_LPTMR); // enable timer interrupt
 interrupts();
// Enter Low Leakage Stop Mode (LLS)
if (SLEEP_ENABLE) {
    Serial.end(); // turn off usb serial to save power
    usbDisable(); // turn off usb serial to save power
    SCB_SCR |= SCB_SCR_SLEEPONEXIT; // sleep after interrupt
    enter_lls();
}

// -----------------------------------------
// ------------------- Main Loop -------------------
// -----------------------------------------
byte buffer_temp[MAX_BUFFER_LEN];
unsigned int len_temp;
time_t lastSampleTime_temp;

void loop (void) {
    // -------- Send data to SD Card --------
    if (numSamplesPerChannel == MAX_BUFFER_SAMPLES) {
        noInterrupts();
        // Make temporary copy of leadBuffer and length to prevent contention with ISR
        memcpy(buffer_temp, leadBuffer, sizeof(leadBuffer));
        len_temp = numSamplesPerChannel;
        numSamplesPerChannel = 0;
        lastSampleTime_temp = lastSampleTime;
        interrupts();

        if (digitalReadFast(chipDetect) == LOW) { // check if SD card is present
            #if ENCRYPT_DATA
                // set iv as (char)lastSampleTime padded with zeros
                char timeStr[N_BLOCK];
                sprintf(timeStr, "%016ld", lastSampleTime_temp); // converts to decimal
                for (int i = 0; i < N_BLOCK; i++) {
                    iv[i] = (byte)timeStr[i];
                }

                // encrypt data in chunk-by-chunk (cbc) mode
                if ( aes.cbc_encrypt(buffer_temp, cipher, NUM_BUFFER_CHUNKS, iv) != SUCCESS ) {
                    // Serial.println("Failed encrypt");
                } else {
                    // Serial.println("Data Encrypted");
                }
            #endif

            // write entire data buffer to SD card
            // Serial.print("Writing data to SD card for numSamplesPerChannel = ");
            // Serial.println(len_temp);

            digitalWriteFast(led, HIGH);
        }
    }
}
File dataFile = SD.open(LOG_FILE, FILE_WRITE);
if (dataFile) { // if the file is available, write to it
    #if PRINT_TIMESTAMP
    // write time
    dataFile.write((lastSampleTime_temp >> 24) & 0xFF);
dataFile.write((lastSampleTime_temp >> 16) & 0xFF);
dataFile.write((lastSampleTime_temp >> 8) & 0xFF);
dataFile.write((lastSampleTime_temp >> 0) & 0xFF);
    #endif
    // write data
    #if ENCRYPT_DATA
    dataFile.write(cipher, MAX_BUFFER_LEN);
    #else
    dataFile.write(buffer_temp, MAX_BUFFER_LEN);
    #endif
    // finalize write
    dataFile.close();
    // Serial.println("Finished writing to SD card");
} else { // if the file isn't open, pop up an error
    // Serial.print("Error opening ");
    // Serial.println(LOG_FILE);
}
digitalWriteFast(led, LOW);
} else {
digitalWriteFast(led, HIGH);
}
// -----------------------------------------
if (SLEEP_ENABLE) {
    SCB_SCR |= SCB_SCR_SLEEPONEXIT; // sleep after interrupt
    enter_lls();
}
// -----------------------------------------
// ---- LPTMR ISR - Called every sample ----
// -----------------------------------------
void lptmr_isr (void)
{
    LPTMR0_CSR |= LPTMR_CSR_TCF; // clear interrupt

    // ------- Sample ADC_0 & ADC_1 ------- (~9us)
    result = adc->analogSynchronizedRead(adcPin0, adcPin1);
    // -----------------------------------------

    // ------- Save Data to Mem -------
    leadBuffer[numSamplesPerChannel * TWO_SAMP_CHAR_LEN] = (result.result_adc0 >> 8) & 0x00FF;
leadBuffer[numSamplesPerChannel * TWO_SAMP_CHAR_LEN + 1] = (result.result_adc0 >> 0) & 0x00FF;
leadBuffer[numSamplesPerChannel * TWO_SAMP_CHAR_LEN + TWO_SAMP_CHAR_LEN / 2] = (result.result_adc1 >> 8) & 0x00FF;
leadBuffer[numSamplesPerChannel * TWO_SAMP_CHAR_LEN + TWO_SAMP_CHAR_LEN / 2 + 1] = (result.result_adc1 >> 0) & 0x00FF;

numSamplesPerChannel++;
// -----------------------------------------

// -------- Check buffer length --------
if (numSamplesPerChannel == MAX_BUFFER_SAMPLES) { // Exit low power mode
  noInterrupts();
  SCB_SCR &= ~SCB_SCR_SLEEPONEXIT; // disable sleep after interrupt
  asm("DSB"); // Ensure effect of last store takes effect
  if (lastSampleTime == now()) {
    digitalWriteFast(led, HIGH);
  }
  lastSampleTime = now();
} // -----------------------------------------

// -----------------------------------------
// ---- RTC Time Functions ----
// -----------------------------------------

bool getTime(const char *str)
{
  int Hour, Min, Sec;
  if (sscanf(str, "%d:%d:%d", &Hour, &Min, &Sec) != 3) return false;
  tm.Hour = Hour;
  tm.Minute = Min;
  tm.Second = Sec;
  return true;
}

const char *monthName[12] = {

bool getDate(const char *str)
{
  char Month[12];
  int Day, Year;
  uint8_t monthIndex;
  if (sscanf(str, "%s %d %d", Month, &Day, &Year) != 3) return false;
  for (monthIndex = 0; monthIndex < 12; monthIndex++) {
    if (strcmp(Month, monthName[monthIndex]) == 0) break;
  }
}
if (monthIndex >= 12) return false;
    tm.Day = Day;
    tm.Month = monthIndex + 1;
    tm.Year = CalendarYrToTm(Year);
    return true;
}
% ECG FFT
% by Daniel Soski

% Clear workspace
clear;

% Set data file variables
data_file_name = 'raw.csv';
f_samp = 500; % measurement sampling frequency

% Set program options
plot_time = true;
plot_freq = true;

% Load data file
ecg_data = load(data_file_name);

data_file_name = 'raw.csv';
f_samp = 500; % measurement sampling frequency

% Calulate data file variables
tstep = 1 / f_samp;
max_index = length(ecg_data);

% Create time array
for i = 1:max_index
t(i) = (i-1)*tstep;
end

% Create frequency array
for i = 1:max_index
    freq_range(i) = (1/tstep)*(i-1)/max_index; % range of 0hz to sampling freq (1/tstep)
end

% Take FFT of data and convert to watts
ecg_data_fft = fft(ecg_data_max_index);
ecg_data_fft_mag = 2*ecg_data_fft.*conj(ecg_data_fft)/(max_index)^2; % multiply by 2/n^2 to get units of watts

% Filter 1 - 1Hz to 150Hz (Blocking Bandpass Filter)
ecg_data_fft_filtered = ecg_data_fft;
f_min = 1;
f_max = 150;
for i = 1:max_index
    % must reflect the filtering to the other side of the frequency range
    if ((freq_range(i)<f_min || freq_range(i)>f_max) && ...
        (freq_range(i)<f_samp-f_max || freq_range(i)>f_samp-f_min))
        ecg_data_fft_filtered(i) = 0.0;
    end
end

% Calculate Average Power
fft_sum = 0;
for i = 1:max_index
    fft_sum = fft_sum + ecg_data_fft_filtered(i);
end
fft_avg_pwr = fft_sum/max_index;

% Filter 2 - 59Hz to 61Hz (Averaging Bandstop Filter)
f_min = 59;
f_max = 61;
for i = 1:max_index
    % must reflect the filtering to the other side of the frequency range
    if ((freq_range(i)>f_min && freq_range(i)<f_max) || ... 
        (freq_range(i)>f_samp-f_max && freq_range(i)<f_samp-f_min))
        ecg_data_fft_filtered(i) = fft_avg_pwr;
    end
end

% Convert filtered FFT data to watts
ecg_data_fft_mag_filtered = 2*ecg_data_fft_filtered.*conj(ecg_data_fft_filtered)/(max_index)^2;

% Take the inverse-FFT of the filtered data
ecg_data_filtered = ifft(ecg_data_fft_filtered);

% Plot the time-domain data
if (plot_time)
    figure(1);
    plot(t,ecg_data,t,ecg_data_filtered);
end

% Plot the frequency-domain data
if (plot_freq)
    figure(2);
    semilogy(freq_range(1:max_index/2),ecg_data_fft_mag(1:max_index/2), ... 
        freq_range(1:max_index/2),ecg_data_fft_mag_filtered(1:max_index/2));
end