6-9-2015

Low-cost electrocardiogram device for preventative healthcare in rural populations of developing countries

J.P. Ertola
Santa Clara University

Michael Whalen
Santa Clara University

Santa Clara University
DEPARTMENT of COMPUTER SCIENCE AND ENGINEERING

Date: June 9, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

J.P. Ertola
Michael Whalen

ENTITLED

Low-Cost Electrocardiogram Device For Preventative Healthcare in Rural Populations of Developing Countries

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THESIS ADVISOR

DEPARTMENTCHAIR

Low-Cost Electrocardiogram Device For Preventative Healthcare in Rural Populations of Developing Countries

by

J.P. Ertola
Michael Whalen

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 9, 2015

Low-Cost Electrocardiogram Device For Preventative Healthcare in Rural Populations of Developing Countries

J.P. Ertola
Michael Whalen

Department of Computer Science and Engineering
Santa Clara University
June 9, 2015

ABSTRACT

We have created a prototype electrocardiogram (ECG) device to screen rural populations of developing countries for cardiovascular diseases (CVDs). Our device is affordable and easy to use so it can be accessible to as many people as possible. We leverage widespread SMS infrastructures to remotely report cardiovascular health to doctors; thus not requiring the doctor to be on-site to interpret test results. Our device can be used to screen people for cardiovascular warning signs. Therefore, it is not meant to be a diagnostic tool.

Table of Contents

List of Figures

Chapter 1

Forward

Forward by Gigi Groener and Camille Young; Senior Public Health students at Santa Clara University

Nigeria, Africa's most populated country (177,155,754 people), borders the Gulf of Guinea in West Africa[3]. The official language spoken in Nigeria is English; other common languages include Hausa, Yoruba, Igbo, and Fulani[3]. The major ethnic groups that reside in Nigeria are the Hausa and Fulani (29%), Yoruba (21%), and Igbo (18%), followed by Ijaw (10%), Kanuri (4%), Ibibio (3.5%) and Tiv (2.5%)[3]. Fifty percent of Nigerians identify as Muslim, 40% Christian, and 10% practice indigenous traditions. Nigeria's health profile can be can be further understood by taking a look at Nigerian political and economic histories[3].

Before independence, the British maintained control over Nigeria through claims that local civilian leaders easily succumbed to bribery and were incapable of providing an honest administration without misrule and oppression. However, British government was no stranger to corruption, oppression or bribery. Nigeria gained independence from British control and influence in 1960 after civilian efforts forced public exposure of the British leadership; transparency demonstrated a number of hypocrisies and inadequacies within British control. Yet, the country continued to face issues such as a 16-year military control, violence, corruption and government mismanagement[3]. Although recent violent and religious tensions emerged during the 2007 elections, Nigeria is now considered to be maintaining its longest peaceful period of civilian government since the adoption of its current constitution, in 1999[3].

The existing government structure is relatively stable, but the country has been unable to generate a sustainable economy throughout all of Nigeria. Agriculture, natural gas, oil, and telecommunication industries have sustained economic growth at rates about 6-8 percent per year, since the 1970s. Specifically, agriculture contributes to 40 percent of the GDP, but more recently the growth rate has begun to decrease. Nigeria relies heavily on the rural population, especially women, to sustain its economy. Yet, infrastructure development is largely ignored in these areas. Rural areas are deficient of necessary elements for successful production, such as reliable power, safe transportation, road systems, and access to clean water, schools, and healthcare. The lack of rural roads, irrigation, and power lead to spoilage, increased cost of food, and limited access to new technology that would increase yields. These are some of the most critical factors that threaten agriculture's profitability in the future. Without a stable economy, already scarce resources will become even more limited. Even more critical are the conditions of the workers, themselves.

The agriculture workforce is comprised of 90 percent of the rural population, yet 80 percent of this population lives under the poverty line, and 70 percent lives on less than $1.25 per day. As mentioned earlier, women make up the majority of the rural agriculture workforce. Because they are not held to the same social status as their male counterparts, they are unable to migrate to the cities to earn an education, receive further farming training, and visit health care centers. The government does subsidize housing in rural areas, but the income most women make is not enough to cover the cost of food and/or health services, let alone transportation. It's imperative that the government focuses its efforts towards healthcare development in rural Nigeria.

In 2010, the federal government of Nigeria issued a progress report for the international Millennium Development Goals (MDGs). The MDGs are eight, quantifiable, international goals aimed to address extreme poverty by 2015. The 2010 report projected that the country would not meet the goals by 2015; as a result, the federal government implemented the National Strategic Health Delivery Plan (NSHDP). The current health care system in Nigeria is comprised of multiple different state, national, international, public, and private organizations. Under management of Nigeria's Federal Ministry of Health, programs funded by World Bank, multilateral organizations (such as the United Nations (UN) and the World Health Organization (WHO)), and nongovernmental organizations are working to build the necessary infrastructure for basic primary health care services (ng.undp). By 2012, Nigeria had met the MDG target of gender parity (goal 3) and is on track to meet goals four and five by the end of 2015, to reduce child mortality and improve maternal health, respectively. Primarily, creating self-sustaining medical facilities and increasing the number of trained professionals, as well as providing care to the poorest and rural population pose significant challenges to achieving the health care goals (undp).

In order to address some of these challenges, the United States Center for Disease Control (CDC) has helped Nigeria establish its own CDC and epidemiology training facility to prevent and and respond to public health issues, such as disease outbreaks. Additionally, Nigeria's Federal Ministry of Health has 20 teaching hospitals, 22 federal medical centers and 13 specialty hospitals. However, two major barriers for receiving healthcare services in rural populations are cost and inconsistent local level implementation of services. First, in 2010, about 69 percent of healthcare expenditures were paid for out-of-pocket. Although a national health insurance scheme was successful in reducing costs for many Nigerians, a national level program failed to ensure coverage for the most vulnerable population—rural women and/or farmers. Thus, community based health insurance has since been piloted to provide more extensive coverage in rural populations and reduce the overall cost for individuals. Decentralization of health care into rural areas was initially done during military reign and there was little authentic accountability to the people being served. For this reason, partnership with private sector health care providers for rural populations is promising.

Some low cost ECGs have already been developed, however a further look at the qualities of the devices will reveal that the design of the existing ECGs would need to be modified in order to be effective in rural communities in Nigeria. We will discuss three existing ECG devices.

The first device consists of an ECG amplifier powered by a 9 volt transistor radion battery with four components: (1) three electrodes, (2) ECG amplifier, (3) USB interface and a (4)"Flexis JM Demonstration board." At only $4.76 per device, the portable and reusable electrodes, modest power consumption, and digitized readings sent via a USB or bluetooth method on a telephone seem promising. The device does not require the internet, does not need to be charged frequently, and most importantly, it is easy to travel with. However, the device requires the use of a laptop and Windows interface. Despite the devices ability to send digital ECG readings, the readings must be viewed using a laptop computer. This may complicate the process of a medical professional actually reading the results because many trained medical professionals in developing countries are not trained in computer interfaces.

AliveCor is another ECG device that relies on a cellular phone. The AliveCor device is a 1 lead system that connects to a smart phone through bluetooth. This $200 device is small and very portable but it requires the use of Internet or network data to send digitized ECG readings. This might pose a problem in rural Nigeria where access to cell data is not guaranteed. Additionally the device, with only one lead, is much less accurate than a three lead device.

Finally, General Electric has designed the Mac series ECG device for India. The device is user-friendly, costs about $500, portable, and provides printed ECG readings. Further, it functions with almost the same precision as an expensive ECG machine, making the readings very useful to the doctor. Yet, this ECG machine would be very difficult to integrate into rural community health care where the proper infrastructure is lacking. One challenge for integrating this device into routine care practices in rural communities is that it uses a cell phone battery, which would require frequent charging. Another difficulty, would be getting the printed ECG readings to a doctor in a timely manner.

Of all non-communicable disease, cardiovascular disease was the leading cause of death for adults in Nigeria in the year 2014. It was estimated that, of the approximate 2 million total deaths throughout the country, 140,000 deaths (7%) could be attributed to cardiovascular diseases (CVD)[4].

The most common cardiovascular diseases include coronary heart disease, heart attacks, and strokes. However, complications such as heart arrhythmias, congestive heart failure, or valve-related problems are also common[5]. There are many associated risk factors for CVDs. In general, smoking, alcohol consumption, high blood pressure and obesity significantly influence one's likelihood of experiencing cardiovascular disease. It was estimated that 6% of the adult population in Nigeria used some sort of tobacco product (2011), 35% experienced high blood pressure (2008) and 7% of the population was obese (2008)[4], thus putting a large proportion of the Nigerian population at risk. As they increasingly target and market to the most vulnerable populations - women and children, globalization has created markets for both the tobacco and alcohol industries [6]. Although tobacco and alcohol use are both preventable risk factors for cardiovascular disease, Nigeria is a potential area for expansion for both of these industries and thus tobacco and alcohol consumption are rising in conjunction with cardiovascular disease are.

Another potential explanation for the rise in deaths due to cardiovascular diseases throughout Nigeria is the globalization and urbanization of the country. As fast food restaurants are becoming increasingly prominent, the diet of an average Nigerian has significantly changed over the last 10 years [6]. The unfortunate consequence of this is a higher salt and sugar based diet, ultimately contributing to a greater number of obese individuals [6].

Nationally, Nigeria has set up a surveillance and monitoring system that is responsible for reporting all non-communicable diseases. However, the country lacks an action plan to integrate the risk factors for disease. Thus the actual implementation of the surveillance and monitoring system that they have designed is minimal or completely nonexistent in the most vulnerable regions.

Chapter 2

Engineering World Health

Our project is part of an an initial effort to join engineering senior design projects with Public Health students. The Frugal Innovation Lab (FIL) and department of Public Health were inspired by the non-profit organization Engineering World Health (EWH) to create a platform for collaboration between the two disciplines. EWH strives to implement biomedical engineering in a way that aligns with many of FIL's core value. Providing affordable access to medical equipment is paramount to both the FIL and EWH. As explained later in this thesis, our project's main goal was to create a low-cost electrocardiogram to be used in the developing world by people who do not currently have access to such resources. Because of the wonderful new partnership between the FIL and Public Health department and their shared mission aligning with EWH, we were fortunate enough to work with Public Health students in researching a case study.

Chapter 3

The Problem

In many developing countries healthcare is often described as an 80/20 problem. That is, eighty percent of the population has access to twenty percent of the country's healthcare resources. Many rural populations may live over sixty miles away from a clinic with sufficient medical equipment and even further from a full-service hospital. This inability to access proper healthcare for rural populations paired with rapidly increasing cardiovascular disease rates poses a serious problem.

Additionally, cardiovascular disease (CVD) is a global problem. As seen in Gigi and Camille's research, Nigeria has a massive portion of their population living under the poverty line who are also afflicted with high CVD rates. The high coupling of these two problems serve as the main motivations for our project because this marriage of poverty and heart disease is not unique to Nigeria.

Chapter 4

Introduction To Electrocardiograms

An electrocardiogram device, referred to as an ECG in this report but which can be also be called an EKG, is a medical device used to record the heart's electrical activity. Electrode leads, attached to the surface of the skin, pick up electrical signals that represent the heart's electrical activity. These are analog signals and are transmitted to electronics equipment that filters, amplifies, and digitizes them. A computer then analyzes the digital wave and outputs it to a monitor or prints it onto graph paper. The results are then interpreted by a doctor who determines the patient's cardiovascular health. The ECG wave is comprised of many parts, each indicating something important about the patient's heart.

An ECG wave is comprised of P, Q, R, S, and T waves. These waves can be combined to form segments, complexes, and intervals as seen in Figure 4.1. Different variations of this waveform correspond to different irregular heart beats called heart arrhythmias. There are many different kinds of arrhythmias with some being more immediately life-threatening than others. The most common is atrial fibrillation which occurs when the nerves controlling the heart's right atrium do not fire in a normal and rhythmic manner. Atrial fibrillation occurs in roughly 1% of world's population and increases their risk of stroke by 15%.

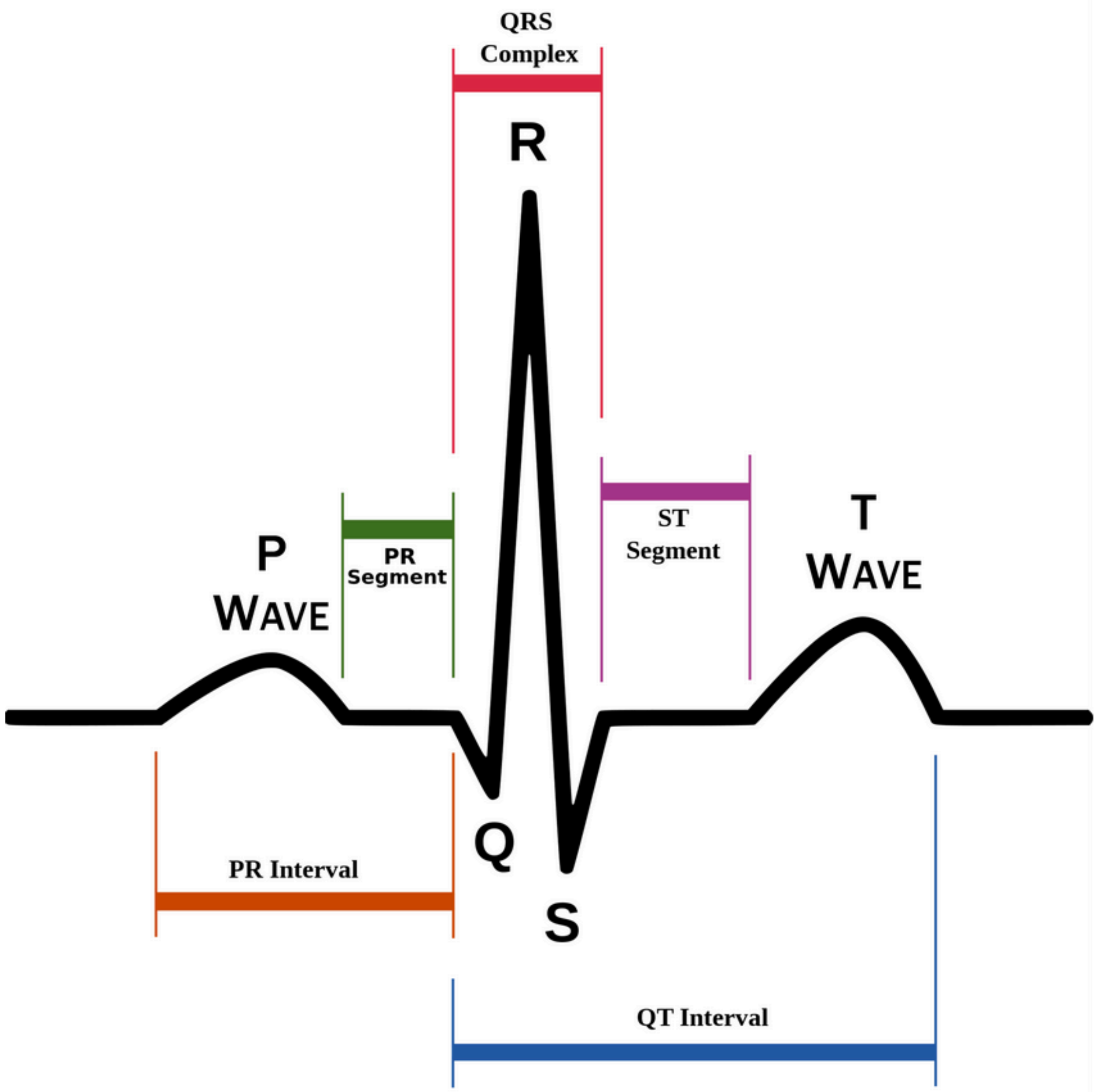

Figure 4.1: Diagram of an ECG Waveform

Chapter 5

Current Solutions

While there are affordable ECG devices in existence, they still fall short for one main reason: the doctor is required to interpret results in-person. This means the doctor has to be out in the field with patients or, even worse, test results must be transported back to the doctor whose analysis must be relayed back to the patient. Requiring a doctor to be out in the field asks for a lot of their time and also places a burden on the clinic or hospital they would normally be at. Doctors are hard to come by in the developing world so keeping them stationed at their normal posts is of utmost importance. This also strains the healthcare worker's resources as they have to travel back and forth between doctor and patient. Because our targeted patients live far away from adequate healthcare infrastructure, all of this travel is extremely hard work and can take several days. Although General Electric (GE) has developed an ECG device costing only $500 (an excellent and realistic price point for its applications), their device is plagued by the aforementioned problems.

Chapter 6

Overview of Our Solution

Our solution is to build an affordable electrocardiogram device to be used to screen rural populations in developing countries for possible heart disease. Traditionally, test results are outputted locally from an expensive ECG machine. The administering of an ECG test and interpreting its results are a tightly coupled process because a doctor is required to be on-site in order to interpret the ECG waveform. In rural settings, it is unrealistic to ask a doctor to be out in the field for ECG tests. Therefore, it is our goal to reduce the costs, manpower, and time involved with administering ECG tests to remote populations through the development of an affordable and portable electrocardiogram device. After the patient's ECG wave has gone through some analog signal processing and analog to digital conversion (ADC), we use a microcontroller to analyze the patient's wave. With our device, a trained person can administer ECG tests that relay information back to doctors via SMS (text messaging) rather than requiring a patient to travel to a clinic or hospital for this basic screening. We provide doctors with a string of text describing each important part of the patient's ECG wave. By providing doctors with this detailed text message, they will be able to determine if something is immediately alarming and can suggest for the patient to seek proper testing. We do not provide any diagnosis or suggest a possible diagnosis. Instead, we simply provide doctors with data in an effort to screen patients for dangerous heart conditions. Our ECG allows doctors to remotely screen patients and provide further treatment if necessary.

Chapter 7

Requirements

To establish a greater understanding of what is required of our device, we have compiled a list of functional requirements, nonfunctional requirements, and design constraints.

- Functional requirements define what tasks the system must perform.
- Nonfunctional requirements define the manner in which the system performs the functional requirements.
- Design constraints limit the way the system is designed and implemented.

7.1 Functional Requirements

Our device:

- Will filter and amplify the incoming analog signal using appropriate signal processing.
- Will convert the processed analog signal into a digital signal.
- Will report ECG data to doctors via text message.

7.2 Nonfunctional Requirements

Our device:

- Will be user-friendly so there are no operator issues when used out in the field.
- Will send SMS messages containing data about the patient's ECG wave in a formatted and easy-to-understand string.

7.3 Design Constraints

Our device:

- Will be affordable so to be accessed by rural populations in developing countries who could otherwise not afford basic healthcare.

Chapter 8

Diagrams

The following diagrams help visually describe our system in more detail.

8.1 Architectural Diagram

Figure 8.1 is a high-level view of our device. Beginning on the top left, our device will take in the patient's analog signal through external ECG leads. The analog signal is then filtered through signal processing techniques. The signal is formatted into a string of data describing the processed wave. The data is sent via SMS message to the doctor.

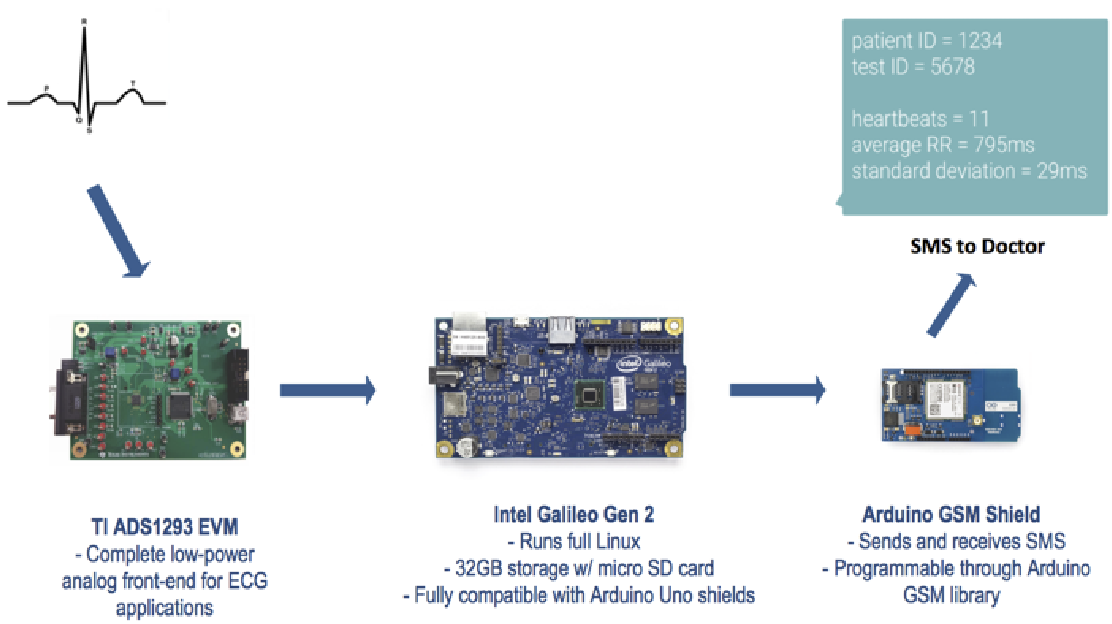

Figure 8.1: Architectural Diagram describing the overall flow of our device.

8.2 Interaction Diagrams

These use case diagrams describe the actions that can be taken by the trained professional, doctor, and patient in order for our system to perform its functional requirements.

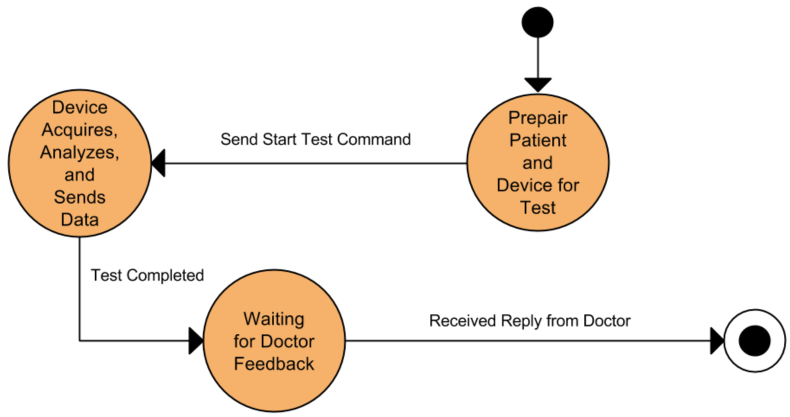

Figure 8.2: Diagram showing the trained professional interacting with our system.

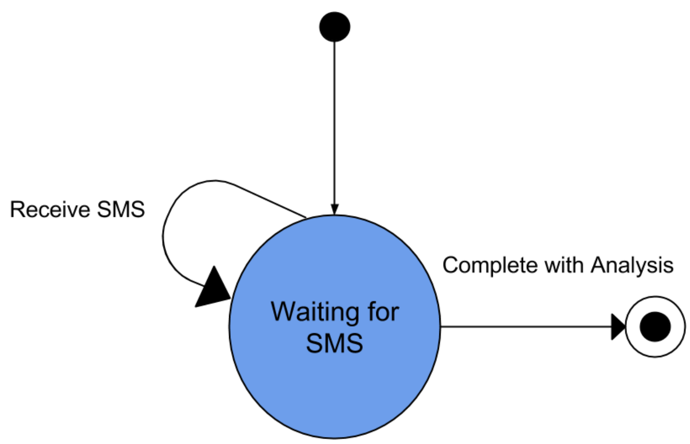

Figure 8.3: Diagram showing the doctor interacting with our system.

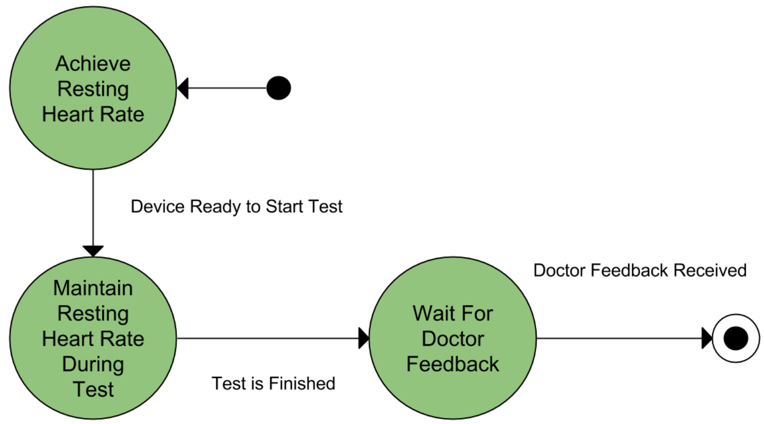

Figure 8.4: Diagram showing the patient interacting with our system.

8.3 Finite State Machine

This finite state machine shows the states our ECG can be in, the transitions between states, and reasons for those transitions.

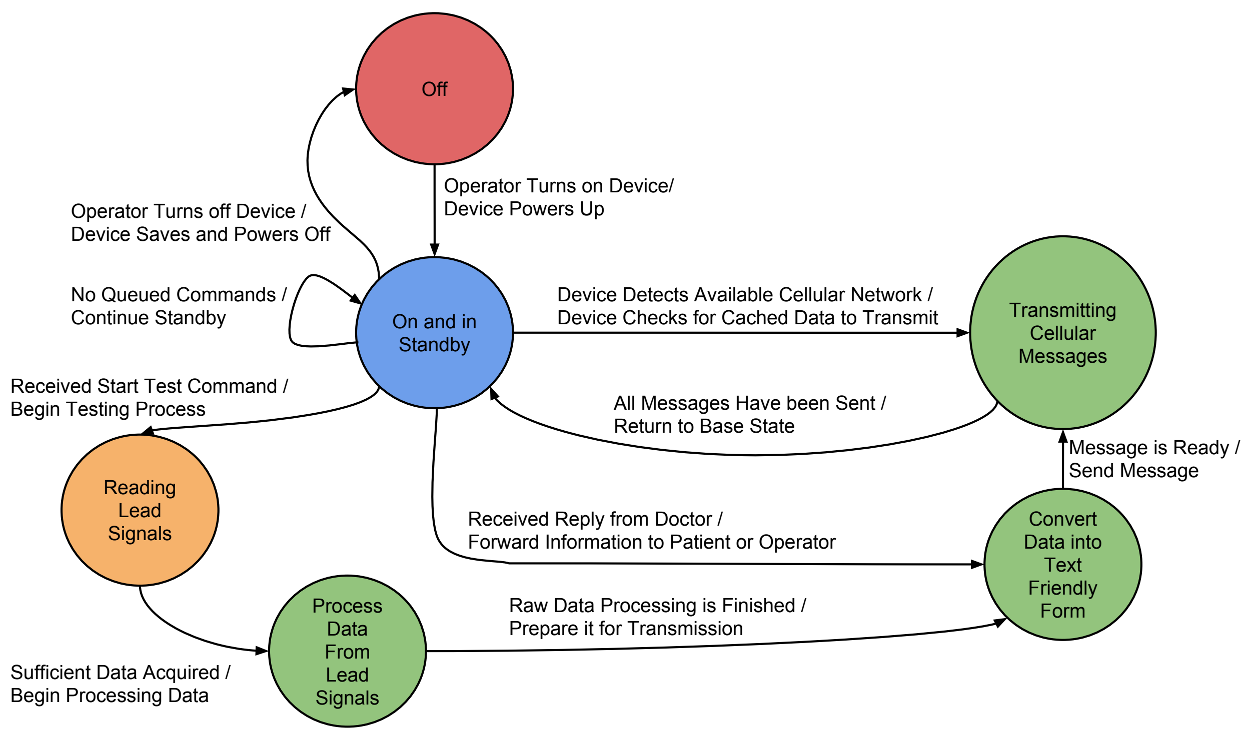

Figure 8.5: Finite State Machine Diagram showing the different states the device can be in and how it transitions from state to state.

Chapter 9

Technologies Used

9.1 Analog Signal Processing

In order to obtain an ECG wave that can be analyzed by our microcontroller, we must first filter the analog signal coming from the patient. Electrical noise is one of the biggest concerns of an ECG because a noisy signal will not allow our device to perform accurate analysis. Traditionally, common signal noise comes from both the patient as well as the mains power which the ECG machine is plugged into. Noise can come from the patient through muscle contractions around where the leads are attached on their body. Another common artifact occurring in ECG signals is baseline drift. This occurs when the waveform drifts upward from the horizontal axis, degrading the integrity of waveform. Baseline drift and signal noise can be corrected, to a certain degree, through different filtering mechanisms.

In order to obtain a clean signal, we used the Texas Instruments ADS1293EVM complete low power integrated analog front end for ECG applications. The evaluation board (ADS1293EVM) seen in Figure 9.1 is used to support the chip while the actual chip itself is the ADS1293. This chip is responsible for filtering and digitizing the incoming analog signals. The ADS1293 is an application specific integrated circuit (ASIC) and therefore performs all of the analog signal processing as well as the analog to digital conversion (ADC) with a 24-bit resolution. This high resolution ADC is necessary in order to provide doctors with the most accurate data possible. The ADS1293 is well suited for our device because ”the ADS1293 incorporates all features commonly required in portable, low-power electrocardiogram (ECG) applications. With high levels of integration and exceptional performance, the ADS1293 enables the creation of scalable medical instrumentation systems at significantly reduced size, power, and overall cost.” (Texas Instruments) Despite its $100 price, the ADS129EVM is more accurate and has higher performance than anything we could design.

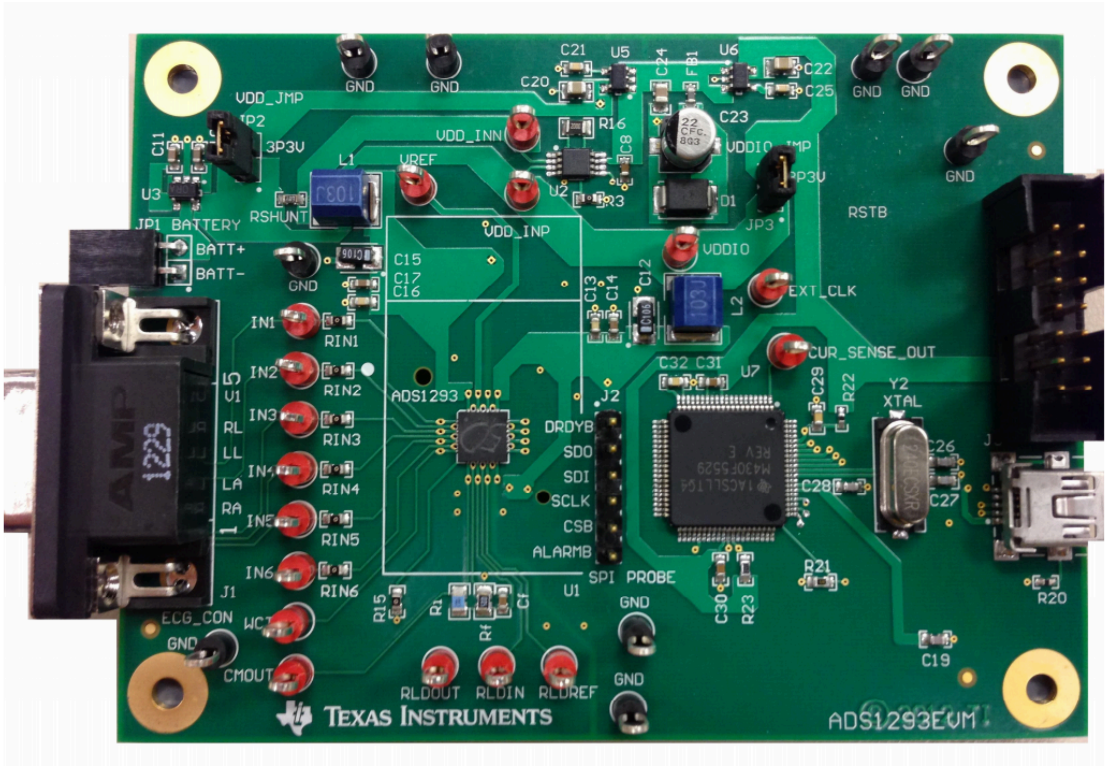

Figure 9.1: Photograph of the ADS1293EVM.

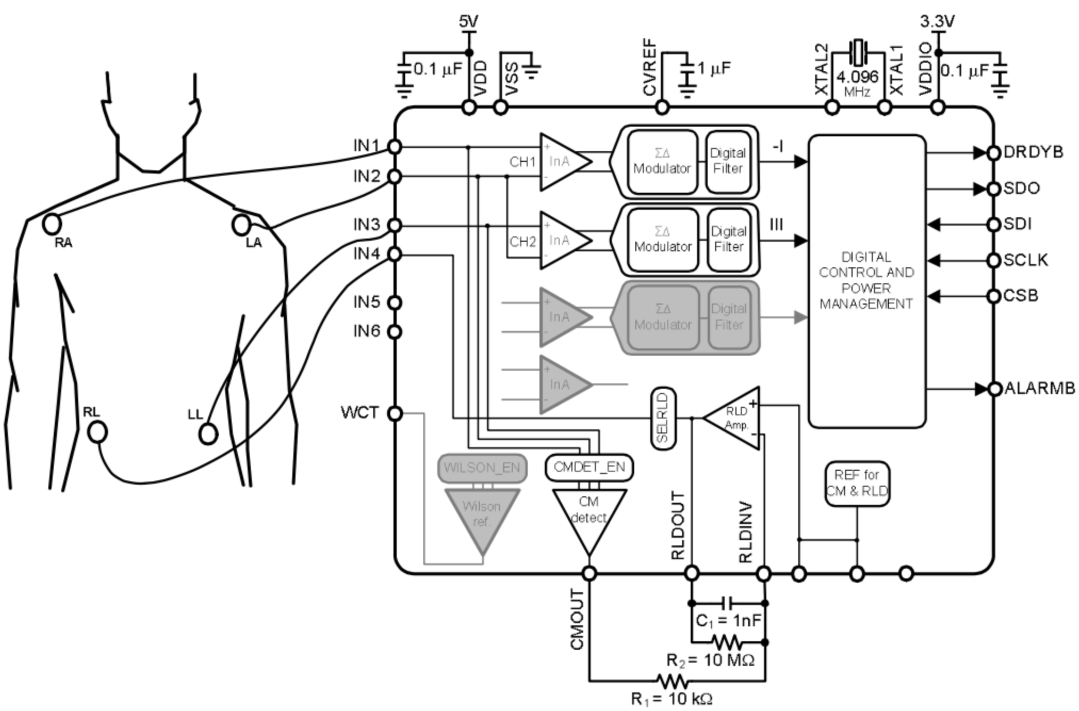

Figure 9.2: Block diagram of the ADS1293 with a 3 lead ECG connection to the patient's body.

9.2 Intel Galileo Gen 2 Microcontroller

Aruidno and Intel worked in conjunction to make the Galileo series of microcontrollers that can be used for many different applications. We will use a this microcontroller to perform the core functionality of our device. This core functionality includes digital signal analysis, data formatting for SMS, and sending and receiving SMS messages. The Galileo is based on the Intel Quark architecture. This provides a 32-bit Intel Pentium brand system on chip (SoC), allowing us to run Linux under the hood. Having access to Linux allows us to easily and efficiently store patient data using its file system. Other features of the Intel Galileo include the Ethernet port and micro-SD slot. The Ethernet port allows us to SSH into the Linux running on the board and the micro-SD card slot allows up to 32GB of accessible storage for which we can store patient records and data. The digital signal will come into the Galileo through its digital I/O pins. The Galileo has a built-in analog to digital converter (ADC) with a 12 bit resolution. However, we require an ADC with a higher resolution so we use the Texas Instruments ADS1293. Lastly, the $70 price tag and open source Arduino software make the Intel Galileo an ideal microcontroller for our project.

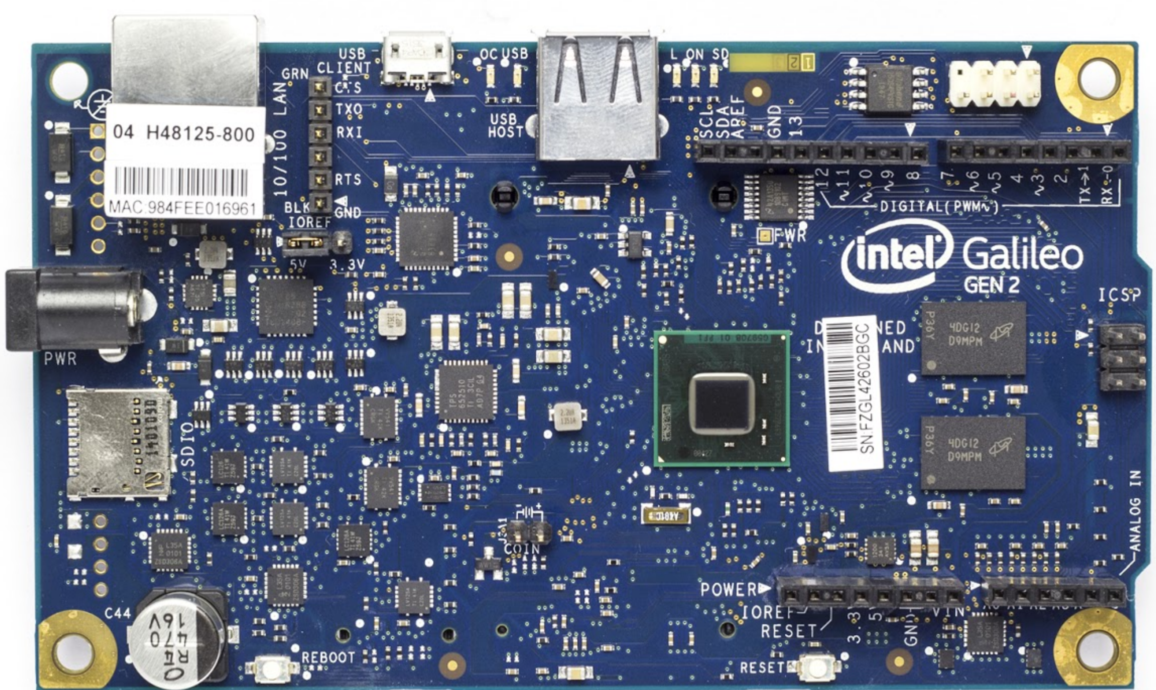

Figure 9.3: This is the Intel Galileo Gen 2 microcontroller used in for this project.

9.3 Data Reporting and Interfacing Through SMS

This section describes the data reporting and how a user interacts with our device. Despite our device not having a traditional user interface like one might find on a website or mobile application, we nonetheless have a means for interaction. Ideally, there will be cell coverage wherever our device is operating. The trained user will be able to begin a test and send results all through texting commands to the phone number associated with our device.

Our device leverages SMS for all of its communication. All users interact with our device through text messaging in a variety of ways depending on the user's role. A visual representation of how each user interacts with our device can be seen in the Figure 8.2, Figure 8.3, and Figure 8.4. Section 9.4 and Section 9.5 specify the two commands we implemented and how they work. All commands given to our device follow a simple format comprising of a comma-separated list of arguments.

9.4 Conduct Test

Conducting a test comprises of a three steps. First, it records the patient's ECG signal. Second, it analyzes the results. Third, it sends those results to the doctor. A unique test ID is generated by our device for each test. The format for conducting a test is as follows:

`command, patient ID, doctor phone number`

example: `0, 1234, 4085554321`

9.5 Send Test Results

Our device is able to store test data and results through a combination of the Intel Galileo's expandable memory and its Linux file system. Therefore, we are able to send test results to doctors for as long as they exist on the Galileo's microSD card. Test results are accessed through a combination of a patient ID and test ID. A patient ID is obviously unique across all patients while a test ID is unique across all tests. Our device generates a unique test ID for each conducted test. Therefore, a specific test can be accessed by a combination of patient ID and test ID. To send saved test results the command format is as follows:

`command, patient ID, test ID, doctor phone number`

example: `2, 1234, 5678, 4085554321`

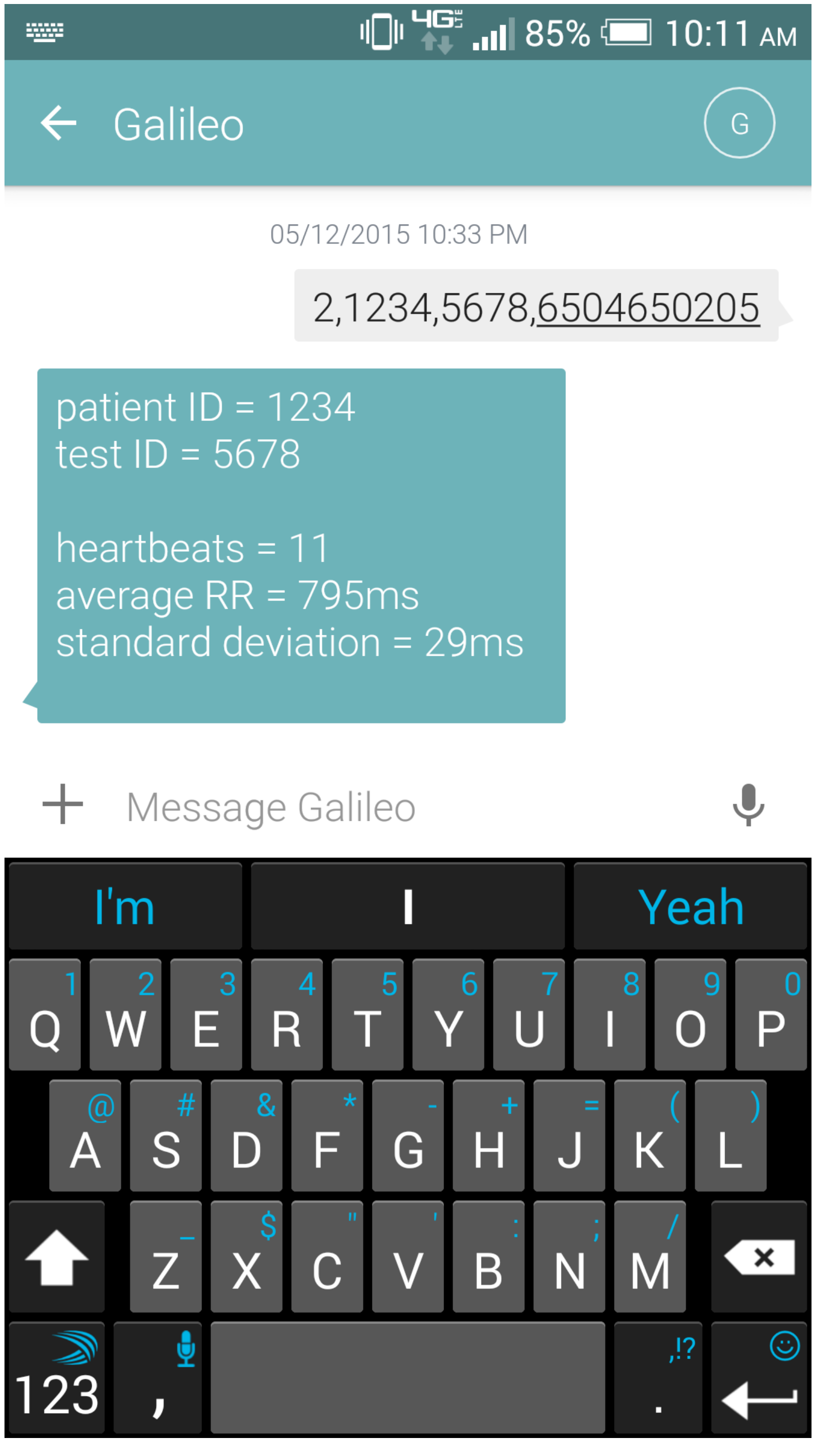

Figure 9.4: Screenshot of SMS interaction with the Galileo

9.6 Arduino GSM Shield

Arduino makes a variety of attachments for their microcontroller boards called ”shields.” We use their GSM shield to send and receive text messages. GSM stands for Global System for Mobile Communication (originally Groupe Special Mobile in French) and was developed by the European Telecommunications Standards Institute to describe protocols for digital cellular networks. The shield allows the trained user to communicate with our device and for our device to send the patient's ECG data to a doctor via text message. The GSM shield requires an active SIM card in order to function. A SIM card, or subscriber identification module card, is an integrated circuit that securely stores the information used to identify and authenticate subscribers on mobile telecommunications devices. The SIM card is registered with a mobile telecommunications provider and must be activated through a paid contract in order for our device to have access to the cellular network. Although the GSM shield is expensive at $90, it is crucial to making this device mobile as well as easy to use.

The patient's ECG data is sent back to doctors in the form of a text message. Each message is limited to 160 characters of digits, letters, and symbols. The text message contains the patient ID, test ID, average RR interval, and RR standard deviation (discussed more later).

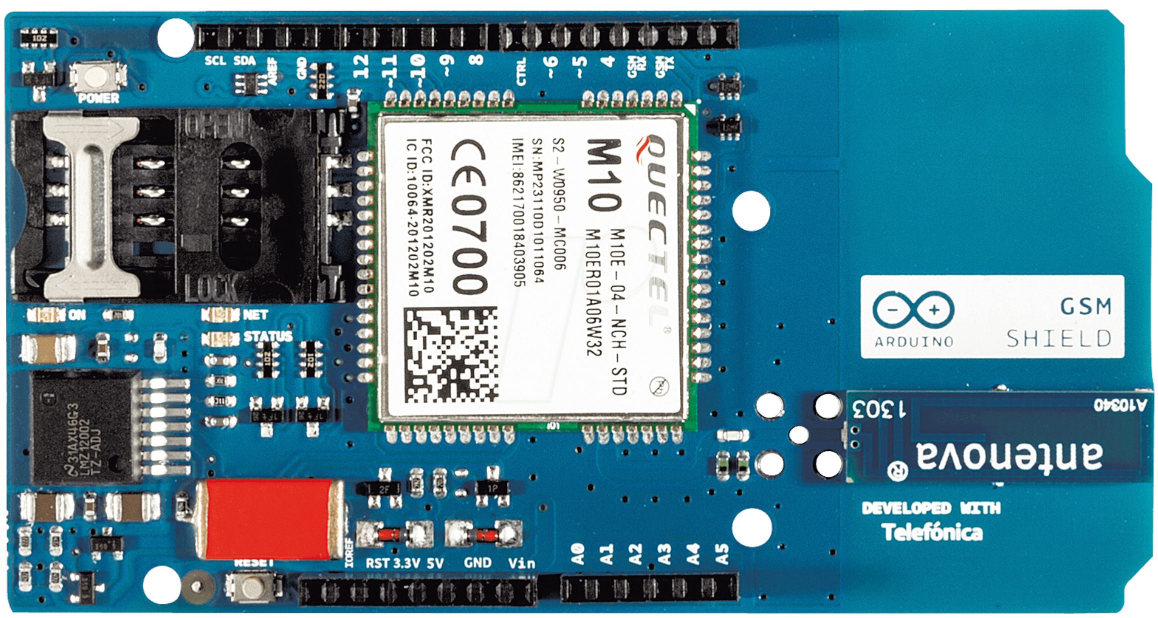

Figure 9.5: GSM Shield used to send SMS messages and interact with device.

Chapter 10

Project Risks

There are a number of risks that our group may encounter throughout the year. Risk level is determined by its overall impact on our project. The impact is specified has high, medium, and low. Risks need to be mitigated in order to ensure the most successful project.

1. High Impact

    (a) **Bugs** are inevitable in any software project however they are unacceptable in medical devices to ensure patient safety.

        i. **Mitigation:** Testing during development.

    (a) **Incorrect Signal Processing** would be devastating for our device because it is impossible to expect correct outcomes when working with faulty data.

        i. **Mitigation:** Cross-checking our algorithms using MATLAB.

    (a) **High Cost** is unacceptable because then our device would not be accessible by our target populations.

        i. **Mitigation:** Being diligent about our budget and spending the necessary time required researching affordable technologies to use.

2. Medium Impact

    (a) **Time:** We don't pay attention and rush our work just before deadlines.

        i. **Mitigation:** Stick to our schedule and meet the deadlines outlined for the project.

3. Low Impact

    (a) **Inactive Group Members:** We fall behind due to illness, unexpected personal obstacles, normal course work.

i. **Mitigation:** Make sure we each know about each aspect of the project. With only two group members, we are expected to be able to work on any part of the project if needed (at least temporarily).

Chapter 11

Results

We chose to analyze the patient's RR interval. As seen in Figure 4.1, The R wave represents the middle point in a person's heartbeat. Therefore, the time duration in between R waves is called the RR interval. By accumulating all of the RR intervals during an ECG test, we were able to look at the average time in between a person's heartbeat. Furthermore, we also calculated the standard deviation of RR intervals. These numeric values are reported in milliseconds as it is standard to do so. We analyzed RR intervals for two reasons. First, the R wave is the easiest to detect because it has the largest amplitude relative to all other parts of the ECG wave. Second, there are two specific medical studies examining the importance of average RR interval and RR standard deviation. The first study determined that young, insulin dependent yet otherwise healthy male diabetics consistently had a shorter average RR interval and smaller RR standard deviation compared to their non-diabetic counterparts [2]. The second study concluded that patients with a very small RR standard deviation just after they had a heart attack often resulted in a high mortality rate. [1]

Although we unfortunately were not able to analyze other parts of the waveform, the C code running on the Galileo that analyzed the data was cross-checked using MATLAB. Figure 9.4 shows a sample text message a doctor would receive containing the patient's average RR interval and RR standard deviation along with their patient ID and test ID for identification purposes.

Chapter 12

Development Timeline

To adequately pace ourselves throughout Senior Design, we have constructed Gantt Charts for each quarter to plot our responsibilities over time.

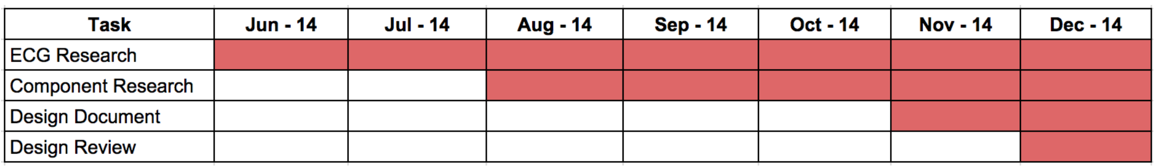

Figure 12.1: Summer and Fall Quarter Development Timeline

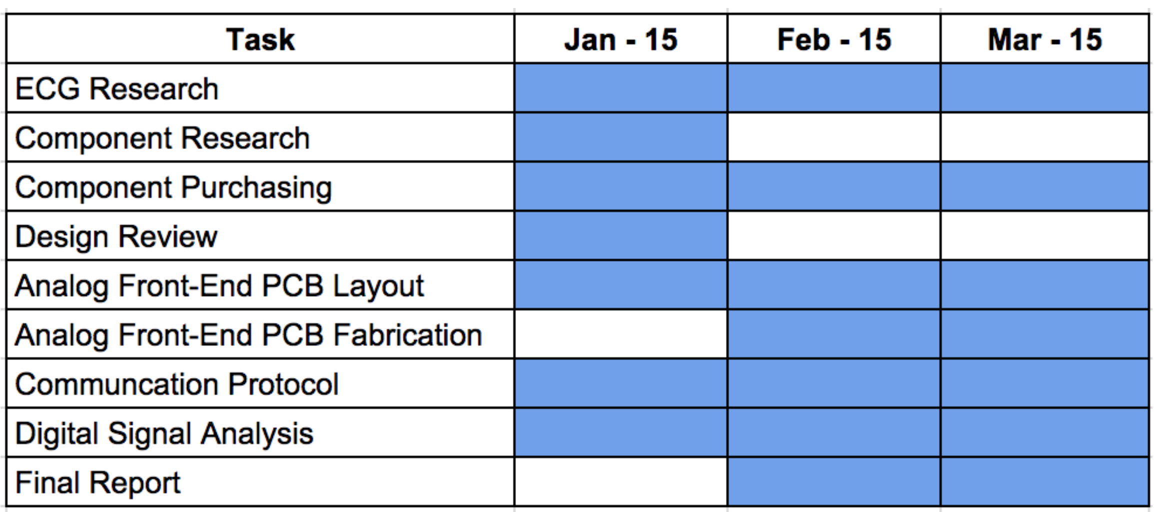

Figure 12.2: Winter Quarter Development Timelime

Task	Apr - 15	May - 15	June - 15
Component Purchasing			
Communcation Protocol			
Digital Signal Analysis			
Final Presentation			
Final Report			

Figure 12.3: Spring Quarter Development Timelime

Chapter 13

Non-Technical Issues

13.1 Ethical

Our device would allow rural communities of developing countries to access healthcare they would otherwise not be able to. Our device would overcome the obstacles that currently exist in providing these peoples with access to healthcare. Our project is inherently ethical; it was developed with constant attention paid to aligning it with the core competencies of the Frugal Innovation Lab (FIL).

13.2 Science, Technology, and Society

Our project has a direct focus on science, technology, and society. We aim to empower underprivileged people by giving them access to healthcare opportunities they previously have not had access to. We received our funding through the Roelandt's grant in SCU's Center for Science, Technology, and Society because we believe they embody what we are trying to accomplish.

13.3 Civic Engagement

A device like ours would most likely need some sort of regulatory approval before being allowed to operate in a country. We would like to research this more with Public Health students next year but this was not within the scope of the project this year.

13.4 Economic

We need our device to be as low-cost as possible in order for it to be accessible by as many people as possible. This means being more affordable than existing technologies such as descried earlier in this thesis.

13.5 Health and Safety

Safety is paramount in every medical device and ours is no different. This is part of the reason we chose the ADS1293EVM because it has electrical safeguards in-place to protect the patient.

13.6 Usability

Our device has been designed to be simple and easy to use in order to minimize error. Misuse could result in incorrect data being sent to a doctor and/or an improper response from a doctor.

13.7 Sustainability

We did not focus on sustainability issues for our project.

13.8 Environmental Impact

We did not focus on environmental issues for our project.

Chapter 14

Conclusions

14.1 Summary

In conclusion, our senior design project is an electrocardiogram device for rural populations in developing countries. We have used an affordable technology stack comprising of the TI ADS1293EVM, Intel Galileo Gen 2, and Arduino GSM shield. Our device frees the doctor from the requirement of being on-site to interpret results. This means that they do not have to leave their normal clinics to go out in the field. Lastly, our device is a screening mechanism, not a hard-and-fast diagnostic tool.

14.2 Difficulties and Lessons Learned

We faced many challenges throughout this project and we were lucky enough to learn and grow from them. First, we had no knowledge regarding the medical aspects of electrocardiograms and engineering technology that went into these devices. We spent much of the fall quarter researching everything we needed to know in order to make the proper decisions in choosing our technologies. In retrospect, this research could have been done over the summer so we could have begun our implementation when the school year started.

Another lesson we learned was that working with off-the-shelf microcontrollers and electroncics boards are not as easy as they might seem. We underestimated the time required to acclimate ourselves with these technologies and subsequently spent time simply familiarizing ourselves with our components.

A final takeaway has been the difficulty in working with small-scale electrical signals. We were unfortunately unable to overcome noise that left the smaller P, Q, S, and T waves unreadable. Because we decided on using the ADS1293EVM, we were naïve in thinking that it would work perfectly right out of the box. Having no experience in this type of work, we did not pay enough respect to the precision and precaution required to sample signals on the microvolt level.

Bibliography

[1] Malik, M., MD. ”Electrophysiology, Pacing, and Arrhythmia.” Clinical Cardiology 13 (1990): 570-76. Print.

[2] Murray, A et al. “RR Interval Variations in Young Male Diabetics.” British Heart Journal 37.8 (1975): 882–885. Print.

[3] CIA: World Fact Book
https://www.cia.gov/library/publications/the-world-factbook/geos/ni.html

[4] American Heart Association: What is Cardiovascular Disease
http://www.heart.org/HEARTORG/Caregiver/Resources/WhatisCardiovascularDisease/What-is-Cardiovascular-Disease

[5] The burden of non-communicable diseases in Nigeria; in the context of globalization
http://www.annalsafrmed.org/article.asp?issn=1596-3519;year=2014;volume=13;issue=1;spage=1;epage=10;aulast=Maiyaki

[6] Nigeria WHO Country Office
http://www.afro.who.int/en/nigeria/who-country-office-nigeria.html

[7] Nigeria Health Systems Strengthening
http://www.cdc.gov/globalhealth/countries/nigeria/what/hss.htm

[8] Nigeria State Health Investment Project (NSHIP) Additional Financing
http://www.worldbank.org/projects/P149936?lang=en

[9] MDGs Survey Report 2015
http://www.ng.undp.org/content/nigeria/en/home/library/mdg/NigeriaMDGsSurveyReport2015/

Appendix A

ADS1293 Setup Through SPI Interface (C Code)

```
#include <SPI.h>

const int pin_DRDYB      = 9;
const int pin_SS         = 10;

void setup()
{
    Serial.begin(9600);

    //configure pinMode for each pin
    pinMode(pin_DRDYB, INPUT);
    pinMode(pin_SS, OUTPUT);

    SPI.begin();

    setupECG();

    SPI.end();
}

void loop()
{

}

void setupECG()
{
    //LOD control register
    bitManip(0x06, 2, 0);
    bitManip(0x06, 3, 0);

    //enable LOD for each channel
    writeRegister(0x07, B00001111);

    //enable clock to digital
    bitManip(0x12, 2, 1);
```

```
    //enable hi-res for each channel
    writeRegister(0x13, B00111111);

    //configure data ready pin source to ECG channel 1
    bitManip(0x27, 1, 1);

    //configure each ECG channel for loop read back mode
    bitManip(0x2F, 4, 1);
    bitManip(0x2F, 5, 1);
    bitManip(0x2F, 6, 1);
}

/*
    params:
        reg: the register to access
        shift_count: number of bit you want to access
        bit_value: what you want the bit to be set to
*/
void bitManip(byte reg, byte shift_count, byte bit_value)
{
    byte reg_data = readRegister(reg);

    byte check_bit = (reg_data >> shift_count) & 1;

    if((bit_value == 0 && check_bit == 1) || (bit_value == 1 &&
       ↪ check_bit == 0)) {
        reg_data ^= 1 << shift_count;
    }

    writeRegister(reg, reg_data);

}

byte readRegister(byte reg)
{
    byte data;

    reg |= 1 << 7; //set read/write bit of command field to 1

    digitalWrite(pin_SS, LOW);   //drive SS pin low to open SPI
       ↪ communication

    SPI.transfer(reg);   //transfer read register address with read bit
       ↪ set
    data = SPI.transfer(0); //transfer dummy data in order to get back
       ↪ the read data from the SPI bus

    digitalWrite(pin_SS, HIGH);   //drive SS pin high to close SPI
       ↪ communication

    return data;   //return byte read from the SPI bus
}
```

```
void writeRegister(byte reg, byte data)
{
    reg &= ~(1 << 7);   //set read/write bit of command field to 0

    digitalWrite(pin_SS, LOW);

    SPI.transfer(reg);
    SPI.transfer(data);

    digitalWrite(pin_SS, HIGH);
}
```

Appendix B

SMS interface C Code

```
//include the GSM library
#include <GSM.h>

//PIN number for the SIM
#define PINNUMBER ""
#define SETUPTEST '0'
#define STARTTEST '1'
#define SENDTEST '2'

//initialize the library instances
GSM gsmAccess;
GSM_SMS sms;

//Array to hold the number a SMS is retreived from
char senderNumber[20];
unsigned int testId;

void setup() {

  //initialize serial communications and wait for port to open
  Serial.begin(9600);

  Serial.println("SMS Messages Reveiver");

  //connection state
  boolean notConnected = true;

  //Start GSM connection
  while (notConnected) {
    if (gsmAccess.begin(PINNUMBER) == GSM_READY)
      notConnected = false;
    else {
      Serial.println("Not connected");
      delay(1000);
    }
  }

  Serial.println("GSM initialized");
```

```
  Serial.println("Waiting for messages");
}

void loop() {

  checkSMS();
  delay(1000);

}

void checkSMS() {
  char c;
  char txtMsg[200];
  memset(txtMsg, 0, sizeof(txtMsg));
  if (sms.available()) {
    Serial.println("Messages recieved from:");

    //Get remote number
    sms.remoteNumber(senderNumber, 20);
    Serial.println(senderNumber);

    //Read message bytes and print them
    int i = 0;
    while (c = sms.read()) {
      Serial.print(c);
      txtMsg[i] = c;
      i++;
    }

    Serial.println("\n END OF MESSAGE");

    //Delete message from modem memory
    sms.flush();
    Serial.println("MESSAGE DELETED");

    checkCommand(txtMsg, senderNumber);
  }
}

void sendSMS(char txtMsg[200], char remoteNum[20]) {

  //sms text
  Serial.println(remoteNum);
  Serial.println("SENDING");
  Serial.println();
  Serial.println("Message:");
  Serial.println(txtMsg);

  //send the message
  sms.beginSMS(remoteNum);
  sms.print(txtMsg);
  sms.endSMS();
  Serial.println("\nCOMPLETE!\n");
}
```

```
void checkCommand (char txtMsg[256], char phoneNumber[20]) {
  switch (txtMsg[0]) {
    case '0' :
      Serial.println("Setup Test");
      setupTest(txtMsg, phoneNumber);
      break;

    case '1' :
      Serial.println("Start Test");
      startTest(txtMsg);
      break;

    case '2' :
      Serial.println("Send Test");
      sendTest(txtMsg);
  }
}

void setupTest(char txtMsg[200], char phoneNumber[20]) {

  char * pch = strtok (txtMsg, ",");
  int count = 0;
  char * patientId;
  while (pch != NULL) {
    if (count == 1)
      patientId = pch;
    pch = strtok (NULL, ",");
    count++;
  }
  Serial.println(patientId);

  char *patientDir = "/home/root/senior_design/patients/";

  char * _mkdir = "mkdir -p ";

  char *command = (char *)malloc(strlen(_mkdir) + strlen(patientDir) +
     ↪ strlen(patientId) + 1);
  memset(command, 0, strlen(command));
  strcat(command, _mkdir);
  strcat(command, patientDir);
  strcat(command, patientId);

  Serial.print("Command: ");
  Serial.println(command);
  system(command);

  //generate/read test ID and strcat onto end of patientDir
  FILE *fp;

  fp = fopen("/home/root/senior_design/patients/test_id_log.txt", "r+");
  char buffer[50];

  fscanf(fp, "%d", &testId);
```

```
  fclose(fp);
  testId++;

  // Write incremented testId to file and close
  fp = fopen("/home/root/senior_design/patients/test_id_log.txt", "w+");
  fprintf(fp, "%d", testId);
  fclose(fp);

  Serial.print("testId: ");
  Serial.println(testId);
  char msg[200];
  memset(msg, 0, sizeof(msg));
  sprintf(msg, "%d", testId);

  memset(patientId, 0, strlen(patientId));
}

void startTest(char txtMsg[200]) {
  char * pch = strtok (txtMsg, ",");
  int count = 0;
  char * testId;
  while (pch != NULL) {
    if (count == 1)
      testId = pch;
    pch = strtok (NULL, ",");
    count++;
  }

  Serial.println(testId);

}

void sendTest(char txtMsg[200]) {
  //example text message format:
  //command, patientId, testId, docNumber

  int patientId_is_present = 0;
  int testId_is_present = 0;
  FILE *fp;


  char* rr_command = "./rr ";

  char * pch = strtok (txtMsg, ",");
  int count = 0;
  char * testId;
  char * docNum;
  char * patientId;
  while (pch != NULL) {
    if (count == 1)
      patientId = pch;
    if (count == 2)
      testId = pch;
    if (count == 3)
```

```
    docNum = pch;
  pch = strtok (NULL, ",");
  count++;
}

char* patientId_command = (char*)calloc(512, sizeof(char));
sprintf(patientId_command, "cd /home/root/senior_design/patients/ && [ -e %s ] && echo \"1\" > /home/root/senior_design/patients/patientId_is_present.txt || echo \"0\" > /home/root/senior_design/patients/patientId_is_present.txt", patientId);


Serial.print("patientId = ");
Serial.println(patientId);
Serial.print("testId = ");
Serial.println(testId);

Serial.print("patientId_command = ");
Serial.println(patientId_command);

system(patientId_command);

fp = fopen("/home/root/senior_design/patients/patientId_is_present.txt", "r");
fscanf(fp, "%d", &patientId_is_present);
fclose(fp);

if (patientId_is_present == 1) {
  //cd into the patient directory and check if testId is there
  char* testId_command = (char*)calloc(512, sizeof(char));
  sprintf(testId_command, "cd /home/root/senior_design/patients/%s && [ -e %s.txt ] && echo \"1\" > /home/root/senior_design/patients/testId_is_present.txt || echo \"0\" > /home/root/senior_design/patients/testId_is_present.txt", patientId, testId);



  Serial.print("testId_command = ");
  Serial.println(testId_command);

  system(testId_command);

  //check if testId is there
  fp = fopen("/home/root/senior_design/patients/testId_is_present.txt", "r");
  fscanf(fp, "%d", &testId_is_present);
  fclose(fp);

  if (testId_is_present == 1) {

    //execute ./rr
    char* rr_command = (char*)calloc(256, sizeof(char));
```

```
        sprintf(rr_command, "cd home/root/senior_design/patients/%s && ./
            ↪ rr %s.txt %s_out.txt", patientId, testId, testId);

        Serial.print("rr_command = ");
        Serial.println(rr_command);

        system(rr_command);

        char* test_out_path = (char*)calloc(256, sizeof(char));

        sprintf(test_out_path, "home/root/senior_design/patients/%s/%s_out
            ↪ .txt", patientId, testId);

        fp = fopen(test_out_path, "r");
        char* test_out = (char*)calloc(256, sizeof(char));

        fseek(fp, 0, SEEK_END);
        unsigned len = ftell(fp);
        fseek(fp, 0, SEEK_SET);
        test_out = (char*)calloc(len + 1, sizeof(char));
        fread(test_out, 1, len, fp);
        fclose(fp);

        char* test_info = (char*)calloc(256, sizeof(char));

        sprintf(test_info, "patient ID = %s\ntest ID = %s\n", patientId,
            ↪ testId);

        char* test_sms = (char*)calloc(256, sizeof(char));

        sprintf(test_sms, "%s\n%s", test_info, test_out);

        Serial.println("test_out: ");
        Serial.println(test_sms);

        sms.beginSMS(docNum);
        sms.print(test_sms);
        sms.endSMS();
      }
      else {
          //do nothing
      }
    }
    else {
          //do nothing
    }
}
```

Appendix C

RR Interval C Code

```
#define MAX_SAMPLES 20000
#define MAX_R_PEAKS 20

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <math.h>


unsigned findMaxRPeaks(unsigned);
unsigned max(unsigned);
void quickSort(int, int);
int partition(int, int);
double findStandardDeviation(unsigned, FILE*);


struct DataSample {
    unsigned sampleNumber;
    double timeStamp;
    double voltage;
};

struct DataSample samples[MAX_SAMPLES];
struct DataSample r_peaks[MAX_R_PEAKS];
double rr_intervals[MAX_R_PEAKS];

int main(int argc, char *argv[])
{
    FILE *fp;

    char buffer[50];
    char *token;

    unsigned i = 0;
    bool neg_flag = false;

    unsigned num_samples = 0;
```

```c
    fp = fopen("results.txt", "r");

    while(fgets(buffer, sizeof(buffer), fp) != NULL) {

        token = strtok(buffer, ",");
        samples[num_samples].sampleNumber = strtoul(token, NULL, 0);
        token = strtok(NULL, ",");
        samples[num_samples].timeStamp = atof(token);
        token = strtok(NULL, ",");

        if(strstr(token, "-")){
            //ignore the negative values
        }
        else {
            samples[num_samples].voltage = atof(token);
        }

        num_samples++;
    }


    fclose(fp);

    unsigned num_r_peaks = findMaxRPeaks(num_samples) - 1;

    //sort R peaks based on timeStamp
    quickSort(0, num_r_peaks);

    fp = fopen(argv[1], "w+");

    fprintf(fp, "num_r_peaks = %d \n", num_r_peaks);

    for(i = 0; i < num_r_peaks; i++) {
        rr_intervals[i] =  r_peaks[i+1].timeStamp - r_peaks[i].timeStamp
            ↪ ;
        fprintf(fp, "RR Interval = %lf \n", rr_intervals[i]);
    }

    fprintf(fp, "standard deviation = %lfms \n", 1000*
       ↪ findStandardDeviation(num_r_peaks, fp));

    return 0;
}

double findStandardDeviation(unsigned num_r_peaks, FILE *fp)
{
    unsigned i = 0;
    double avg = 0;
    double std_deviation = 0.0;
    double variance = 0.0;

    for(i = 0; i < num_r_peaks; i++) {
        avg += rr_intervals[i];
    }
```

```
    avg = avg / num_r_peaks;

    fprintf(fp, "average = %lfms \n", 1000*avg);

    for(i = 0; i < num_r_peaks; i++) {
        variance = (avg - rr_intervals[i]) * (avg - rr_intervals[i]);
    }

    std_deviation = sqrt(variance);

    return std_deviation;
}

unsigned findMaxRPeaks(unsigned num_samples)
{
    unsigned i = 0;
    unsigned saved_index;
    unsigned r_peaks_index = 0;

    do {

        //get max
        saved_index = max(num_samples);

        r_peaks[r_peaks_index].voltage = samples[saved_index].voltage;
        r_peaks[r_peaks_index].timeStamp = samples[saved_index].
            ↪ timeStamp;
        r_peaks[r_peaks_index].sampleNumber = samples[saved_index].
            ↪ sampleNumber;

        r_peaks_index++;

        for(i = saved_index + 8; i >= saved_index; i--) {
            samples[i].voltage = 0;
        }
        for(i = saved_index - 8; i <= saved_index; i++) {
            samples[i].voltage = 0;
        }

    } while(r_peaks[r_peaks_index -1].voltage > 0.0025);
    //0.25 is our voltage threshold on R peak

    return r_peaks_index;
}

unsigned max(unsigned num_samples)
{
    double max = samples[0].voltage;
    unsigned i = 0;
    unsigned saved_index;

    for(i = 0; i < num_samples; i++) {
        if(samples[i].voltage > max) {
            max = samples[i].voltage;
```

```
            saved_index = i;
        }
    }

    return saved_index;
}

void quickSort(int l, int r)
{
   int j;

   if( l < r )
   {
        // divide and conquer
       j = partition(l, r);
       quickSort(l, j-1);
       quickSort(j+1, r);
   }
}

int partition(int l, int r) {
   double pivot;
   int i, j;
   struct DataSample t;
   pivot = r_peaks[l].timeStamp;
   i = l;
   j = r+1;

   while(1)
   {
       do
          ++i;
       while( r_peaks[i].timeStamp <= pivot && i <= r );
           do
          --j;
       while( r_peaks[j].timeStamp > pivot );
           if( i >= j )
          break;
       t = r_peaks[i];
       r_peaks[i] = r_peaks[j];
       r_peaks[j] = t;
   }

   t = r_peaks[l];
   r_peaks[l] = r_peaks[j];
   r_peaks[j] = t;

   return j;
}
```