



University  
of Windsor

## Cardless Security System

SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE COURSE

**88-400-01: CAPSTONE DESIGN PROJECT**

FINAL REPORT

*Department of Computer and Electrical Engineering*

August 3rd, 2018

# **Cardless Security System**

**Goleberski, Robert** 104262016

**Santos, Matthew** 103103503

**Li, Zuwei** 104165063

*Department of Computer and Electrical Engineering*

**UNIVERSITY OF WINDSOR**

Submitted August 3rd, 2018

# Cardless Security System

No action by any design team member contravened the provisions of the Code of Ethics and we hereby reaffirm that the work presented in this report is solely the effort of the team members and that any work of others that was used during the execution of the design project or is included in the report has been suitably acknowledged through the standard practice of citing references and stating appropriate acknowledgments. The presence of the author's signatures on the signature sheet means that they are affirming this statement."

Santos Matthew	103103503		August 3 <sup>rd</sup> , 2018
Li Zuwei	104165063		August 3 <sup>rd</sup> , 2018
Golebenski Robert	104262016		August 3 <sup>rd</sup> , 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Background</b>	<b>7</b>
<b>3</b>	<b>Project Description</b>	<b>8</b>
3.1	Design Objectives . . . . .	8
3.2	Project Schedule . . . . .	8
3.3	Proposed Budget . . . . .	9
<b>4</b>	<b>Prototype Design</b>	<b>10</b>
4.1	Analog Front End . . . . .	10
4.2	Microprocessor Circuit . . . . .	14
4.3	Power Distribution System . . . . .	15
4.3.1	Motor Controller . . . . .	20
4.3.2	Voltage Indicator . . . . .	23
<b>5</b>	<b>Biometric Algorithm</b>	<b>23</b>
5.1	Prepossessing . . . . .	24
5.2	Feature Identification . . . . .	25
5.3	Classifier Creation . . . . .	26
5.4	Simulated Results . . . . .	26
<b>6</b>	<b>Prototype Fabrication</b>	<b>27</b>
6.1	PCB Layout Design . . . . .	27
6.2	Reflow Soldering . . . . .	29
6.3	Improvised Modifications . . . . .	31
6.4	Implementation Cost . . . . .	32
<b>7</b>	<b>Prototype Evaluation</b>	<b>32</b>
<b>8</b>	<b>Recommendations</b>	<b>33</b>
<b>9</b>	<b>Conclusions</b>	<b>34</b>
<b>References</b>		<b>35</b>
<b>Appendices</b>		<b>37</b>
<b>A</b>	<b>Circuit Schematics</b>	<b>38</b>
<b>B</b>	<b>Bill of Materials</b>	<b>43</b>
<b>C</b>	<b>Project Schedule</b>	<b>44</b>
<b>D</b>	<b>Octave SVM Generator Script</b>	<b>45</b>

<b>E Octave SVM Testing Script</b>	<b>46</b>
<b>F Examples of Custom Octave Sub-functions</b>	<b>47</b>
<b>G MPU Embedded Code Main Loop</b>	<b>48</b>
<b>H Examples of an Embedded Function and Interrupt</b>	<b>49</b>

# List of Figures

1	Standard Human Electrocardiogram Profile [1]	6
2	Overall Design Architecture	10
3	Circuit Protection and Noise Filtering	11
4	Optoisolator	11
5	Custom Designed Instrumental Amplifier	12
6	High-pass Filter with 0.5Hz cut-off frequency	12
7	Cascade of Second Order Low-pass Filters with $f_c = 100\text{Hz}$	13
8	DC Offset Injection	13
9	Thermistor Temperature Sensor	14
10	Analog Front End Net Transfer Function	14
11	SD Card Data Transfer Methodology	15
12	Pulse Width Modulation Methodology [2]	16
13	Flyback Converter Schematic [3]	17
14	Flyback Converter Configurations [3]	17
15	Flyback Converter Timing Diagram [3]	18
16	Step Down Regulator Circuit [4]	19
17	Inductor Current Waveform [4]	19
18	Effect of Input Capacitor on Rectification [4]	19
19	Motor Controller Schematic	20
20	Internal Design of a 555 Timer [5]	20
21	H-Bridge Principle of Operation [6]	21
22	H-Bridge Circuit [7]	22
23	Motor Controller Timing Diagram	22
24	Voltage Indicator Circuit Schematic	23
25	Biometric Algorithm Architecture	24
26	FScores for the 27 Most Prominent Features	25
27	Support Vector Machine Manual 2D Classifier Analysis	26
28	Completed Prototype	27
29	Completed PCB Design Layout	28
30	PCB Layout with 3D Part Models Included	29
31	Using the Stencil to Apply Solder Paste in Preparation for Reflow	30
32	Reflowing using Household Toaster Oven	31
33	Example of PCB Trace Modifications	32
34	Recovered ECG Signal under Ideal Conditions	33

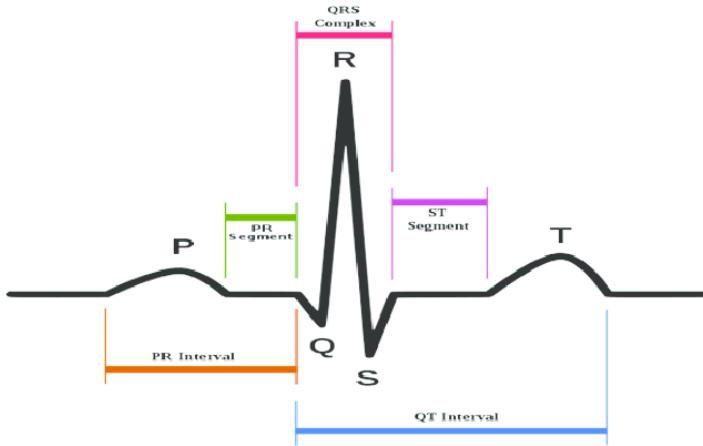
# 1 Introduction

Every building is equipped with some sort of security system. This ranges from simple locks to alarms that go off at unauthorized entry. Most security measures make use of a key that would allow access. Although they are very reliable, they can be lost or stolen and reproduced causing an inconvenience at best or stolen property/data at worst. An innovation tackling this problem, as well as increasing the complexity of the key, is biometric security. Metal keys and cards are being replaced by peoples own fingerprints, eyes and voice.

Biometrics offer a new way of handling security. Not only does it offer the convenience of being the key meaning it is always with a person it provides a difficult key to replicate offering a more secure system. The issue with current systems is that they are expensive and still have a possibility of being duplicated allowing for an intruder to bypass the security measures in place. A fingerprint can be replicated with gelatin or a piece of tape. Biometric systems relying on the face can be tricked with photos from Facebook. A solution that is being proposed with this project is to add a new measure to the list: the heartbeat.

With every heartbeat tiny electric signals are sent throughout the hearts muscles. These signals can be recorded through the process of electrocardiography by placing electrodes on the skin. The structure of this signal is unique to every individual as there are multiple characteristics that are shown in Figure 1. These characteristics can vary from person to person offering a potential for a unique key to use for a biometric security system.

Figure 1: Standard Human Electrocardiogram Profile [1]



This project is composed of separate parts, each providing an essential function in order to realize this system. First of all, there is the analog front end, which filters the signal as well as providing an amplification so that it can be read by the microprocessor, which is the next part of the circuit. The microprocessor uses a support vector machine in order to test the input heartbeat with others stored in the database. If the heartbeat detected is one that matches in the database the microprocessor will then send a trigger to the motor control, signalling to begin the process of unlocking the door. All of this is powered through a train of batteries that is converted to meet the voltage and current requirements of each part.

## 2 Background

The term biometrics comes from the Greek words bio meaning life and metrics meaning to measure. This idea of measuring humans and recording their characteristics for identification purposes goes back to early civilizations. For example, early Egyptian traders would be identified by their physical description to differentiate between trusted and untrusted traders as well as those new to the market. On the other hand fingerprint identification dates back to the Persian empire in the 14th century.

A more systematic system of biometrics was not introduced until 1858 in India. Here, Sir William Herschel documented handprints of his workers in order to make sure that the right employee was being paid. Then in 1870 Alphose Bertillon developed a method of identifying criminals based on detailed records of their physical attributes. This was used to tell whether or not a criminal was a repeat offender in order to give a harsher sentence. Although it was used around the world, it quickly faded into obscurity when it was discovered that the measurements that they were making were shared among criminals. It was not until the 1890s that fingerprint classification systems were developed. In 1892 Sir Francis Galton found a method to classify the prints from all ten fingers; although it was not until 1896 that an efficient method of storing the information was made. These methods just like the one before were also used in order to identify criminals. In 1903 that New York State prisons began using fingerprints for criminal identification laying the foundation for the rest of the prisons to start to use them. It was not until 1969 that a push towards automation was seen. The FBI contracted a third party to speed up the process of fingerprint identification as it required too much time to identify a single individual. The two key challenges faced were scanning and identifying fingerprint cards as well as comparing and matching the fingerprints. In 1975 the FBI continued its push towards biometrics by increasing its funding towards the digitizing of fingerprint identification. The development would continue for the next few decades bringing along new technologies and improvements to replace the old methods. In 1994 the FBI held a competition in order to decide as to who to award a contract to develop their Integrated Automated Fingerprint Identification System (IAFIS) that is still in use today. The contract was awarded to Lockheed Martin and the company finished its development in 1999.

The foundation of the electrocardiogram was laid in 1838. At the University of Pisa, Carlo Matteucci was studying the phenomenon of electric current inside living things. As with many of his companions and predecessors, this work was done on frogs. Carlo discovered that each of the frogs heartbeat had an accompanying electric current. This would shed some light on previous works on animal electricity as scientists beforehand discovered that sending electricity through the muscles of dissected animals would cause them to twitch. In 1843 Emil Du bois-Reymond discovered small voltage potential in a resting muscle that would diminish when the muscle contracted. Although his theories as to the why were later proven wrong his work started what is known as electrophysiology which is the study of the flow of ions in biological tissue. He also developed one of the worlds most sensitive galvanometers for the given time in order to study this phenomenon. Following up on Emils work, in 1856 two scientists in Germany were able to measure electric current across a human heart. This confirmed Matteuccis previous findings that the body was an organism driven by electrical impulses. In 1878 the work on the heart was further expanded on as two British physiologists

were able to record the hearts electrical current showing it consists of two phases which would later be labelled as QRS and T. In 1895 Einthoven further improved on this concept using improved measurement devices as he was able to identify 5 deflections of the electrical current from the heartbeat which he names P, Q, R, S and T. A few years later, in 1901 he developed the string galvanometer which was the first machine to produce electrocardiograms that were reliable and easily read by medical professionals. This machine required the patient to submerge their hands and one of their legs in separate buckets of saline solution which would act as the electrodes. It was not until 1932 that a more practical method of applying electrodes to a person was used. This method was the use of chest leads which resemble small patches that would act as the electrodes. Over the years this concept would be further developed as the most efficient placements of leads were yet to be discovered.

In 1977 the United states military were developing biometric methods and they discovered that a modified ECG signal can be used. This is the first instance of ECG being seeing usage for biometrics. Although this study only had a size of 30 participants a study in 1990 done at the Royal Victoria Hospital done with 900 patients confirmed the idea. It was found that the reason the ECG signal differentiates between people is the result of variation in the ion-paths, blockages and other differences in an individuals body due to their lifestyle.

## 3 Project Description

Nowadays, efficiently protecting personal privacy is getting more and more important. Current modern methods of tackling this problem are regarded as either too expensive or lacking in strength. Electrocardiogram biometric systems are a new field of research with potential to improve upon existing systems while also serving to increase the complexity of the authentication token. This project seeks to create a prototype security system based upon this new technology with the implementation focus being on a practical electronic door locking unit.

### 3.1 Design Objectives

With the problem firmly established the following design requirements were outlined in order to assist in quantifying the success level of the project upon completion.

1. Differentiate between individuals based on electrocardiogram signals
2. Develop a secure mechanical locking system driven by an electric motor

### 3.2 Project Schedule

Overall the project was setup for completion within an 8 month period starting in January 2018. The first few months were dedicated to the formulation of a formal project proposal and the time afterwards was split between design and implementation. A summary of the project schedule is provided in Table 1, for further details on the project timeline see Appendix C.

*Table 1: Gantt Chart Description*

Task Name and Description	Start Date	End Date	Duration	Status
Choosing the Capstone Project Project officially get approved and started on January 13 <sup>th</sup> /2018	January 8 <sup>th</sup>	January 13 <sup>th</sup>	15 hours	Complete
<b>Proposal</b>				
Gather reference and define goals	January 30 <sup>th</sup>	February 25 <sup>th</sup>	40 hours	Complete
Generate Tasks and discuss budget	February 2 <sup>nd</sup>	February 11 <sup>th</sup>	30 hours	Complete
Divide Tasks for group members	February 11 <sup>th</sup>	February 11 <sup>th</sup>	2 Hours	Complete
Approve Proposal	February 20 <sup>th</sup>	February 25 <sup>th</sup>	15 hours	Complete
Submit Proposal	February 25 <sup>th</sup>	February 25 <sup>th</sup>	1 hour	Complete
<b>Circuit Schematic Design</b>				
Generate Block Diagram	February 16 <sup>th</sup>	February 26 <sup>th</sup>	20 Hours	Complete
Power System	February 20 <sup>th</sup>	February 26 <sup>th</sup>	15 hours	Complete
Sensor Circuitry	February 24 <sup>th</sup>	March 5 <sup>th</sup>	30 hours	Complete
DSP Processing	February 20 <sup>th</sup>	March 30 <sup>th</sup>	60 hours	Complete
Additional Features	February 26 <sup>th</sup>	March 10 <sup>th</sup>	15 hours	Complete
Finish Circuit Design	February 20 <sup>th</sup>	April 20 <sup>th</sup>	40 hours	Complete
<b>Hardware Design</b>				
Mechanical Design	April 1 <sup>st</sup>	April 18 <sup>th</sup>	30 hours	Complete
Prepare Progress Report	April 5 <sup>th</sup>	April 6 <sup>th</sup>	8 hours	Complete
Submit Progress Report	April 6 <sup>th</sup>	April 6 <sup>th</sup>	2 hours	Complete
Part Selection	April 5 <sup>th</sup>	April 20 <sup>th</sup>	25 hours	Complete
PCB Layout	April 20 <sup>th</sup>	May 15 <sup>th</sup>	80 hours	Complete
<b>Simulation</b>				
Analog Front End Simulation	May 1 <sup>st</sup>	May 10 <sup>th</sup>	15 hours	Complete
Microprocessor Simulation	May 4 <sup>th</sup>	June 15 <sup>th</sup>	60 hours	Complete
Mechanical Locking Simulation	May 5 <sup>th</sup>	May 20 <sup>th</sup>	10 hours	Complete
Project Verification	June 5 <sup>th</sup>	July 10 <sup>th</sup>	85 hours	Complete
<b>Prototype Assembly</b>				
Prototype Construction	June 20 <sup>th</sup>	July 20 <sup>th</sup>	120 hours	Complete
Final Stage Testing	July 15 <sup>th</sup>	July 20 <sup>th</sup>	30 hours	Complete
<b>Final Presentation</b>				
Create Presentation Materials	July 20 <sup>th</sup>	July 23 <sup>rd</sup>	5 hours	Complete
Presentation Poster	July 22 <sup>nd</sup>	July 25 <sup>th</sup>	5 hours	Complete
Practice Presentation	July 25 <sup>th</sup>	July 26 <sup>th</sup>	2 hours	Complete

### 3.3 Proposed Budget

After some research into the field of biometrics the following budget outline was proposed. Significant effort was made throughout the design process to ensure adherence to these budget restrictions.

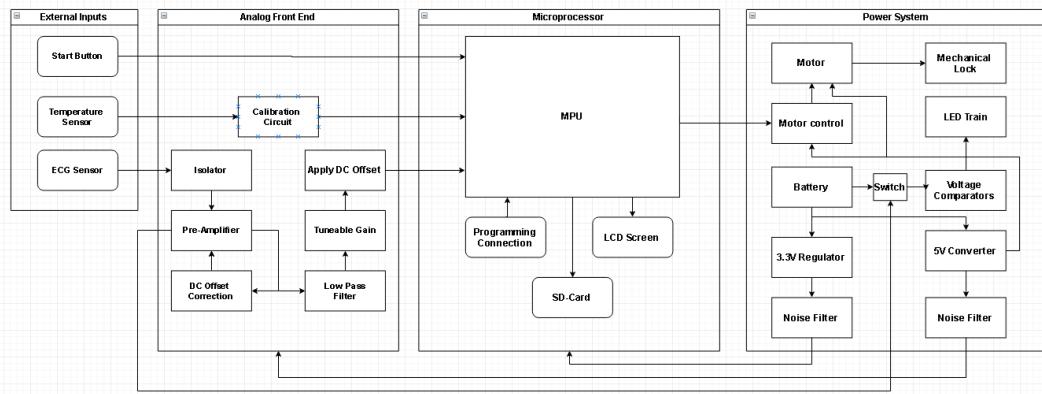
- \$75 for Microprocessing unit including peripherals
- \$100 for Electromechanical systems and all power distribution components

- \$100 for Analog front end including all sensors and user control inputs
- \$50 for Printed circuit board manufacturing and miscellaneous expenditures
- Grand Total of \$325 which allows for reasonable design flexibility

## 4 Prototype Design

In order to fulfill the design requirements the overall project was first separated into smaller more manageable sub-projects. These sub-projects include the development of an analog front end for obtaining the real world data, a microprocessing unit for performing data analysis, and a power distribution system for facilitating these operations. A brief overview of the design can be seen in Figure 2 which outlines the various components in each sub-project and as well as their interconnections. The sections that follow endeavor to elaborate on the functionality of these specific design elements. For an even more detailed overview of the circuit design see Appendix A for the complete circuit schematics.

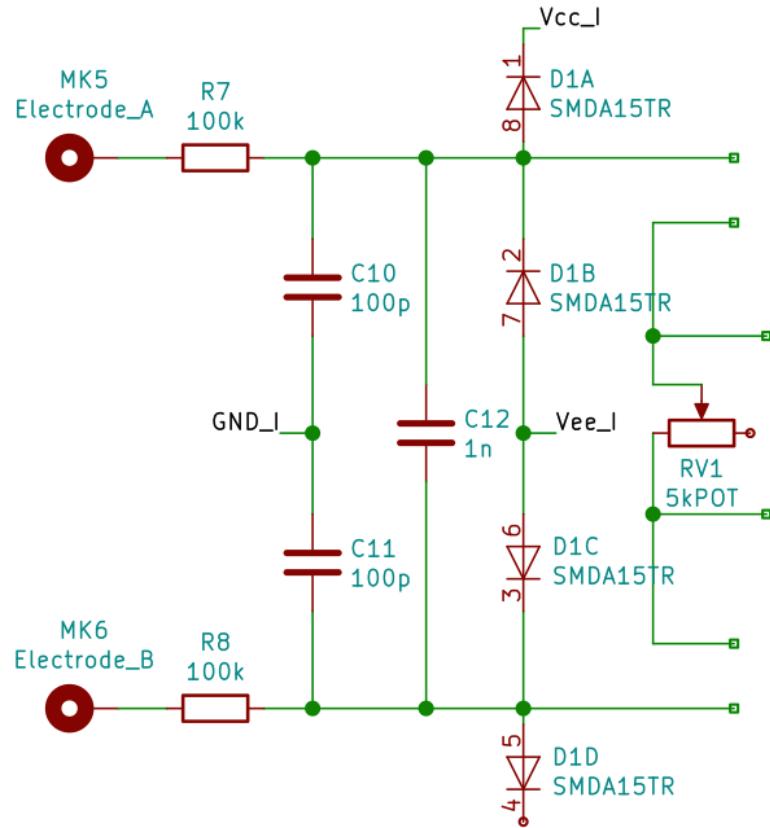
Figure 2: Overall Design Architecture



### 4.1 Analog Front End

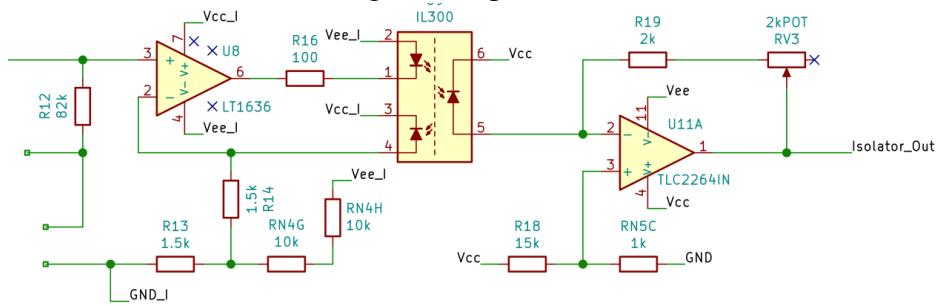
This portion of the project was responsible for providing the interface to and refinement of real world signals before directing them to the microcontroller for acquisition and processing. First of all, getting the ECG signal from the fingers is very weak, in order for further processing accuracy, the signal has to be amplified before it gets to the filters stage. Also, As the measurement of real signals is always accompanied by noise preliminary analog filters were strategically implemented in order extract clear ECG signals. The analog ECG processing consists of an instrumentation amplifier which is used for its low noise gain and isolation.

Figure 3: Circuit Protection and Noise Filtering



Because of this specific part of the design has direct contact with human body, safety measures will be the first priority. The team has this problem solved by adding the Circuit Protection design in the front of the Instrumentation amplifier, in order to protect human body from getting electric shock.

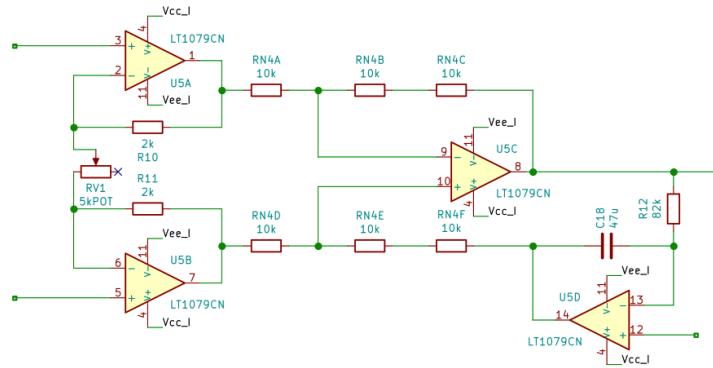
Figure 4: Optoisolator



Also in order to protect the circuit getting damaged from high current going through, a bipolar optoisolator was used in the circuit design. This component will prevent high current going through the circuit, as we are getting the same amount of current as the input.

Filters could have a significant effect on the test results of ECG signal processing. Exact and clear ECG signals will be needed in order to identify an individual, therefore different filters will be applied in the signal processing unit. These filters are going to be used to remove those unwanted noises. In this ECG signal processing unit, instrumentation amplifier, high-pass filter and low-pass filter are been used, in order to get to reduce frequencies where noise occurs, while allowing the signal frequency through.

Figure 5: Custom Designed Instrumental Amplifier



The ECG signal levels are very small, in order to process the signal an amplifier will be needed. The design of this first order amplifier has been set the gain of 18. This will allowed the signal be viewed clear within the voltage of 3V to 0.

Figure 6: High-pass Filter with 0.5Hz cut-off frequency

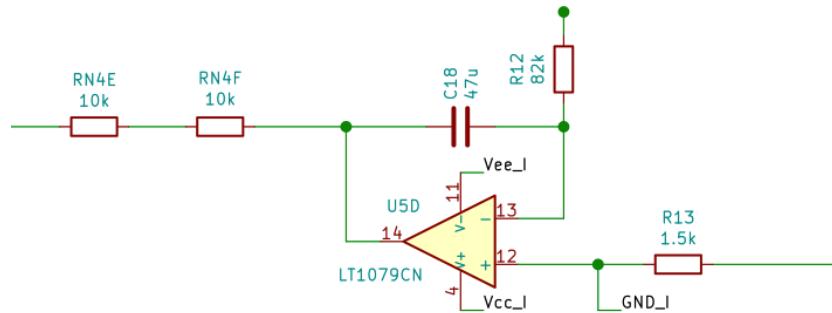
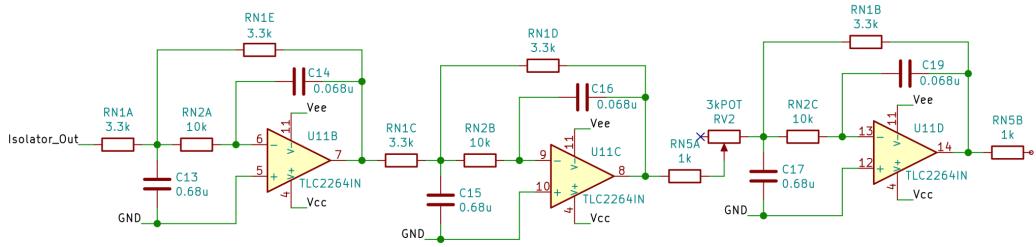
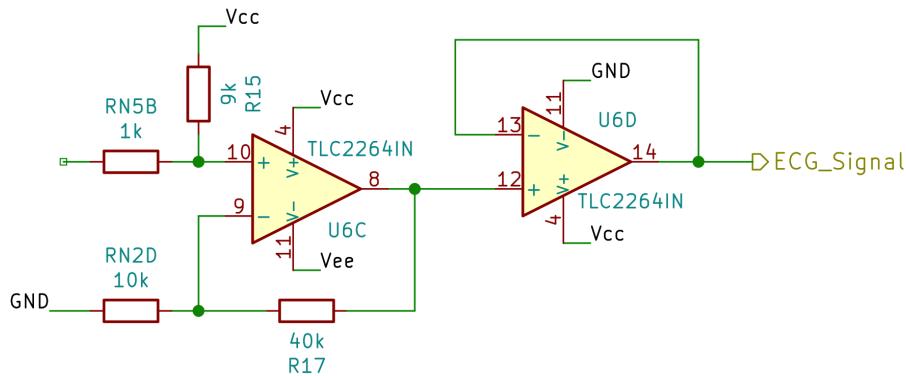


Figure 7: Cascade of Second Order Low-pass Filters with  $f_c = 100\text{Hz}$



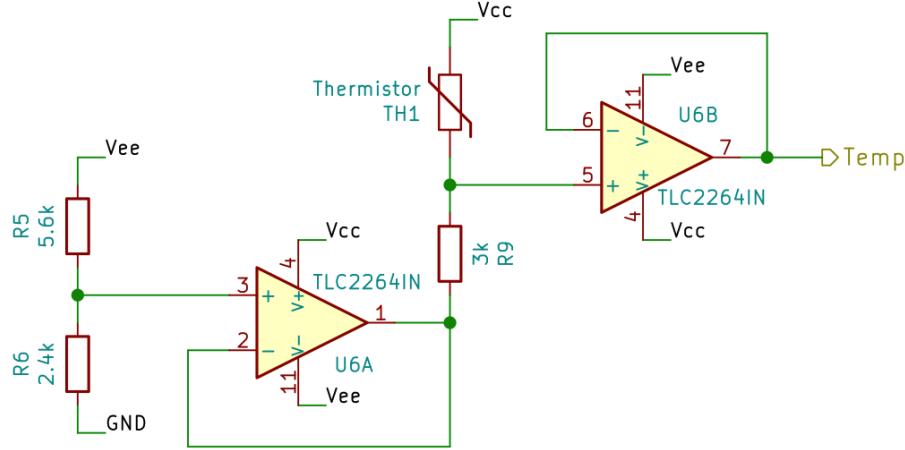
Based on research conducted it was found that human ECG signals produce P and T wave frequencies between 0.5 to 10 Hz and QRS complex frequencies between 4 and 20 Hz. In order to get a clearer signal a first-order high-pass filter configured to provide reference feedback to the instrumental amplifier for low-frequency noise removal and lastly a pair of 2nd order low-pass filters which have been present to a cut-off frequency of 100Hz.

Figure 8: DC Offset Injection



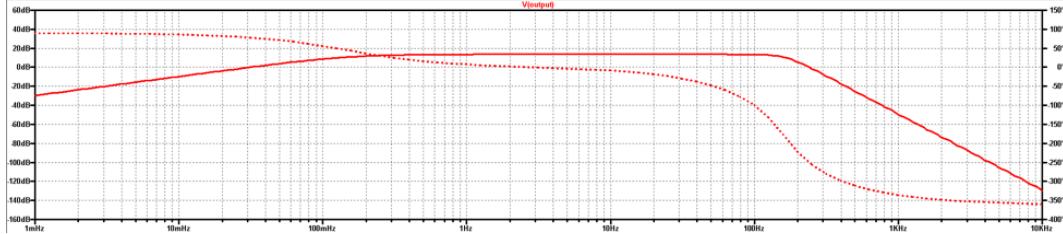
For further processing, in order to make sure the accuracy of the calculation, the team has to set the signal be viewed clear within the voltage of 3V to 0. This DC injection will offset the signal and keep it within the range for future calculation.

Figure 9: Thermistor Temperature Sensor



For the better performance and security enhancement, an additional temperature sensor will be installed for this system. This temperature sensor will prevent non-human body signal going into the circuit, any digital copy of the ECG signal will not be accepted.

Figure 10: Analog Front End Net Transfer Function

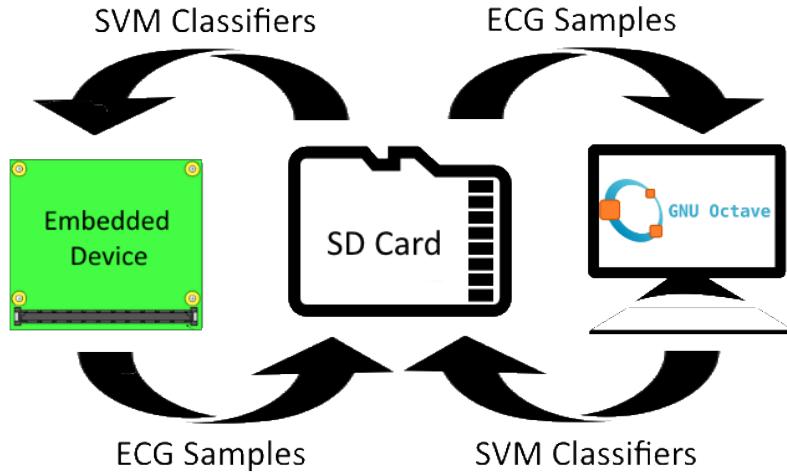


## 4.2 Microprocessor Circuit

In order to perform in situ calculations a suitable microprocessing unit was required. In addition there was need for peripherals in order to facilitate the transfer of information between the processing unit and the user. After careful review the dsPIC33FJ128GP802 processor from Microchip was selected for its enhanced digital signal processing capabilities and suitably large memory which also included a sizable region capable of direct memory addressing. For peripherals the proposed budget prevented the acquisition of more advanced display modules and so instead a simple character based LCD screen was adopted. This was utilized for the displaying of important user prompts as well as to illustrate the stage of analysis currently being executed for the purpose of illustrating progress. Lastly a method of easily loading in or modifying the existing user database was needed. This was accomplished by the addition of an SD card interface which was connected through SPI directly to the processor. The methodology behind the SD card is illustrated in Figure 11 and shows how the card is used to facilitate the generation of the database by offloading the job to a more powerful desktop computer. It is noted that all peripheral components were specifi-

cally selected to operate under identical supply requirements to assist in reducing the overall design complexity.

Figure 11: SD Card Data Transfer Methodology



In addition to part selection and circuit design there was also a significant amount of embedded programming involved in the creation of the device. Complete inclusion of all the source code in this report is unfeasible so instead select critical portions of the code have been included and are illustrated in Appendix G and H respectively.

### 4.3 Power Distribution System

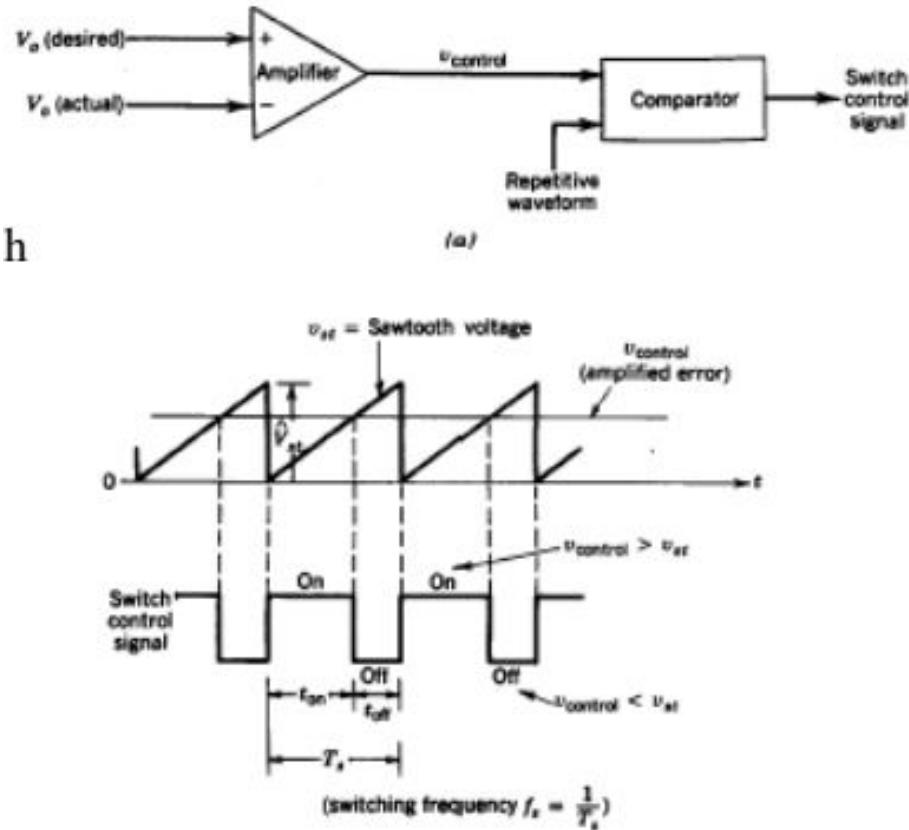
As with all electric circuits, providing power to all the components is an important issue that needs to be resolved. For this project a DC power source coming from 4 AA batteries was chosen. The battery pack is connected in series with a fuse in order to protect the circuits from receiving too much current. In order to distribute the power from the batteries two separate systems were designed and implemented. One to power the analog front end of the circuit and the other to power the microprocessor. The two different distributions were needed since both systems require a different voltage in order to function properly.

To understand how the power is regulated a few basic circuit concepts need to be first reviewed. First off the converter and regulator both operate around the functionality provided by an inductor. The structure of an inductor typically consists of an insulated copper wire that is wrapped around a ferromagnetic material that amplifies the magnetic field produced by the coil of wire. When a current is applied, the coil builds up a magnetic field that is attempting to oppose the current. This happens until the voltage source is interrupted which causes the inductor to counteract the sudden change in current by dissipating its stored energy into the circuit.

Another important component of the converter and regulator is a switch control signal which dictates the amount of voltage that is output by the circuit. This process is known as

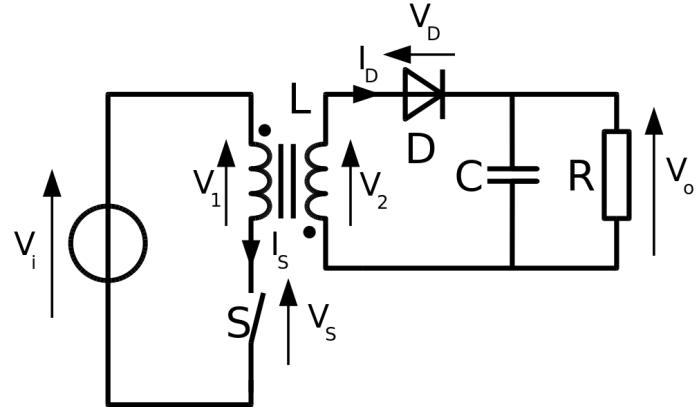
pulse width modulation. The block diagram as well as the resulting waveforms can be seen in Figure 12.

Figure 12: Pulse Width Modulation Methodology [2]



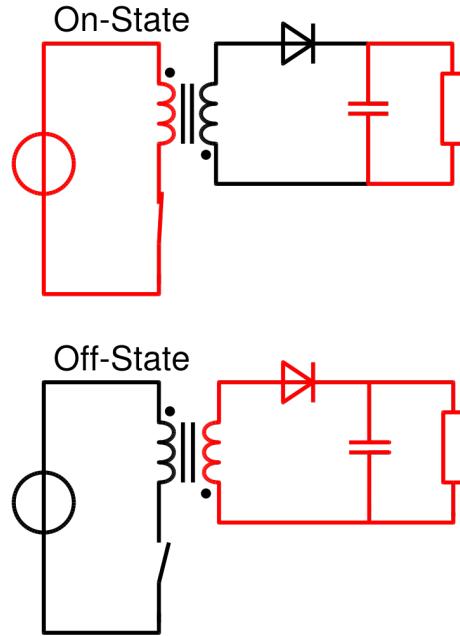
The actual voltage and desired voltage are compared and there is a control outputted into a comparator. This comparator compares the voltage with a periodic signal such as a sawtooth signal and the output signal is what controls the switches which are usually MOSFETS. When the voltage of the sawtooth wave is below the voltage control signal, the switch control is on which results in the switch to turn on. When the voltage of the sawtooth rises above the control voltage the switch turns off. An important takeaway from this which is used to help determine the relationship between the input and output voltages of the converter is the duty ratio. The duty ratio is expressed as  $D$  and it equals the time that the switch control signal is on divided by the period of the signal.

Figure 13: Flyback Converter Schematic [3]



In order to provide power to the analog front end an Isolated DC/DC converter was used. A schematic of the inside of a typical converter can be seen in Figure 13. The converter is isolated due to the transformer. The isolation comes from the fact the output voltage of the transformer is no longer referenced to the ground of the input voltage, meaning that when one uses the heartbeat sensors they will not get electrocuted from them. It is also a safety net in case there is a current surge from another part of the circuit such as motor. Since inductors cannot change their current instantaneously, the analog front end will not feel any sort of current surge if it were to happen. It also prevents the front end from any interference from noisy ground currents which is critical to ensure an accurate reading of the users heartbeat.

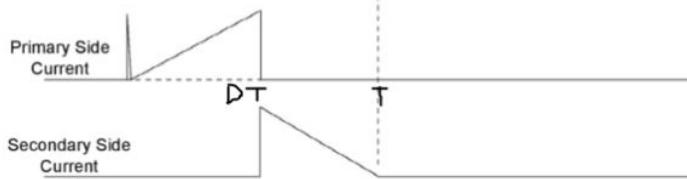
Figure 14: Flyback Converter Configurations [3]



In order to step down the voltage a switching signal is used to open and close a switch.

The frequency of the switch opening and closed determines the voltage that will be received by the output. As can be seen in Figure 14, during the on state of the switch, the primary side of the transformer on the input side of the converter stores energy from its DC input. The secondary winding has an opposite polarity, the voltage induced on the secondary winding is negative which cannot flow through the output side due to the diode blocking a negative flow of current. The voltage across the load is instead supplied by the capacitor that has stored energy from when the switch is open. When the switch is off, the polarity of the transformer switches due to the collapsing magnetic field. This allows for the secondary side to have a positive current to pass through the diode which results in the charging of the capacitor as well as a voltage across the output. In order to obtain the equations that govern the operation of the converter one must look at the resulting waveforms. Shown in figure three are the primary and secondary side current waveforms when at the start of the waveform the switch is on, it turns off at  $DT$  and then back on at  $T$ . Since the transformer sides can be treated as separate inductors and with continuous operation the average current will be constant and the following relation can be obtained. The slopes are represented  $m_1$  and  $m_2$  which are multiplied by the amount of time they last.  $D$  represents the duty ratio of the switch.

Figure 15: Flyback Converter Timing Diagram [3]



$$\Delta i_L = 0 = m_1 DT + m_2(T - DT) \quad (1)$$

$$= V_i Z D T + -V_o Z N_1 N_2 (1 - D) T \quad (2)$$

$$= V_i D - V_o N_1 N_2 (1 - D) \quad (3)$$

$$V_i D = V_o N_1 N_2 (1 - D) \quad (4)$$

$$V_o = D V_i N_2 N_1 \quad (5)$$

To provide power to the microprocessor a step-down switching regulator was used. This operates in a similar fashion as to the previously described converter. A typical circuit of a step down regulator can be seen in Figure 16. This circuit works by using a switch controlled by pulse width modulation. When the switch is on it allows for the current to pass through charging the inductor as well as providing voltage to the load and capacitor. When the switch is turned off the inductor releases its stored energy which flows through the load and back through the diode. To understand the relationship between the input and output voltage it is essential to examine the inductor current waveform which is seen in Figure 17 Since the average current remains constant the following equation can be derived. The slopes are represented  $m_1$  and  $m_2$  which are multiplied by the amount of time they last.  $D$  represents the duty ratio of the switch.

$$\Delta i_L = 0 = m_1 DT + m_2(T - DT) \quad (6)$$

$$= V_i - V_o Z D T + -V_o Z (1 - D) T \quad (7)$$

$$= (V_i - V_o) D - V_o (1 - D) \quad (8)$$

$$= V_i D - V_o D + V_o D - V_o \quad (9)$$

$$= V_i D - V_o \quad (10)$$

$$V_o = D V_i \quad (11)$$

Figure 16: Step Down Regulator Circuit [4]

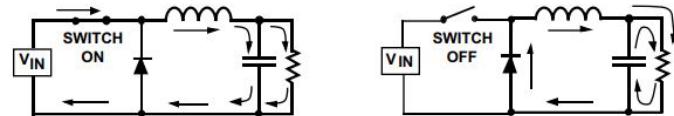
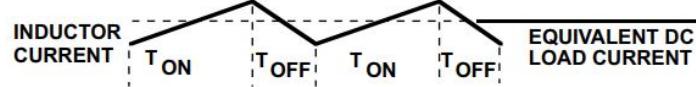
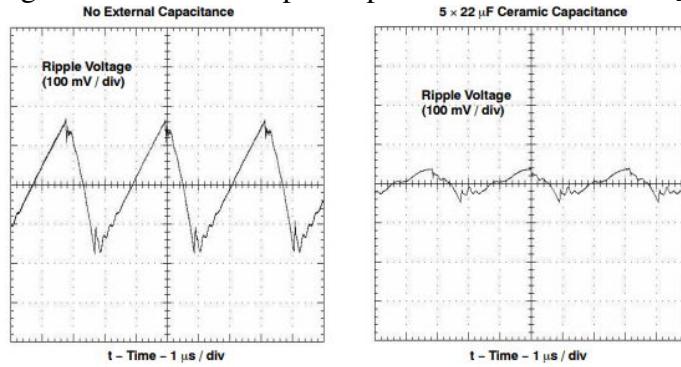


Figure 17: Inductor Current Waveform [4]



Both the converter and regulator use ceramic capacitors at the input and output to improve its performance. The purpose of the capacitor at the input is to reduce the ripple voltage seen at the input of the regulator/converter that results from the chopping of the voltage inside the circuit. A ceramic capacitor must be used due to the low equivalent series resistance and inductance as well as their wide operating temperature range. The effect of the input capacitor can be seen in Figure 18.

Figure 18: Effect of Input Capacitor on Rectification [4]



### 4.3.1 Motor Controller

Figure 19: Motor Controller Schematic

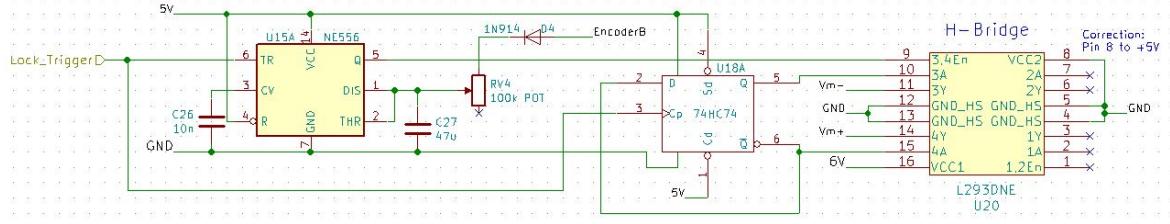
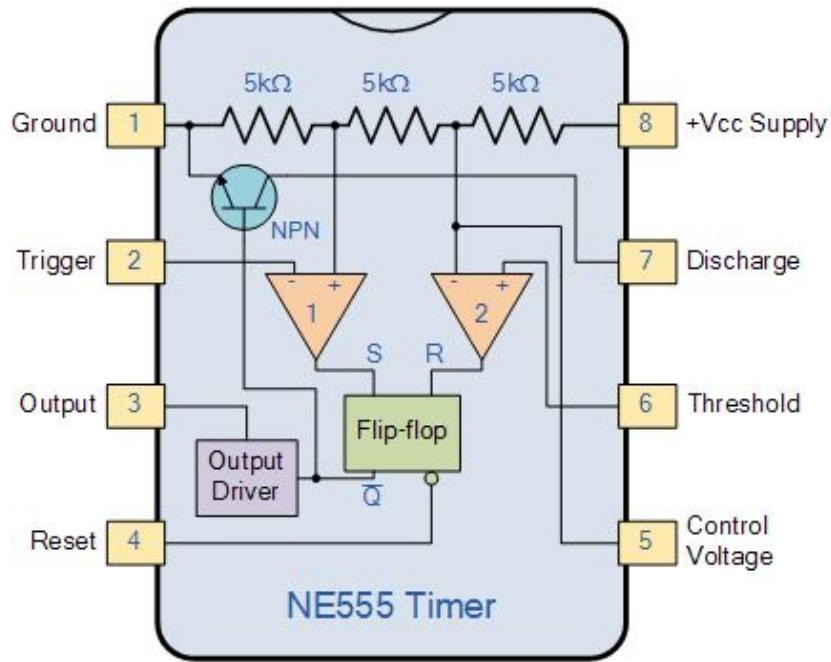


Figure 20: Internal Design of a 555 Timer [5]

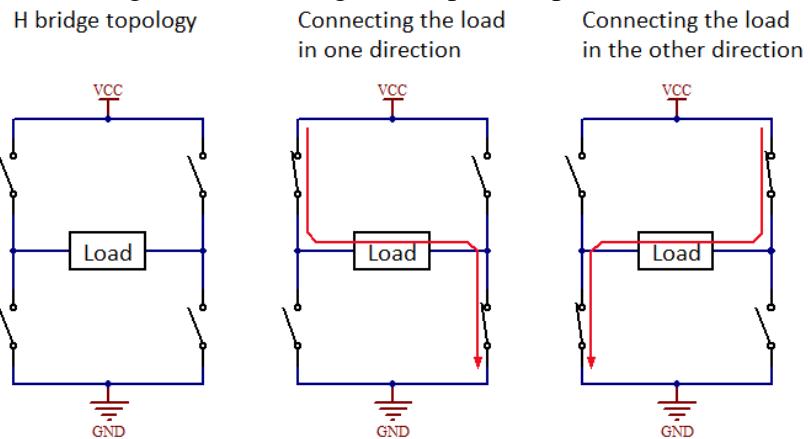


The full schematic of the motor controller can be seen in Figure 19. It makes use of a 555 timer, D flip-flop and an H-bridge. First off the 555 timer is operating in monostable operation. A block diagram that will be referenced to explain this operation can be seen in Figure 20. When a negative pulse coming from the microprocessor is sent through the trigger pin, comparator 1 will detect it and set the state of the flip flop which will in turn change the output from a low state to a high state which provides a DC voltage output. This will also turn off NPN transistor that is normally providing a route to the ground. Turning off the transistor removes the short circuit across the external capacitor that is connected to the discharge pin (pin 1 in Figure 19). As a result, the capacitor will charge providing an until it reaches two thirds of the supply voltage. Due to the voltage division inside the 555 timer, the second comparator will detect the higher voltage coming from the capacitor and send an

output to the flip-flop setting it back to the original low value. The length of time that the 555 timer produces an output is determined by the resistor and capacitor values that dictate how long the capacitor takes to reach the threshold voltage to turn the 555 timer off. For this system a potentiometer was used in order to allow for easy tuning during prototype trials as well as an encoder output from the motor. This encoder outputs voltage based on the position of the motor. This voltage charges the discharge capacitor meaning that when the 555 timer is triggered the output pulse takes into account the motor position.

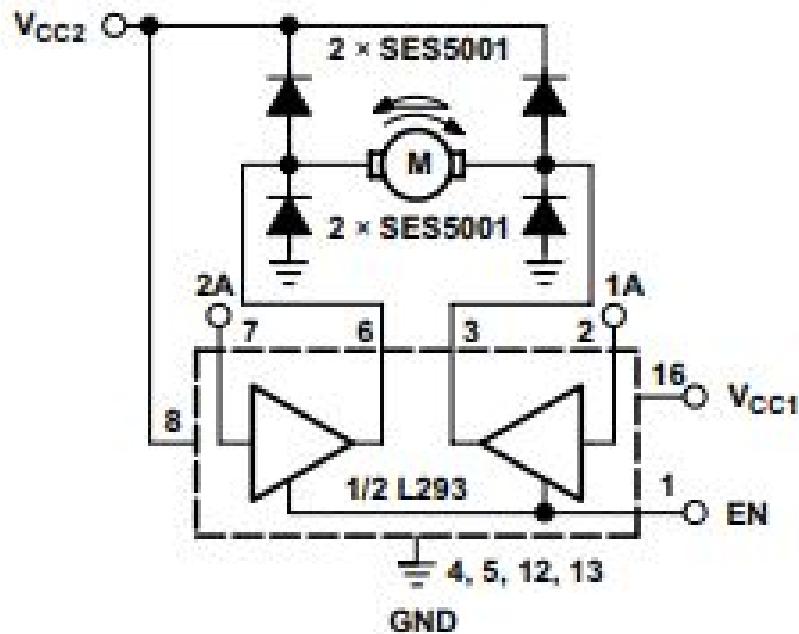
The 555 timer output is sent to an H-bridge enable pin which turns on the switches that will be used to control the motor. A Basic H-bridge topology can be seen in Figure 21. The idea of an H-bridge is to close two switches simultaneously in such a way that allows for the current to flow from the voltage source, through the motor and into the ground. The switches are opened based on which direction the motor needs to rotate in. If current flows through the positive terminal of the motor first, it will rotate in one direction and inversely, current flowing through the negative terminal will cause the motor to spin in the opposite direction.

Figure 21: H-Bridge Principle of Operation [6]



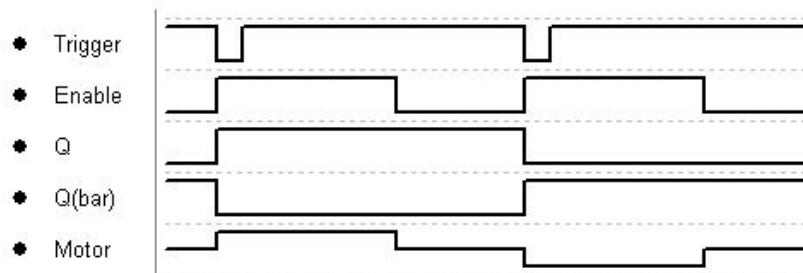
The LM93 H-bridge that has been chosen to drive the motor consists of logic circuits acting as the switches which can be seen in Figure 22. Once the switches are enabled from the 555 timer the switch inputs will read the voltages coming from a D-flip flop. A switch will either get a HIGH or LOW value. If a switch gets a high value it will provide the motor with voltage from the source. If a switch obtains a low value it will act as a ground for the current from the source to flow through. Using a Flip-flop one of the switches will have a HIGH input and the other will be a LOW input. This will result in current flowing through motor causing it to turn. Once the motor turns it will actuate the lock causing it to open and then close after a brief delay.

Figure 22: H-Bridge Circuit [7]



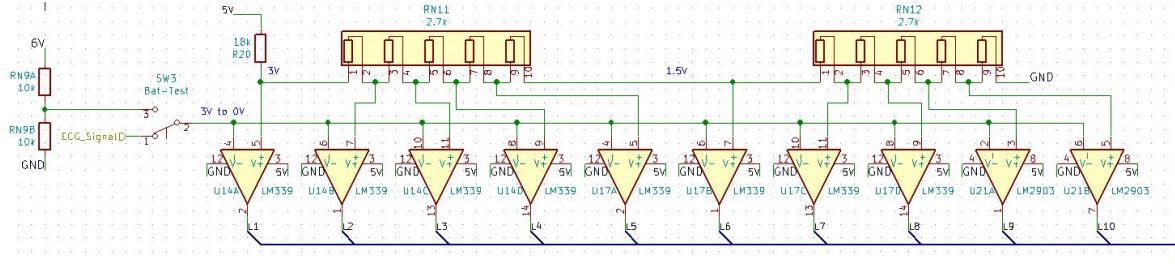
In this circuit, when the microprocessor determines a positive match, it will output two negative pulses. These pulses are sent to the 555 timer as well as the D flip-flop. The 555 timer takes these signals as trigger signals in order to provide and output to the enable line of the H-Bridge. The Flip-flop takes the trigger pulses as the clocking signals that flip the Q and output signals. This ensures that the motor will spin opposite to the direction it has spun on the previous trigger pulse. A timing diagram is shown in Figure 23. For the motor signal the time in which it is not spinning is represented by the signal sitting between high and low, while high would represent it opening the lock and low represents closing the lock.

Figure 23: Motor Controller Timing Diagram



### 4.3.2 Voltage Indicator

Figure 24: Voltage Indicator Circuit Schematic

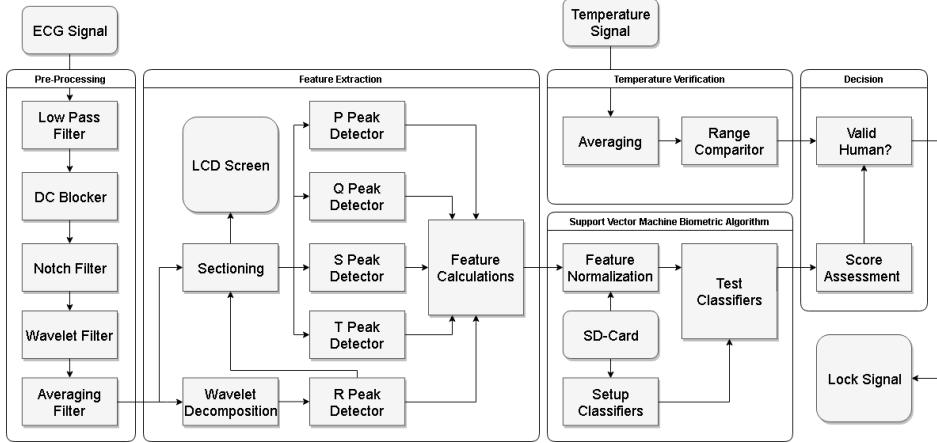


Since a finite source of power is being used a voltage indicator was designed in order to give the user indication as to how much charge the battery has. This was achieved by using voltage division and comparators which can be seen in Figure 24. A reference voltage from the 5V converter is fed through two strings of resistors after using a separate voltage division to reduce the voltage to 3 volts. The resistors are set up as such to give a voltage drop of .3V as you go along the string. The voltages are then directed to separate comparators which compare it to the voltage from the battery. When a comparator detects a higher voltage from the battery it will sink current at the output lighting up the LED that is fed to it. This comparing system has a dual purpose as there is a button that acts as a switch between the battery and the ECG signal. Not only does this give a second indication that the signal is being read but it also prevents the battery from being used up by the string of LEDs.

## 5 Biometric Algorithm

At the very core of the security system a custom algorithm which utilizes machine learning to facilitate the differentiation between personal electrocardiogram waveforms. An overview of the inner workings of this algorithm is provided in Figure 25 and briefly expanded on in the sections that follow.

Figure 25: Biometric Algorithm Architecture



## 5.1 Prepossessing

After careful examination of regular ECG waveforms in both the time and frequency domain, a filter network capable of isolating the relevant parts of the electrocardiogram signal was developed. The network consisted of five fundamental filter designs each with varying filter orders. Utilizing matlab these filter designs were combined in the formation of a grand transfer function. Said function consisted of over 60 coefficients and imposed a extremely large intermediate memory requirement. As a result the grand filter was instead split into 27 biquadratic sections which although slower, reused memory addresses and thus significantly reduced program overhead. Since the presentation of the individual filter function coefficients would prove lengthy and provide minimal insight, the code used for their generation is provided instead.

```

function [CleanECG PPFILTER] = Preprocessing(RawECG, sampleRate)
    % Low Pass Filter (F<100Hz)
    f1_f = [100 125];
    f1_M = [1 0];
    f1_A = [0.01 0.01];
    [f1_N,f1_wc,f1_B,f1_type] = kaiserord(f1_f,f1_M,f1_A,sampleRate);
    f1_b = fir1(f1_N,f1_wc,f1_type,kaiser(f1_N+1,f1_B), 'noscale');
    % DC Blocker (adjust with f2_wc)
    f2_wc = 0.995;
    f2_a =[1 -f2_wc];
    f2_b =[1 -1];
    % Notch Filter (remove power noise 60Hz, applied twice)
    [f3_b,f3_a] = pei_tseng_notch(2*60/sampleRate,4/sampleRate);
    % Moving Average Filter (smoothing)
    f4_avg = 8;
    f4_b = ones(f4_avg,1)/f4_avg;
    % Wavelet Filter Denoising (WIP - needs tuneing)
    f5_level = 3;
    f5_name = 'db1';
    %Apply the Filters
    PPFILTERB = conv(conv(conv(conv(f1_b,f2_b),f3_b),f3_b),f4_b)
    
```

```

PPFilterA = conv(conv(f2_a,f3_a),f3_a)
f4Out = filter(PPFilterB,PPFilterA,RawECG);

[f5_c,info]=ufwt(f4Out,f5_name,f5_level);
f5Out = iufwt(f5_c(:,1:2),'db1',1)';
%Must Correct for Delay %f1Out = [f1Out(floor(f1_N/2):length(f1Out))
    %zeros(1,(f1_N/2)-1)]; %Remove delay so we can compare
CleanECG = f5Out;

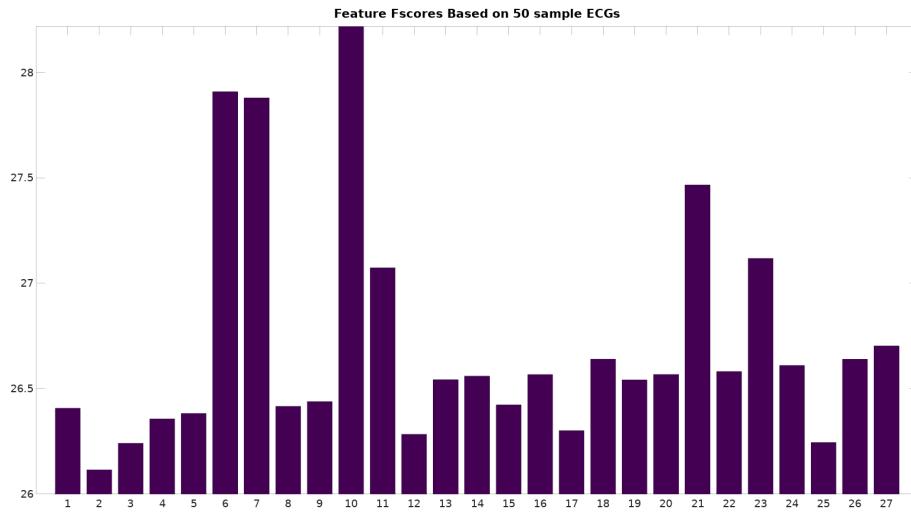
PPFilter = tf2sos(PPFilterB,PPFilterA)
end

```

## 5.2 Feature Identification

After filtering the next stage of the process is to identify the various key features which can be utilized for the unique identification of humans. In order to accomplish this task a large set of measurements were performed on a database of ECG waveforms. These measurements included quantification of inter-peak timing, amplitude, as well as tangents. A study of the variability of these measurements between individuals was conducted using the F-Scores methodology. This allowed for the determination of the measurements which likely possessed the most inter-human differences which would thus be most beneficial for use as features in a machine learning algorithm. Out of all the potential targets the top 27 were selected due to their differentiability and ease of measurement, the relative F-Scores of these selected features is illustrated in Figure 26.

Figure 26: FScores for the 27 Most Prominent Features



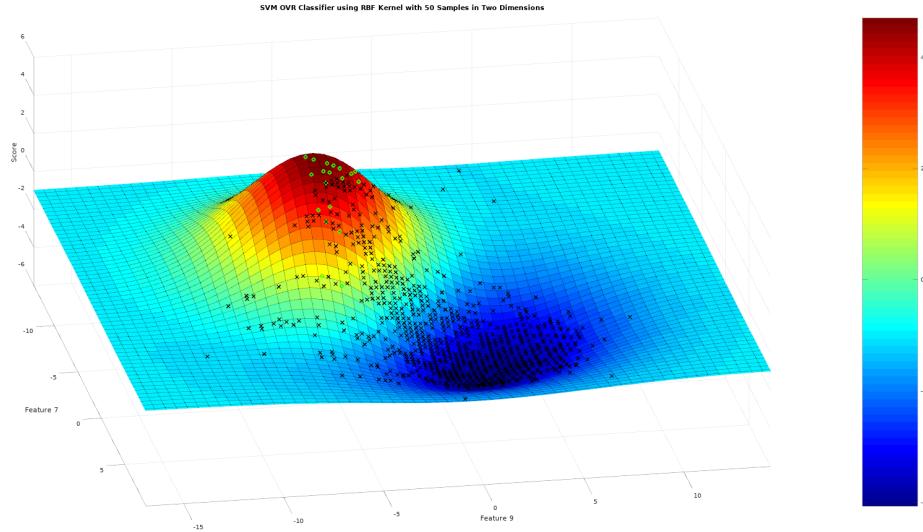
A few frequency domain features were also tested but these were found to provide minimal benefits to classifier accuracy and were computationally expensive to obtain, thus they were excluded from the proposed algorithm. It is noted that the most prominent feature identified by this study was the tangent line formed between the Q and R peaks. It is not fully known

why this particular property has such a strong differentiability between individuals but given the nature of the test it is likely as a result of extreme consistency in its formation.

### 5.3 Classifier Creation

In order for analysis to take place a support vector machine system was employed whereby the features could be used to linearly separate positive and negative results using a one vs rest approach. The radial basis function was selected as the classifier kernel due to its ability to better account for isolated regions of data in multiple dimensions. The generation and subsequent evaluation of the support vector machine classifier was executed using the octave code illustrated in Appendices D and E respectively. A sample of some of the custom octave subfunctions used in this process is also provided in Appendix F for reference. As the machine learning classifier exists in 27 dimensions, it becomes impossible to visualize the process and thus directly evaluate measurement accuracy. In an attempt to gain insight into the inner workings of the produced biometric classifier a 2D examination of the relative distinguishability between two random features across 50 individuals was conducted. The results of this study are presented in Figure 27 which show a higher score was produced in the region containing the positive samples (green) and a negative score for the region with the negative samples. The high localization of positive samples is thus shown to allow for high particularity in measurement which is a good indication that the features under examination do indeed possess uniquely identifying characteristics.

Figure 27: Support Vector Machine Manual 2D Classifier Analysis



### 5.4 Simulated Results

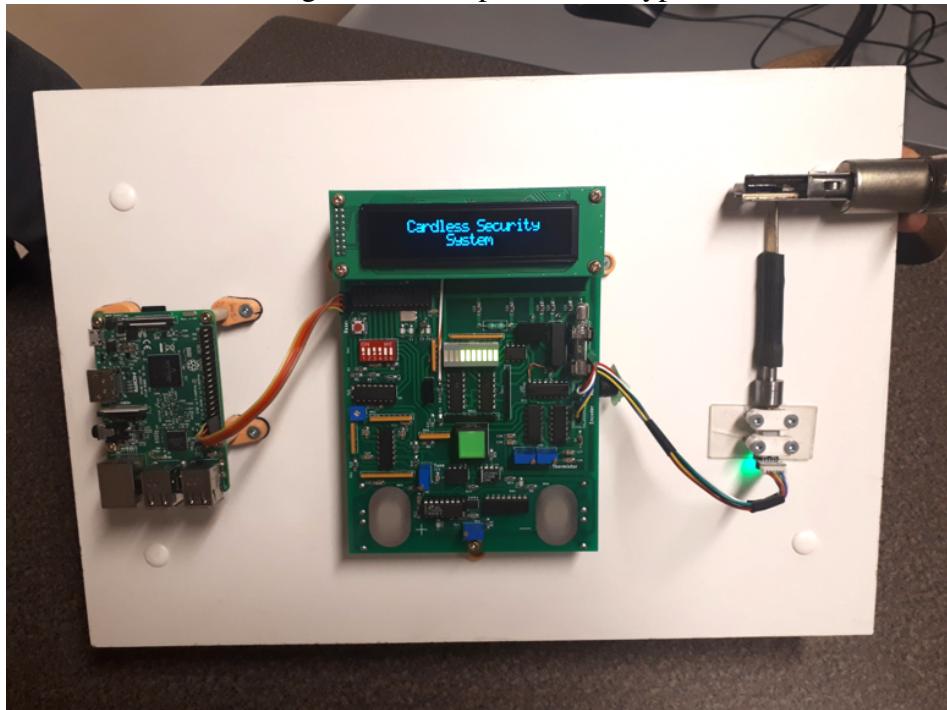
In order to gather insight into the feasibility of using a support vector machine to correctly distinguish between individuals an evaluation of the algorithm needed to be performed. Using the proposed algorithm a 94% positive identification rate with no false positives was obtained

utilizing the ECG database provided by Tatiana Lugovaya[8]. The sample size consisted of 50 different individuals with repeated one to three separate measurement samples per person. Due to the high success rate obtained in this simulation the feasibility of the project was again verified and thus development of the prototype unit approved.

## 6 Prototype Fabrication

After the completion of the design the project moved into its implementation stage, the final product of which is illustrated in Figure 28 for reference. The following sections outline some of the more notable aspects of the design fabrication process.

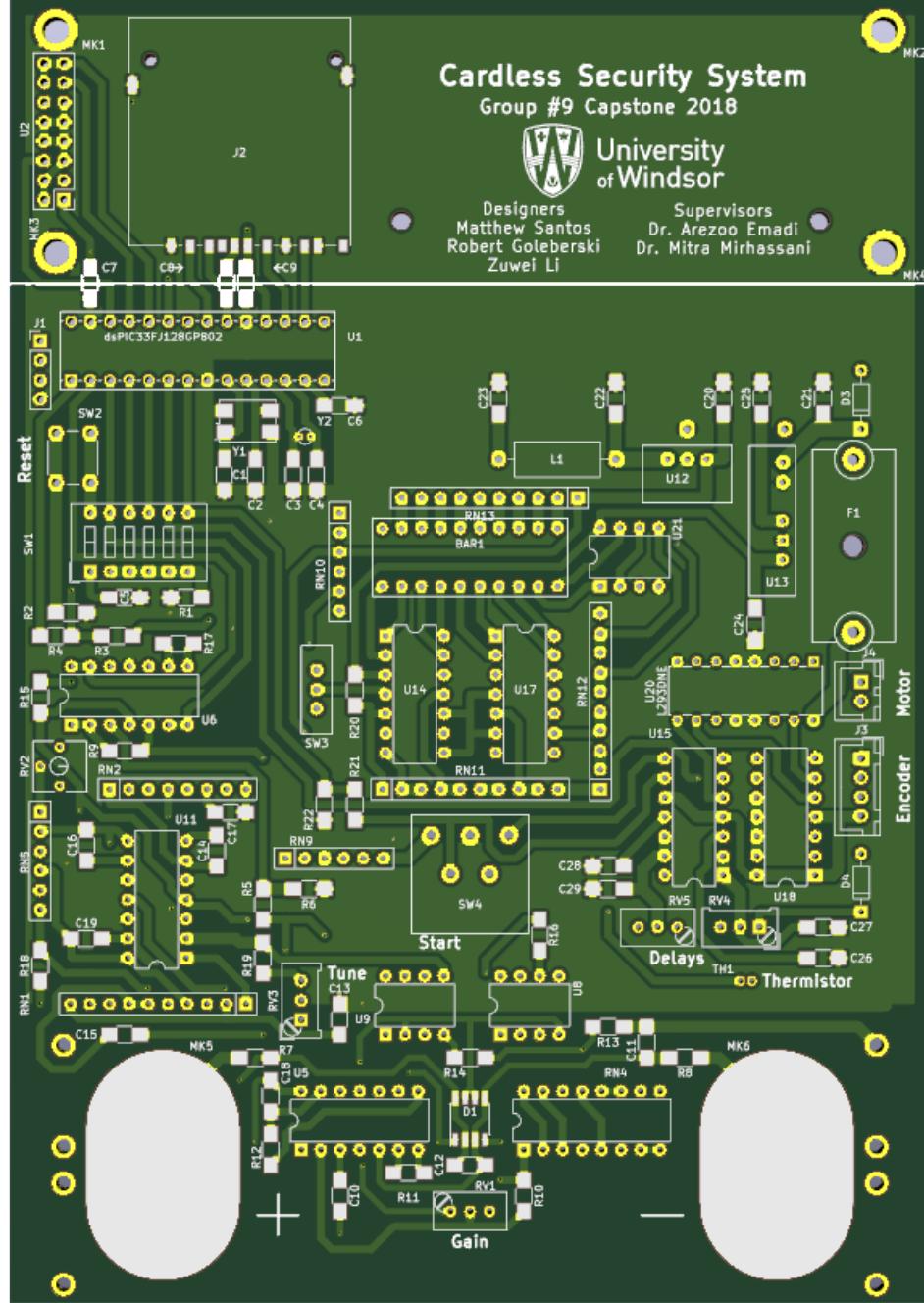
Figure 28: Completed Prototype



### 6.1 PCB Layout Design

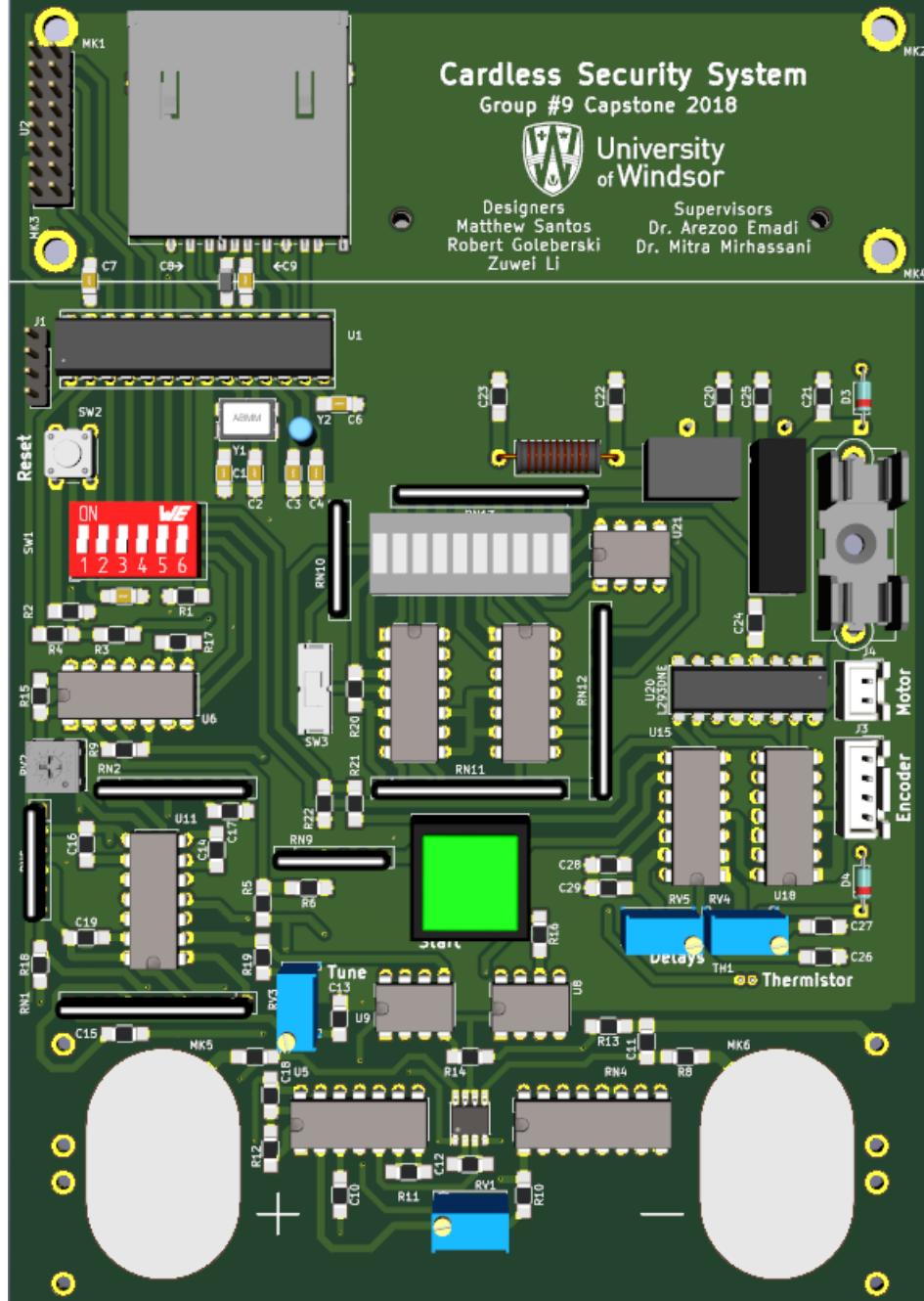
Since an embedded solution was desired the project was always intended to be produced on a printed circuit board (PCB) in order to achieve this a custom PCB layout was produced and is illustrated in Figure 29 with a 2D Render.

Figure 29: Completed PCB Design Layout



Since the project would be skipping the bread-boarding development stage the proposed layout had to be completely free from imperfections. In order to assist in verifying conformity the 3D models of the various components were collected and imported into the layout design software to check for proper size and alignment. A visual illustration of the PCB design with part models superimposed is seen in Figure 30.

Figure 30: PCB Layout with 3D Part Models Included



The PCB design was then forwarded to a third party manufacturer for production. A stencil was also ordered to assist in the assembly process.

## 6.2 Reflow Soldering

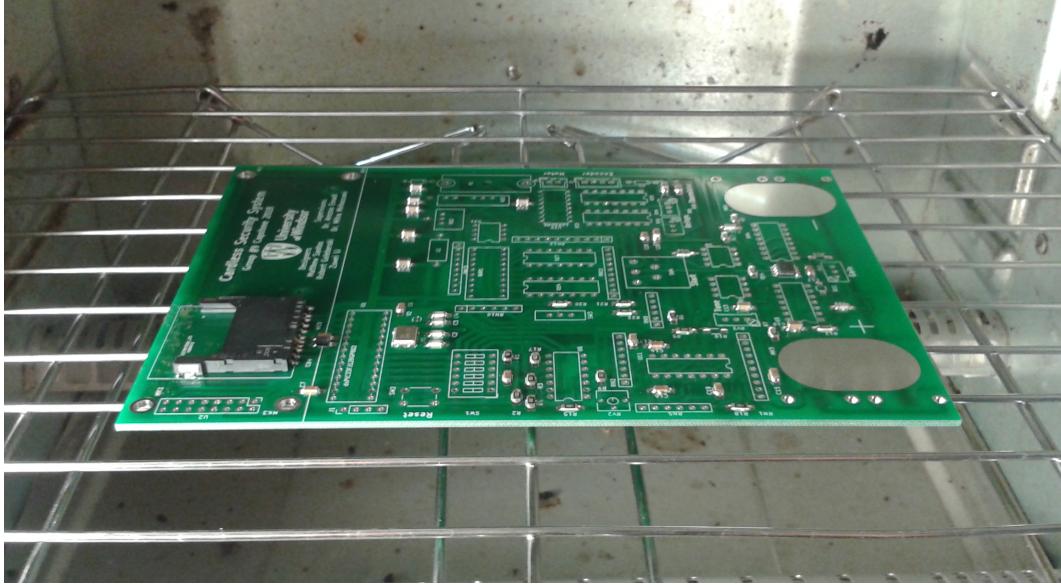
After procuring the printed circuit board, components, and stencil the assembly could officially begin. While the majority of the process consisted of standard hand soldering, the

surface mount components were attached through reflow soldering. The process consisted of utilizing the ordered stencil to selectively apply solder paste to target pads on the PCB as seen in Figure 31. Following this stage the components were placed on top of their pads and the board moved into a small household toaster oven for heating. Over the course of a few minutes the paste melted and secured the components to the board with minimal effort required by the operator. The moment at which reflow occurred was recorded and can be seen in Figure 32.

Figure 31: Using the Stencil to Apply Solder Paste in Preparation for Reflow



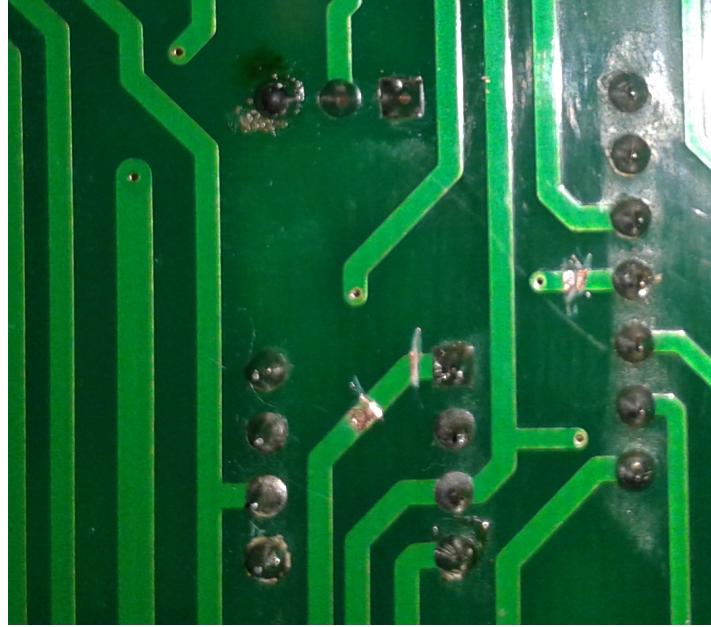
Figure 32: Reflowing using Household Toaster Oven



### 6.3 Improvised Modifications

While implementing this device the team was faced with many challenging problems that required adaptive problem solving. One of these problems is deemed noteworthy enough for mention in this report. This issue was encountered during initial breadboard testing of the LCD screen unit. The device abnormally attempted to draw excessive amounts of current. It was later discovered that the datasheet provided by the distributor was wildly inaccurate. After contacting the manufacturing directly it was found that everything from pinouts to protocols were vastly different and major modifications to the pcb layout were required. An example of one of the layout modifications can be seen in Figure 33 whereby a trace was cut and exposed so that a jumper wire could be attached for re-routing. Additionally alongside the protocol changes were changes to operating voltage levels. This was again not accounted for in the design plan and so logic shifters had to be added underneath the board to allow for LCD operation.

Figure 33: Example of PCB Trace Modifications



As a result of this improper datasheet the amount of work required to get the LCD screen operating correctly was vastly increased but thankfully through the combined efforts of the project team a new screen was not required and the existing part was successfully integrated into the design.

## 6.4 Implementation Cost

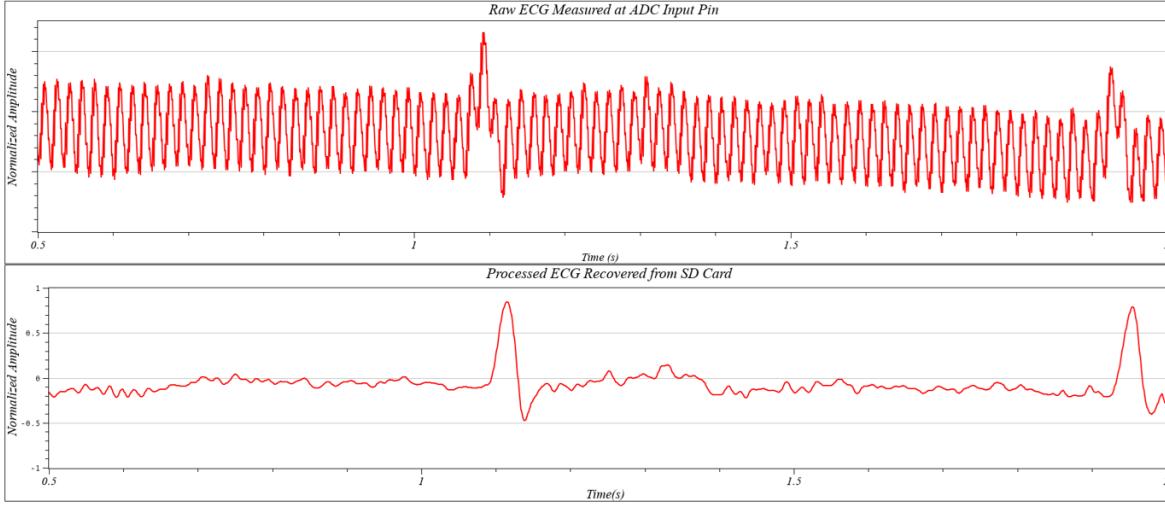
The complete cost of components for this implementation can be found in Appendix B showing a grand total of \$220.50 . The costs for PCB fabrication and assembly were absorbed by the team members but were estimated to total \$50. As a result the project was completed under budget.

## 7 Prototype Evaluation

Following fabrication the unit's various features were evaluated to ensure conformance to expectations. It was found that the prototype performed its job exactly as expected with only minor deviations. There was however unexpected high levels of noise measured through the sensors. Although initial measurement for the amount of electromagnetic interference on human fingers showed 100mV using an oscilloscope, the prototype unit was instead experiencing levels three times as high which caused pre-amplifier noise saturation. Thankfully the unit had been designed with potentiometers for gain adjustment and so the reduction in the gain eliminated all occurrences of saturation. This also however reduced the relative amplitude of the desired ECG waveform. Fundamentally speaking the signal to noise ratio using the physical implementation was significantly lower and as a result the resolution of the recovered waveform suffered. The result of measurements taken from the prototype under

ideal conditions are illustrated in Figure 34 and show reasonable QRS complex accuracy but the P and T waves are barely resolved.

Figure 34: Recovered ECG Signal under Ideal Conditions



This had a dramatic effect on accuracy since the biometric algorithm called for approximately one third of the SVM features to be generated through the detection of the P and T waves. Thus without reliable detection of these peaks the ability for the prototype to successfully distinguish between individuals would be significantly compromised. Unfortunately due to time constraints and limited assistance, a proper quantification of the prototype's biometric identification properties was not performed. There was limited testing under ideal conditions across a sample size of four which showed a positive identification rate of around 70% however given the number of samples utilized it would be hubris to associate any level of reliability on this value. In order to properly evaluate the unit a minimum of 10 unique individuals is recommended as the support vector machine is highly influenced by sample size.

## 8 Recommendations

In light of the results obtained from this project report, the following recommendations are provided for those wishing to continue the development of the produced ECG based security system.

1. Further improve on the resolution of the recovered ECG waveform
2. Get ethics board approval to gather ECG samples to increase classifier accuracy
3. Setup the prototype unit somewhere on campus for a real world evaluation

## **9 Conclusions**

In conclusion the project demonstrates the potential of electrocardiogram security systems but also highlights some of the difficulties associated with their success. While the biometric properties of the ECG waveform have been successfully demonstrated both in this report and other studies, there remains a great deal of importance placed on obtaining high quality electrocardiogram signals in order for a practical system to succeed.

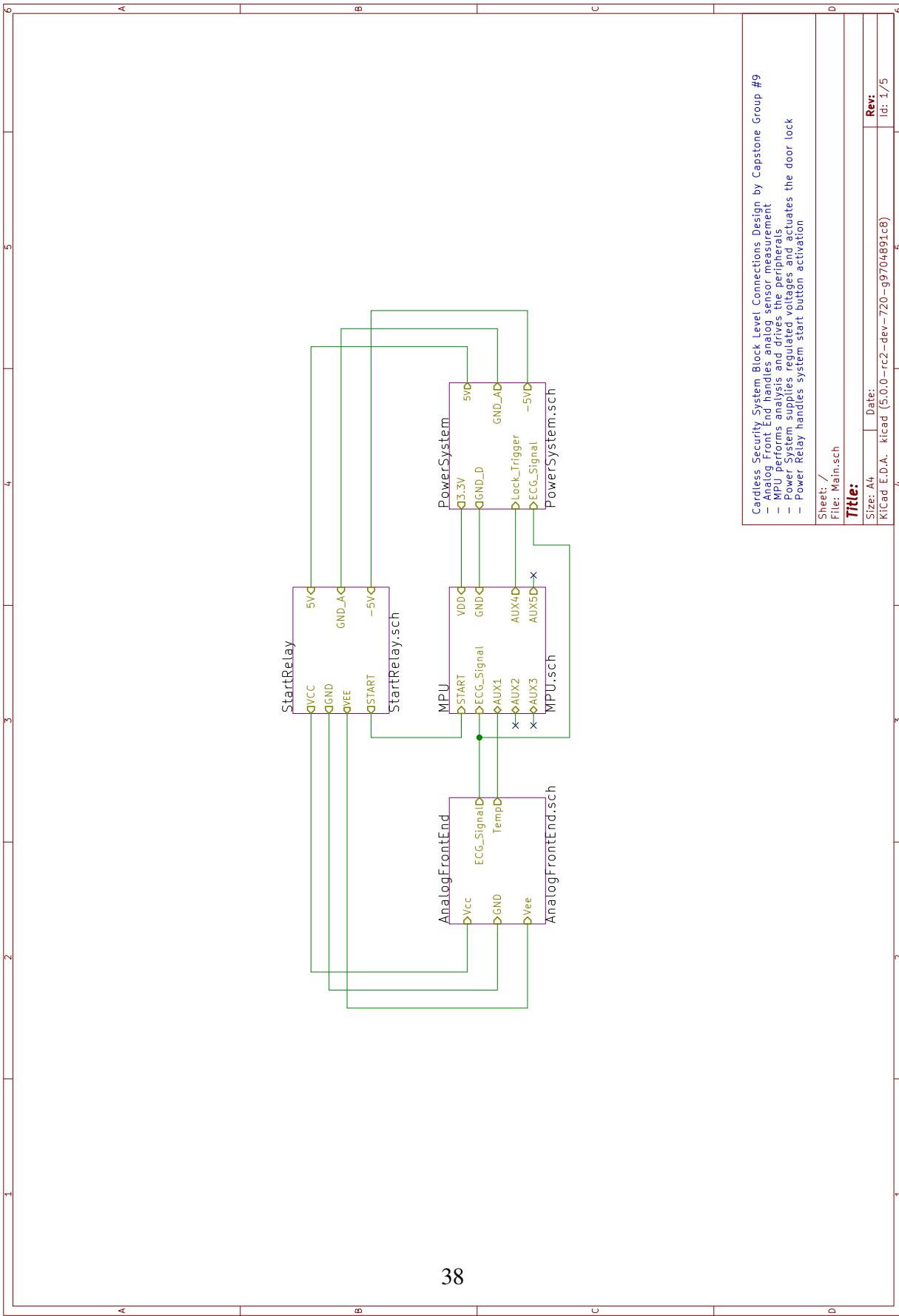
## References

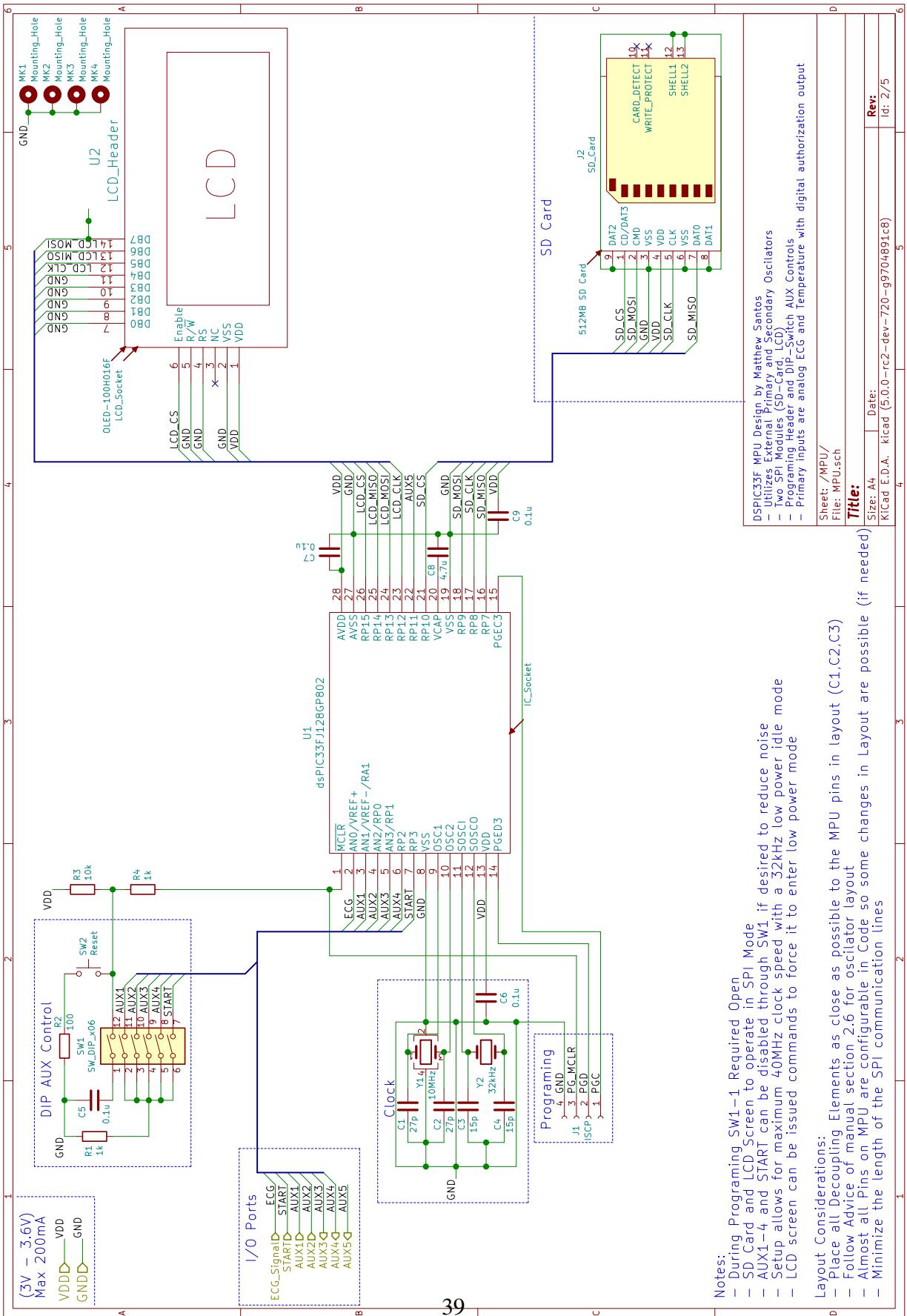
- [1] P. T. Rao, S. K. Rao, G. Manikanta, and S. R. Kumar, “Distinguishing normal and abnormal ecg signal,” *Indian Journal of Science and Technology*, vol. 9, no. 10, 2016.
- [2] N. Kar, “Switch-mode dc-dc converter slides,” *University of Windsor*, 2018.
- [3] “Flyback converter.” [Online]. Available: [https://en.wikipedia.org/wiki/Flyback\\_converter](https://en.wikipedia.org/wiki/Flyback_converter)
- [4] “Switching regulator fundamentals,” *Switching Regulator Fundamentals*, p. 127, 2016.
- [5] “555 timer tutorial - the monostable multivibrator,” *Basic Electronics Tutorials*, Apr 2018. [Online]. Available: [https://www.electronics-tutorials.ws/waveforms/555\\_timer.html](https://www.electronics-tutorials.ws/waveforms/555_timer.html)
- [6] “Design of an h bridge,” *Axotron*. [Online]. Available: [http://axotron.se/index\\_en.php?page=34](http://axotron.se/index_en.php?page=34)
- [7] *L293x Quadruple Half-H Drivers*, Texas Instruments, 2016.
- [8] T. Lugovaya, “Biometric human identification based on ecg,” *Biometric Human Identification based on ECG*, Oct 2012. [Online]. Available: <https://www.physionet.org/pn3/ecgidb/images/>
- [9] S. Venugopalan, M. Savvides, M. O. Grofa, and K. Cohen, “Analysis of low-dimensional radio-frequency impedance-based cardio-synchronous waveforms for biometric authentication,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 8, p. 23242335, Aug 2014.
- [10] P. Zicari, A. Amira, G. Fischer, and J. McLaughlin, “An embedded system for on field testing of human identification using ecg biometric,” *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, 2012.
- [11] M. Sansone, A. Fratini, M. Cesarelli, P. Bifulco, A. Pepino, M. Romano, F. Gargiulo, and C. Sansone, “Influence of qt correction on temporal and amplitude features for human identification via ecg,” *2013 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications*, 2013.
- [12] V. N and S. Jayaraman, “Human electrocardiogram for biometrics using dtw and flda,” *2010 20th International Conference on Pattern Recognition*, 2010.
- [13] J. Pinto, J. Cardoso, A. Loureno, and C. Carreiras, “Towards a continuous biometric system based on ecg signals acquired on the steering wheel,” *Sensors*, vol. 17, no. 10, p. 2228, 2017.
- [14] R. Kountchev and K. Nakamatsu, *New approaches in intelligent image analysis techniques, methodologies and applications*. Springer, 2016.

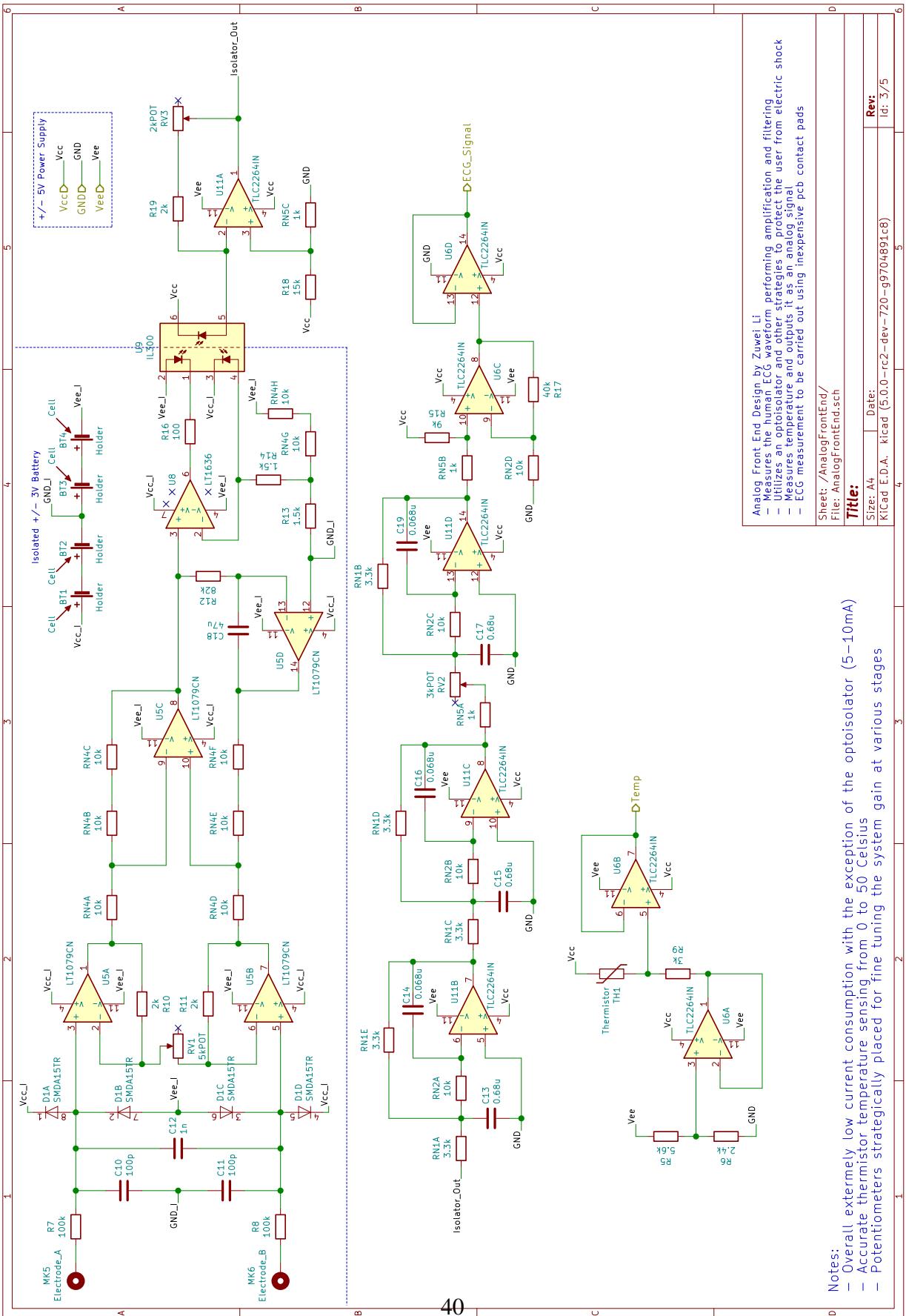
- [15] I. Assadi, A. Charef, N. Belgacem, A. Nait-Ali, and T. Bensouici, “Qrs complex based human identification,” *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2015.
- [16] Q. Zhang, D. Zhou, and X. Zeng, “Heartid: A multiresolution convolutional neural network for ecg-based biometric human identification in smart health applications,” *IEEE Access*, vol. 5, p. 1180511816, 2017.
- [17] Y. Chen and W. Chen, “Finger ecg-based authentication for healthcare data security using artificial neural network,” *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2017.
- [18] H. P. D. Silva, A. Fred, A. Loureco, and A. K. Jain, “Finger ecg signal for user authentication: Usability and performance,” *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013.
- [19] Y. N. Singh and S. K. Singh, “Evaluation of electrocardiogram for biometric authentication,” *Journal of Information Security*, vol. 03, no. 01, p. 3948, 2012.
- [20] H. Hussain, C.-M. Ting, F. Numan, M. N. Ibrahim, N. F. Izan, M. M. Mohammad, and H. Sh-Hussain, “Analysis of ecg biosignal recognition for client identification,” *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2017.
- [21] M. Jahiruzzaman and A. B. M. A. Hossain, “Ecg based biometric human identification using chaotic encryption,” *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2015.
- [22] S. Eberz, N. Paoletti, M. Roeschlin, A. Patani, M. Kwiatkowska, and I. Martinovic, “Broken hearted: How to attack ecg biometrics,” *Proceedings 2017 Network and Distributed System Security Symposium*, 2017.
- [23] P. Selvey, “Ecg filters,” *ECG Filters — MEDTEQ*. [Online]. Available: <http://www.medteq.info/med/ECGFilters>
- [24] A. Loureco, H. P. da Silva, D. P. Santos, and A. L. N. Fred, “Towards a finger based ecg biometric system,” *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*, Jan 2011.
- [25] S. Mayhew, “History of biometrics,” Jul 2018. [Online]. Available: <https://www.biometricupdate.com/201802/history-of-biometrics-2>
- [26] “Ecg timeline - history of the electrocardiogram.” [Online]. Available: <https://ecglibrary.com/ecghist.html>
- [27] “Ecg biometrics: the heart of data-driven disruption?” Jan 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0969476518300110#bib1>

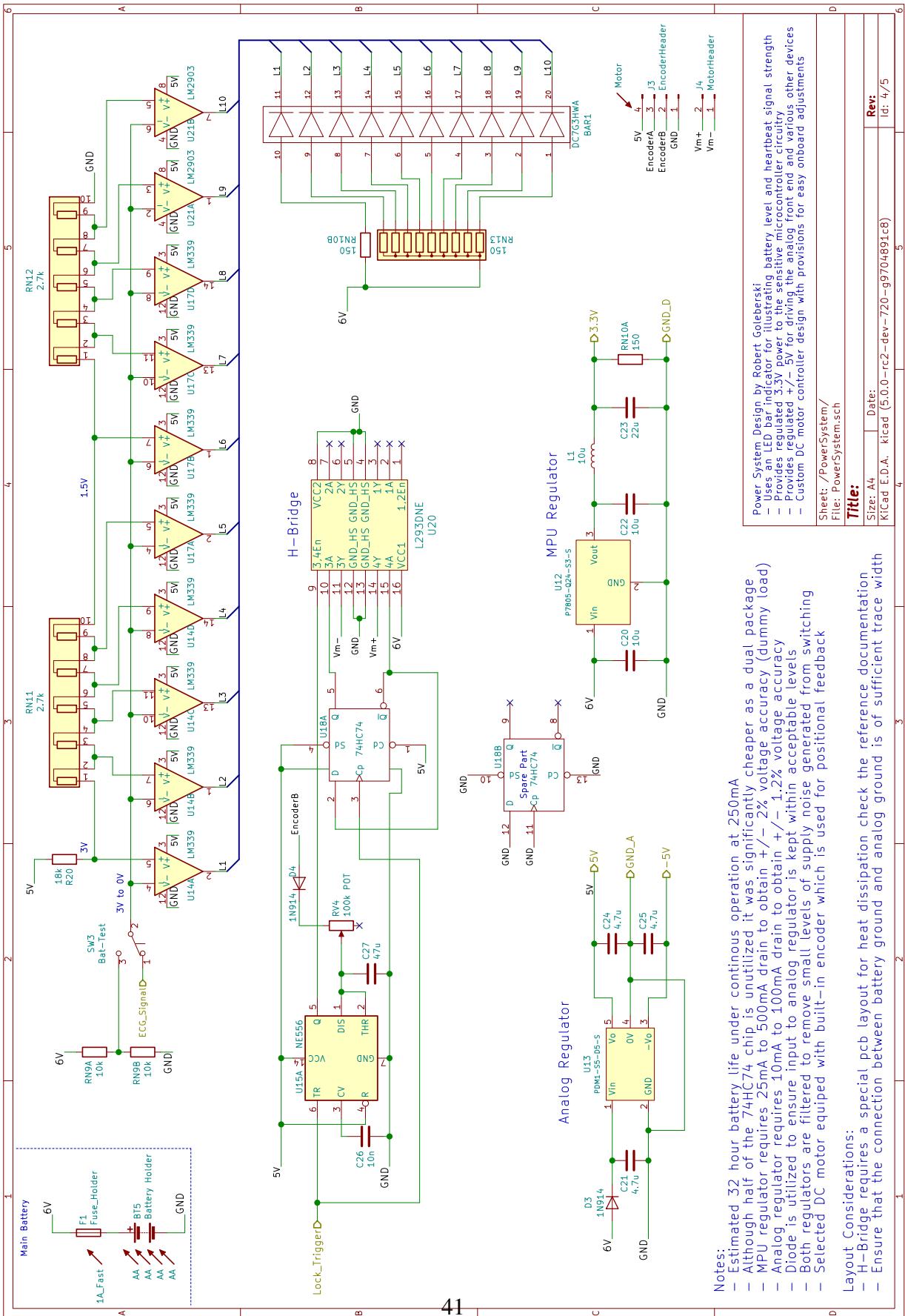
# Appendices

# A Circuit Schematics









## Notes:

- Estimated 32 hour battery life under continuous operation at 250mA
  - Although half of the 74HC74 chip is unutilized it was significantly cheaper as a dual package
  - MPU regulator requires 25mA drain to 500mA drain to obtain  $+/-2\%$  voltage accuracy (dummy load)
  - Analog regulator requires 10mA to 100mA drain to obtain  $+/-1.2\%$  voltage accuracy
  - Diode is utilized to ensure input to analog regulator is kept within acceptable levels
  - Both regulators are filtered to remove small levels of supply noise generated from switching
  - Selected DC motor equipped with built-in encoder which is used for positional feedback

## avout Considerations:

- H-Bridge requires a special PCB layout for heat dissipation check the reference documentation
  - Ensure that the connection between battery ground and analog ground is of sufficient trace width

Size:

Kicā

Power System Design by Robert Goleberski  
Using the **IE3D** software for illustrations

- Provides regulated + / - 5V for driving the analog front end and various other devices
- Custom DC motor controller design with provisions for easy onboard adjustments

Sheet: /PowerSystem/  
File: PowerSystem.sch

Title

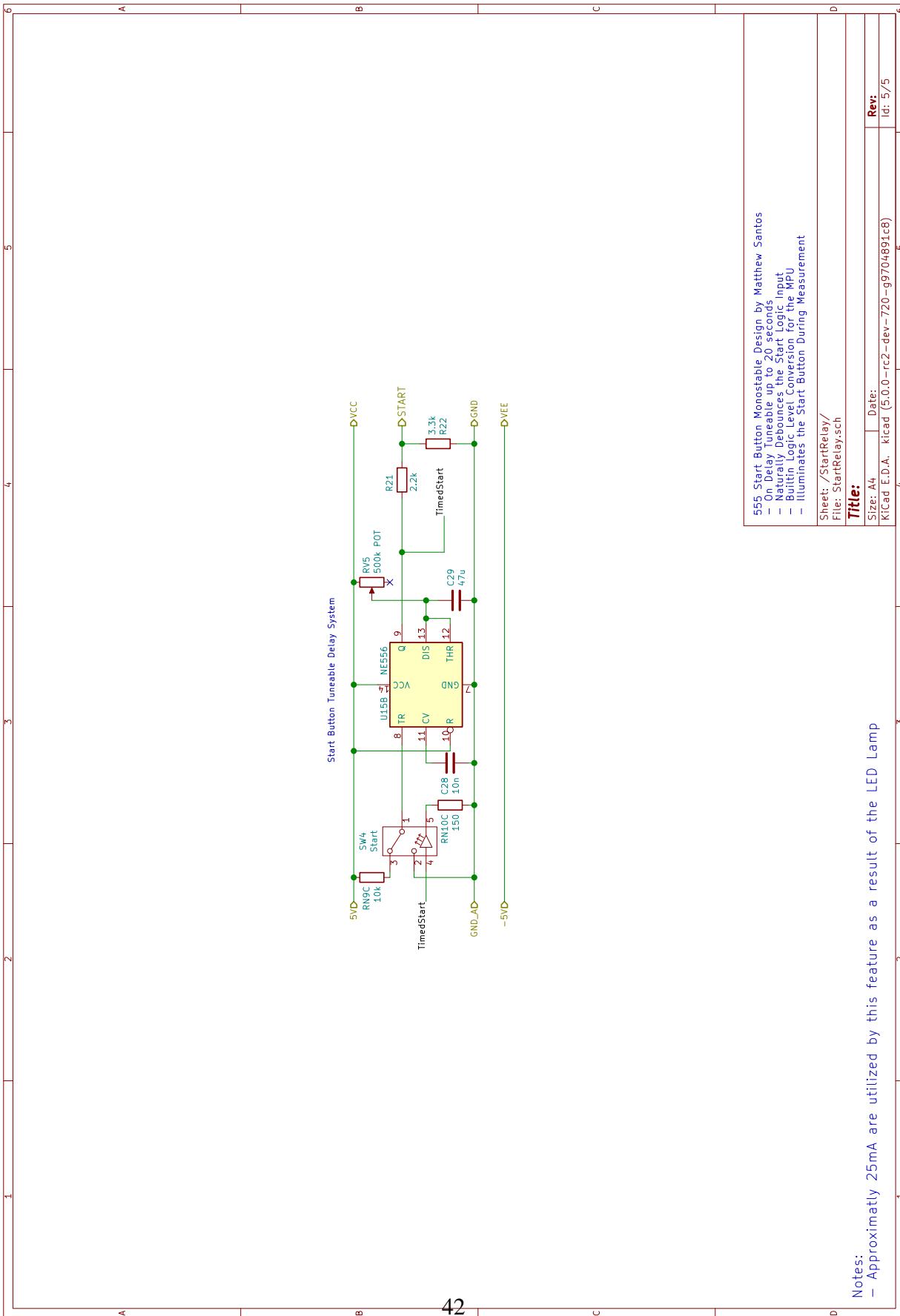
/ 111:

Size: A4 Date:

KiCad EDA kicad (5)

卷之三

41



## B Bill of Materials

**Bill of Materials for Capstone Group #9**

**Total # Parts 116**

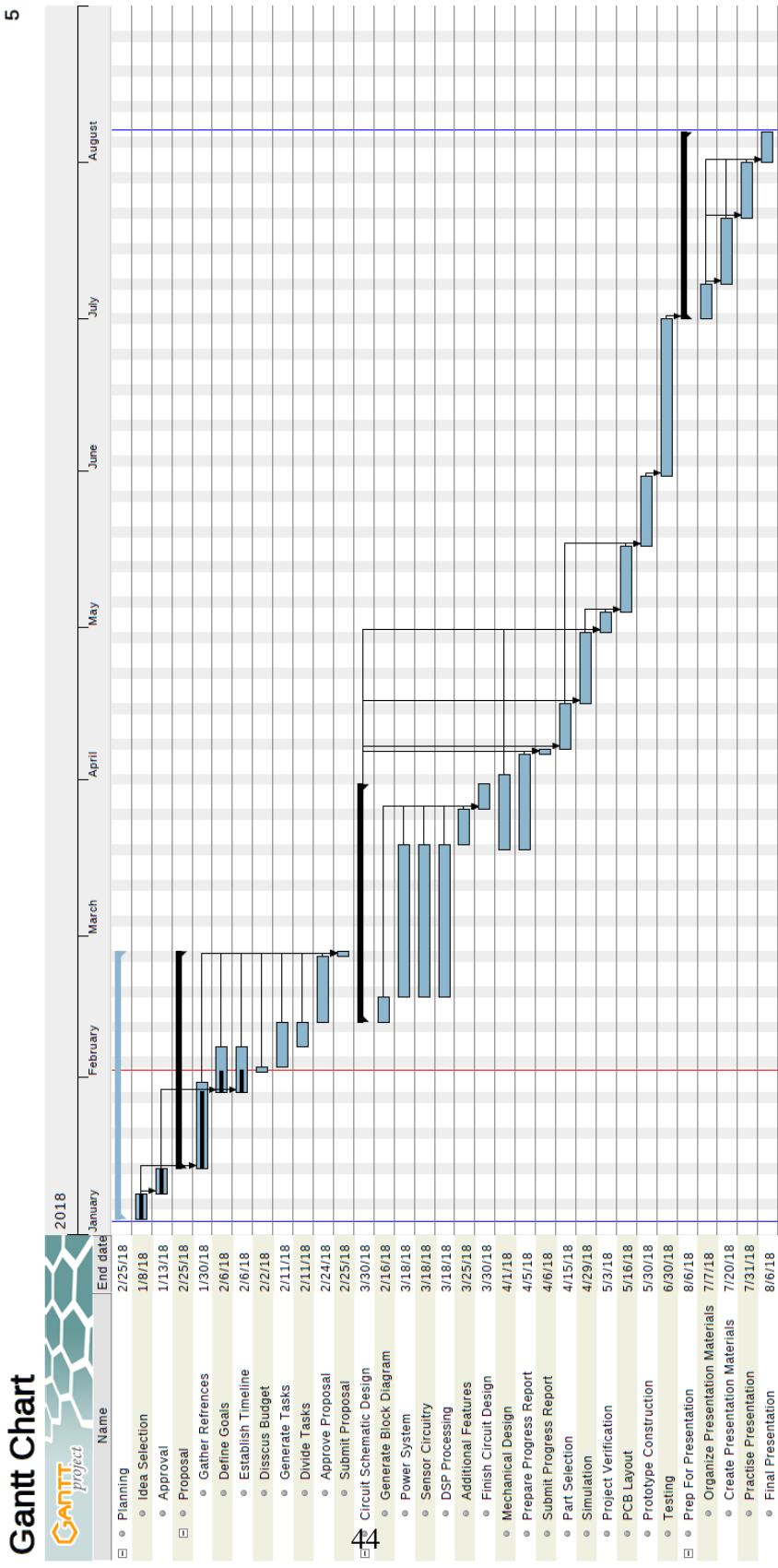
**Total Cost: \$220.50**

Part Info				Part#	Available	Unit Cost
Ref#	Ref Value	Manf#	Quantity			
BAR1	DC7G3HWA	DC7G3HWA	1	<a href="#">754-1671-5-ND</a>	43	\$2.73
BT5	Battery Holder	BH4AAPC	1	<a href="#">BH4AA-PC-ND</a>	2073	\$5.57
BT1-BT4	Holder	BK-5091	4	<a href="#">BK-5091-ND</a>	4220	\$0.42
C1,C2	27p	C1206C270F5GACTU	2	<a href="#">399-9339-1-ND</a>	3930	\$2.02
C10,C11	100p	885012008004	2	<a href="#">732-7864-1-ND</a>	3319	\$0.15
C12	1n	885012208022	1	<a href="#">732-8089-1-ND</a>	3391	\$0.14
C13,C15,C17	0.68u	885012208012	3	<a href="#">732-7684-1-ND</a>	2715	\$0.46
C14,C16,C19	0.068u	885012208086	3	<a href="#">732-8126-1-ND</a>	2550	\$0.20
C18,C27,C29	47u	GRM3LCC80J476KE18L	3	<a href="#">490-8337-1-ND</a>	20000	\$1.21
C20,C22	10u	C3216X5R1C106K160AA	2	<a href="#">445-4052-1-ND</a>	6752	\$0.42
C21,C24,C25	4.7u	C1206C475K4PACTU	3	<a href="#">399-1265-1-ND</a>	20000	\$0.43
C23	22u	C1206C226Z9VACTU	1	<a href="#">399-3028-1-ND</a>	16816	\$0.57
C26,C28	10n	C1206C103J5RACTU	2	<a href="#">399-1237-1-ND</a>	30522	\$0.33
C3,C4	15p	C1206C150J5GACTU	2	<a href="#">399-1194-1-ND</a>	34631	\$0.35
C8	4.7u	F931C475KAA	1	<a href="#">478-8265-1-ND</a>	221	\$0.56
C5-C7,C9	0.1u	C1206C104K4RACTU	4	<a href="#">399-9305-1-ND</a>	40000	\$0.20
D1	SMDA15TR	SMDA15TR	1	<a href="#">1655-1914-1-ND</a>	2500	\$0.72
D3,D4	1N914	1N914	2	<a href="#">1N914FS-ND</a>	23339	\$0.14
F1	Fuse_Holder	0031.8201	1	<a href="#">486-1258-ND</a>	44234	\$0.95
J1	ISCP	61300411121	1	<a href="#">732-5317-ND</a>	20525	\$0.25
J2	SD_Card	145638009211859+	1	<a href="#">478-4059-1-ND</a>	36044	\$4.22
J3	EncoderHeader	B4B-XH-A(LF)(SN)	1	<a href="#">455-2249-ND</a>	184591	\$0.29
J4	MotorHeader	B2B-XH-A(LF)(SN)	1	<a href="#">455-2247-ND</a>	147104	\$0.21
L1	10u	77F100K-TR-RC	1	<a href="#">77F100K-TR-RRCT-ND</a>	10402	\$0.42
R1,R4	1k	RC1206JR-071KL	2	<a href="#">311-1.0KERCT-ND</a>	155592	\$0.15
R10,R11,R19	2k	RC1206FR-072KL	3	<a href="#">311-2.0KFRCT-ND</a>	521394	\$0.15
R12	82k	RC1206FR-0782KL	1	<a href="#">311-82.0KFRCT-ND</a>	2699	\$0.15
R13,R14	1.5k	CRCW12061K50FKEAC	2	<a href="#">541-4184-1-ND</a>	32370	\$0.15
R15	9k	RC1206FR-079K09L	1	<a href="#">311-9.09KFRCT-ND</a>	4463	\$0.15
R17	40k	RC1206FR-0740K2L	1	<a href="#">311-40.2KFRCT-ND</a>	29843	\$0.15
R18	15k	AC1206FR-0715KL	1	<a href="#">YAG3815CT-ND</a>	29346	\$0.15
R2,R16	100	AC1206FR-07100RL	2	<a href="#">YAG3804CT-ND</a>	21159	\$0.15
R20	18k	RC1206FR-0718KL	1	<a href="#">311-18.0KFRCT-ND</a>	209262	\$0.15
R21	2.2k	RC1206JR-072K2L	1	<a href="#">311-2.2KERCT-ND</a>	244667	\$0.15
R22	3.3k	RC1206FR-073K3L	1	<a href="#">311-3.30KFRCT-ND</a>	547028	\$0.15
R3	10k	RC1206JR-0710KL	1	<a href="#">311-10KERCT-ND</a>	528040	\$0.15
R5	5.6k	RC1206FR-075K6L	1	<a href="#">311-5.60KFRCT-ND</a>	59375	\$0.15
R6	2.4k	RC1206JR-072K4L	1	<a href="#">311-2.4KERCT-ND</a>	140621	\$0.15
R7,R8	100k	RC1206FR-07100KL	2	<a href="#">311-100KFRCT-ND</a>	1184160	\$0.15
R9	3k	RC1206JR-073KL	1	<a href="#">311-3.0KERCT-ND</a>	5762	\$0.15
RN1	3.3k	4610X-102-332LF	1	<a href="#">4610X-2-332LF-ND</a>	1991	\$0.79
RN10	150	4606X-102-151LF	1	<a href="#">4606X-2-151LF-ND</a>	6101	\$0.62
RN11,RN12	2.7k	770103272P	2	<a href="#">770-103-R2.7K-ND</a>	1778	\$1.16
RN13	150	4610X-101-151LF	1	<a href="#">4610X-1-151LF-ND</a>	660	\$0.93
RN2	10k	4608X-102-103LF	1	<a href="#">4608X-2-103LF-ND</a>	9321	\$0.85
RN4	10k	4116R-1-103FLF	1	<a href="#">4116R-1-103FLF-ND</a>	26	\$1.31
RN5	1k	4606X-102-102LF	1	<a href="#">4606X-2-102LF-ND</a>	647	\$0.92
RN9	10k	4606X-102-103LF	1	<a href="#">4606X-2-103LF-ND</a>	813	\$0.62
RV1	5kPOT	PV36W502C01B00	1	<a href="#">490-2888-ND</a>	8861	\$2.23
RV2	3kPOT	3362P-1-302LF	1	<a href="#">3362P-1-302LF-ND</a>	1245	\$1.51
RV3	2kPOT	PV36W202C01B00	1	<a href="#">490-2880-ND</a>	2011	\$2.23
RV4	100k POT	PV36W104C01B00	1	<a href="#">490-2876-ND</a>	14427	\$2.23
RV5	500k POT	3296W-1-504LF	1	<a href="#">3296W-504LF-ND</a>	4284	\$3.58
SW1	SW_DIP_x06	418117270906	1	<a href="#">732-3836-5-ND</a>	16	\$2.63
SW2	Reset	TL1105F250Q	1	<a href="#">EG1828-ND</a>	19655	\$0.25
SW3	Bat-Test	EG1218	1	<a href="#">EG1903-ND</a>	36127	\$0.81
SW4	Start	UB15SKG035F-FF	1	<a href="#">360-1947-ND</a>	1787	\$11.65
SYM1	IC_Socket	ED281DT	1	<a href="#">ED3050-5-ND</a>	26059	\$0.46
SYM14	Motor	FIT0487	1	<a href="#">1738-1267-ND</a>	170	\$18.19
SYM2	OLED-100H016F	O100H016FBPP5N0000	1	<a href="#">541-3451-ND</a>	12	\$32.83
SYM3	LCD_Socket	PPTC082LFBN-RC	1	<a href="#">S7076-ND</a>	3122	\$1.38
SYM4	512MB SD Card	AF512SDI-OEM	1	<a href="#">AF512SDI-OEM-ND</a>	255	\$20.16
SYM9	1A_Fast	5MF 1-R	1	<a href="#">S07-1261-ND</a>	6689	\$0.29
SYM10-SYM13	AA	EN91	4	<a href="#">N107-ND</a>	203877	\$0.19
SYM5-SYM8	Cell	A76	4	<a href="#">N402-ND</a>	84722	\$0.79
TH1	Thermistor	DC103G9G	1	<a href="#">615-1054-ND</a>	8305	\$1.56

## C Project Schedule

### Capstone Cardless Security System

#### Gantt Chart



## D Octave SVM Generator Script

```
% Capstone ECG SVM Generator
% Written by Matthew Santos

pkg load signal;
pkg load ltfat;
ECG_Functions;
clear;clc;close all;
set(0, "defaultlinelinenwidth", 1.5);
graphics_toolkit("qt");

% Variables
sampleRate = 500;      %[Hz]
RecordNumbers = [1 2]; %[1:3 5 8:9 11:12 14:18 22:23 25:32 42:43 45:46
                   ↳ 49:53 58 60 62:64 66:69 71 75 78 79 81:83 86 88]; %RecordIndices

% Program Start
%-----
for Record=RecordNumbers
    [y,t] = ReadECG(Record,sampleRate);
    RawECG = Fractionalize(y(1,:),0.1);
    RawECG(Record,:) = Fractionalize(y(1,:),0.1);
    [CleanECG(Record,:)] PPFfilter =
        ↳ Preprocessing(RawECG(Record,:),sampleRate);
    Features{Record} = FeatureExtraction(CleanECG(Record,:),sampleRate,0);
end
[FeatureMean,FeatureStd] =
    ↳ GetFeatureNormFactors(Features,RecordNumbers);
SaveFeatureNorms('feature_norms',length(RecordNumbers),
    ↳ FeatureMean,FeatureStd);
Features =
    ↳ NormalizeFeatures(Features,RecordNumbers,FeatureMean,FeatureStd);
Classifiers = GetClassifiers(Features,RecordNumbers);
SaveClassifiers('classifiers',Classifiers);
%-----
% Program End

%Diagnosing (Feature Plots)
%-----
figure(1);clf;PlotClassifier3D(2,7,9,Classifiers,Features);
Fscores = GetFscores(Features,RecordNumbers);
AvgFscores = mean(Fscores,1);
figure(2);bar(AvgFscores);title("Feature Fscores Based on 50 sample
    ↳ ECGs","fontsize",20);axis([0 28 26 max(AvgFscores)]);
```

## E Octave SVM Testing Script

```
% Capstone ECG Tester
% Written by Matthew Santos

pkg load signal;
pkg load ltfat;
ECG_Functions;
clear;clc;close all;
set(0, "defaultlinelinenwidth", 1.5);
graphics_toolkit("qt");

% Variables
sampleRate = 500;      %[Hz]
TestRecords = 1:2;      %Test Records
%TestRecords = [1:3 5 8:9 11:12 14:18 22:23 25:32 42:43 45:46 49:53 58
%               ↳ 60 62:64 66:69 71 75 78 79 81:83 86 88];

% Program Start
%-----
for Record = TestRecords
    [y,t] = ReadECG(Record,sampleRate);
    RawECG = Fractionalize(y(1,:),0.1);
    CleanECG(Record,:) = Preprocessing(RawECG,sampleRate);
    Features = FeatureExtraction(CleanECG(Record,:),sampleRate,1);
    [NumRecords,FMean,FStd] =
        ↳ LoadFeatureNorms('feature_norms',length(Features));
    Features = NormalizeFeatures(Features,1,FMean,FStd);
    Scores(Record,:) =
        ↳ TestClassifiers(Features',NumRecords,'classifiers');
    [Access(Record) Identity(Record) Margin(Record,:)] =
        ↳ AuthorizeCheck(Scores(Record,:));
end
%-----
% Program End

%Diagnosing
%-----
for Record = TestRecords
    figure(1);clf;plot(CleanECG(Record,:));axis([1
        ↳ 500]);title("CleanECG");
    figure(2);bar(Scores(Record,:));title("Scores");
    figure(3);bar(Margin(Record,:)*100);title("ScoreMargin");
    if Record == Identity(Record)
        printf("Correct Identification #%d \n",Identity(Record));
    else
        printf("Failed to Identify #%d \n",Identity(Record));
    end
    pause();
end
```

## F Examples of Custom Octave Sub-functions

```
%FeatureExtraction, extract features from the filtered ECG signal
function Features = FeatureExtraction(CleanECG,sampleRate,testMode)
    R = RPeakDetector(CleanECG,sampleRate);
    DeltaRPeak = R(2:length(R))-R(1:length(R)-1);
    if testMode == 1 %Perform Sectioning
        [tmp,tmp,CleanECG] = Sectioning(CleanECG,sampleRate,R);
        R = round(0.5*min(DeltaRPeak)); %Obtain Avg R Peak
    else
        [Left,Right,tmp] = Sectioning(CleanECG,sampleRate,R);
        CleanECG = CleanECG/max(CleanECG); %Normalize Overall Amplitude
        for index=1:length(DeltaRPeak)-1 %Normalize Segments to Peak
            CleanECG(Left(index):Right(index)) =
                → CleanECG(Left(index):Right(index))/CleanECG(R(index+1));
        end
    end
    P = PPeakDetector(CleanECG,sampleRate,R);
    Q = QPeakDetector(CleanECG,sampleRate,R);
    S = SPeakDetector(CleanECG,sampleRate,R);
    T = TPpeakDetector(CleanECG,sampleRate,R);
    if length(R)>1 R= R(2:length(R)-1);end
    %Diagnosing Peak Detections
    %figure(1);clf;hold on;%plot(CleanECG);
    %plot(P,CleanECG(P), 'bo');printf("P=%d\n",length(P));
    %plot(Q,CleanECG(Q), 'ro');printf("Q=%d\n",length(Q));
    %plot(R,CleanECG(R), 'ro');printf("R=%d\n",length(R));
    %plot(S,CleanECG(S), 'ro');printf("S=%d\n",length(S));
    %plot(T,CleanECG(T), 'go');printf("T=%d\n",length(T));
    Features =
        → FeatureCalculations(CleanECG,P,Q,R,S,T,DeltaRPeak,sampleRate);
end
%NormalizeFeatures, normalize according to F=(F-mean)/std
function Features = NormalizeFeatures(Features,RecordNumbers,Mean,Std)
    if length(RecordNumbers) == 1
        Features = (Features-Mean). / Std;
    else
        for i=RecordNumbers
            Features{i} = (Features{i}-Mean). / Std;
        end
    end
end
%GetFeatureNormFactors, calculate the feature normalization factors
function [Mean,Std] = GetFeatureNormFactors(Features,RecordNumbers)
    Mean = [];Std = [];
    for i=RecordNumbers
        Mean = [Mean mean(Features{i}')'];
        Std = [Std std(Features{i}')'];
    end
    Mean = mean(Mean)';
    Std = std(Std)';
end
```

## G MPU Embedded Code Main Loop

```
// ECG Biometric Processor Code
// Written by Matthew Santos

//Includes
#include "xc.h"                      //Variable References
#include "Config.h"                    //Configuration Settings
#include "Variables.h"                 //Program Variables
#include "Interrupts.h"                //Interrupt Definitions
#include "Functions.h"                 //Function Definitions
#include "LCD_Functions.h"             //LCD Function Definitions

//Program Start
int main(void) {
    startup();                           //Execute Startup Procedures
    LCD_Initialize();                   //Initialize the LCD
    LCD_SendString(" Cardless Security"); //Reset Message Line 1
    LCD_L2();                          //Move LCD Cursor to Line 2
    LCD_SendString(" System");          //Reset Message Line 2
    Delay(5000);                       //Message Display Delay
    LCD_Clear();                        //Clear the Display
    LCD_L1();                           //Set LCD Cursor to Line 1

    while (1) {
        LCD_Send(LCD_CMD_OFF,0);        //Hide the Display
        LCD_Send(LCD_CMD_PWROFF,0);     //Power Off the Display
        Sleep();                         //Wait For Interrupt
        LCD_Send(LCD_CMD_ON,0);          //Show the Display
        LCD_Send(LCD_CMD_PWRON,0);       //Power ON the Display
        if(Status.start){               //Check if Start Was Pushed
            CollectData();              //Start Sampling
            WriteECGDataToSD("Raw.txt"); //Transfers ECGData to SDCard
            Preprocessing();            //Start Preprocessing
            WriteECGDataToSD("ECG.txt"); //Transfers ECGData to SDCard
            Sectioning();               //Start Sectioning
            WriteECGDataToSD("Sec.txt"); //Transfers ECGData to SDCard
            FeatureExtraction();        //Start Feature Extraction
            FeatureNormalization();     //Perform Feature Normalization
            TestClassifiers();          //Use Support Vector Machine
            authorizeCheck();           //Analyze Results
            LockReply();                //Perform Locking Action
            Status.start = 0;            //Clear the start flag
        }
        else{                           //Nothing should wake CPU
        }
        if (Status.Error){              //If Error Occurred
            LCD_Clear();LCD_L1();      //Clear LCD
            LCD_SendString("An Error Occurred"); //Notify User
            Delay(1000);                //Message Delay
        }
    }
}
```

## H Examples of an Embedded Function and Interrupt

```

void CollectData() {
    LCD_Clear();                                //Clear Display
    LCD_L1();                                   //Goto Line 1
    LCD_SendString("      Get Ready");          //Update User Status
    int i;
    for (i=0;i<16;i++) {                      //Every 500ms
        LCD_Bar(i);                            //Update the Progress Bar
        Delay(500);
    }
    AD1CON1bits.ADON = 1;           //Turn ON ADC1 (rest is interrupt driven)
    LCD_Clear();                                //Clear Display
    LCD_L1();                                   //Goto Line 1
    LCD_SendString("Measuring ECG");            //Output Status Indicator
    LCD_Bar(0);                                 //Output Progress
    while(!Status.sample){                   //Wait for Measurement to Finish
        LCD_Bar(16*ECGBlocksReady / (ECGLength/ADC1_BufferSize)); //Update
        Progress
        if (ECGBlocksReady == ECGLength/ADC1_BufferSize){
            Status.sample = 1;                  //Measurement Finished
            LCD_Bar(16);                     //Update User Status
        }
    }
}

//DMA0 Interrupt (ADC-1)
void __attribute__((__interrupt__, no_auto_psv)) _DMA0Interrupt(void) {
    static unsigned char DMA0ActiveBuffer = 0; //Active Buffer
    register int DSP_ACCA asm("A"); //Define DSP accumulator A
    if (DMA0ActiveBuffer == 0) { //For Active Buffer
        //Append Active Buffer to ECGData
        VectorCopy(ADC1_BufferSize, &ECGData[ADC1_BufferSize *
            ECGBlocksReady], &DMA0_BufferA[0][0]);
        //Take one Point of Temperature Data and Add it to Rolling
        Average
        DSP_ACCA = __builtin_mpy(DMA0_BufferA[1][0], Q15((float)
            ADC1_BufferSize / (float) ECGLength), NULL, NULL, 0, NULL,
            NULL, 0);
        AvgTemp = AvgTemp + __builtin_sac(DSP_ACCA, 0);
    } else {
        //Append Active Buffer to ECGData
        VectorCopy(ADC1_BufferSize, &ECGData[ADC1_BufferSize *
            ECGBlocksReady], &DMA0_BufferB[0][0]);
        //Take one Point of Temperature Data and Add it to Rolling
        Average
        DSP_ACCA = __builtin_mpy(DMA0_BufferB[1][0], Q15((float)
            ADC1_BufferSize / (float) ECGLength), NULL, NULL, 0, NULL,
            NULL, 0);
        AvgTemp = AvgTemp + __builtin_sac(DSP_ACCA, 0);
    }
    ECGBlocksReady++; //Update Number of ECG Blocks Ready
}

```

```
if (ECGBlocksReady == ECGLength / ADC1_BufferSize) { //When Enough
    → Blocks
        AD1CON1bits.ADON = 0; //Turn OFF ADC1
    }
    DMA0ActiveBuffer ^= 1; //Toggle Active Buffer
    IFS0bits.DMA0IF = 0; //Clear the DMA0 Interrupt Flag
}
```