



Final Design Report

Senior Design Project

Team 05

May 10, 2018

Shurjo Maitra

Chinelo Ezenwelu

Temitope Ogunnoiki

Abstract

Winter is an especially harsh time for people residing in Michigan. As a result most of us wear layers of clothing each with a different function which restricts mobility and reduces aesthetic appeal. Ease of access and exposure to inside and outside temperatures is also a problem as a person would have to take off their coats and continuously re-adapt to various temperatures. To make winter clothing more efficient our team created a light, comfortable and smart heated jacket: HotJac. This jacket is electrically heated and can be controlled manually as well as through an android application.

Table of Contents

Abstract	2
Table of Contents	3
Table of Figures	6
Table of Tables	7
1 Introduction	9
1.1 Calvin College Engineering Department	9
1.2 Senior Design for Team 5	9
1.3 Meet Team 05	10
1.4 Project Summary	11
2 Project Design	12
2.1 Current Design	12
2.2 Project Background	14
2.2 Research Techniques	14
2.3 Design Methodology	15
2.4 Design Criteria	15
2.5 Design Criteria for Entire System	15
2.5.1 Generic	15
2.5.2 General	15
2.5.3 Accessibility	16
2.5.4 User-Friendly	16
2.6 Design Criteria for Hardware	16
2.6.1 Heating Element	16
2.6.2 Bone Conduction headphones	16
2.6.3 Real-Time Updates	16
2.6.4 Central Management	17
2.6.5 Sensors	17
2.6.6 Display Interfaces	17
2.6.7 Server	17
3 Design Constraints	17
4 Design Decisions	18

4.1 Heating element	18
4.2 Battery	19
4.3 Microcontroller	20
4.4 Heating	21
4.5 Bluetooth Bone Conduction Headphones	23
4.6 Battery and Charging Elements	24
5 Safety	25
5.1 Possible Hazards	25
5.1.1 Battery	26
5.1.2 Audio interface and bone conduction headphones	27
5.1.3 Heating element	27
5.1.4 Power supply and microcontroller	27
6 Testing	27
7 Results	28
8 Future Work	28
9 Design Norms	29
9.1 Our Approach	29
9.2 Design Norms Explained	29
9.2.1 Trust	29
9.2.2 Care	29
9.2.2 Integrity	30
10 Project Management	30
10.1 Team Organization	30
10.2 Advisors	32
10.3 Schedule	32
11 Business Plan	32
11.1 Budget	32
11.2 Market Analysis	33
11.2.1 Market Survey	33
11.3 Competition	33
11.4 Cost Estimate	35
11.5 Development Costs	35
11.6 Production Costs	35
12 Conclusion	36
13 Acknowledgements	36

14 References	36
15 Appendices	38
A.1 Budget Review	38
A.2 Schedule	41
A.3 Arduino Code	44
A.4 Android code	46
A.5 PSpice diagrams of Components	58

Table of Figures

Figure 1 System Block diagram	12
Figure 2 Lab Project Block diagram	3
Figure 3 Jacket with heating pads on the inside.....	18
Figure 4. Fritzing diagram of Bluetooth and temperature sensor circuit.....	20
Figure 5: Heating areas for hot-JAC ..	22
Figure 6: Heat Rating with Time of Individual Heating Pad.....	23

Table of Tables

Table 1 Battery Comparison20

Table 2 Budget Analysis24

Table 3 Actual Cost for Design.....40

1 Introduction

1.1 Calvin College Engineering Department

The Engineering program at Calvin College is an ABET-accredited program with the aim of educating students to serve the Lord and others through technology, with a strong emphasis on engineering design. In essence, the engineering program seeks to equip students to glorify God by meeting the needs of the world with responsible and caring engineering.

1.2 Senior Design for Team 5

The engineering department includes the Senior Design Project in the student curriculum as the final requirement for students to graduate with the engineering degree. This project mimics an industrial project setting and is spanned for two semesters, approximately eight months. The project starts in the fall semester with senior engineering students forming groups of 3-4, these group of students are then paired up with advisors who work the students, consulting and guiding them on the project from inception to execution. Team 05 comprises of three senior electrical and computer engineering students with similar interest and like-minded goals. Together, and under the supervision of Prof. Mark Michmerhuizen, the team chose to work on a smart heating jacket for consumer comfort.

1.3 Meet Team 05

Chinelo Ezenwelu - Electrical and Computer Engineering



Chinelo Ezenwelu is a Senior Electrical & Computer Engineering student from Anambra, Nigeria. She also has minors in Economics and Mathematics and is currently an Honors student at Calvin College. This past summer, she worked as a Controls Engineering Intern for Highlight Industries, a reputable packaging firm in West Michigan. At the internship she worked on designing, programming and testing the packaging machines. Chinelo likes to cook, vlog and engage in some outdoor activities in her free time.

Temi Ogunnoiki - Electrical and Computer Engineering



Temi Ogunnoiki is also a Senior Electrical & Computer Engineering student from Lagos, Nigeria. She is minoring in Computer Science and is an honors student. Last summer, she worked with Attwood Marine, a company that makes boat motors in Lowell, Michigan. She worked on helping them build an Android Application to run the motors. Besides school, Temi enjoys dancing, and all things fashion.



Shurjo Maitra - Electrical and Computer Engineering

Shurjo Maitra is a Senior Electrical and Computer Engineering student from India. He is minoring in Computer Science at Calvin College. This past year he has worked as a Research assistant on Solar Power, Spectrometry and High-power LEDs. Shurjo is an avid climber and mountaineer and loves to travel.

1.4 Project Summary

At the start of the year, our team was brainstorming on a potential project for senior design. One of the alternatives we considered was a tracking device for climbers to record data on their physical activities and provide feedback. Another alternative and our current project is the heating jacket - Hot-JAC - a smart heating system in-built in a jacket for customer comfort.

Part of the reasons we chose this project among others was because of how much it resonated with us. As international students, transitioning from tropic regions to a temperate region was one of our biggest culture shock and so with our project, we hoped to solve the problem of having to wear bulky layers during winter especially for those living in temperate regions. Furthermore in our research, we noticed in our market analysis that even though there were heating jackets in the market, they were being sold at outrageous prices therefore another goal for this project was to use simple materials to design a cheap, quality product.

2 Project Design

2.1 Current Design

Currently, our design incorporates seven main components: heating elements, PWM (Pulse Width Modulation) drive, Bluetooth module, power supply, Bluetooth module, android phone and an Arduino microcontroller. There are two sides of the design that are simultaneously being developed. Firstly an Android App has been developed to control the basic functionality of the heating pads through the PWM, using connection from a Bluetooth module. It also has multiple fragments namely: instructions, control page and battery status. The main fragment however is the control fragment, which includes turning on and off of the heating elements as a whole system. The second part of the project is focused on delivering and controlling power to the heating pads. This is done through the Arduino uno and PWM. The PWM is powered by an 11.1V, 6000mAh Lithium-Ion battery, and outputs to the heating pads which are connected in parallel. Using jumper cables, the analog signals of the PWM are connected to the Arduino to be read as digital and understood by the Android App. These signals were used to control the duty cycle, which in turn controlled the power the heating pads received, which then helped control the temperature. The BC127 audio board, despite not currently in use, is also connected to the Arduino along with the HC-05 Bluetooth module which is used to connect to the Android App. This is not in use because it impossible to communicate with two modules at the same time unless the master device is dual-band, which is not the case for the HC-05 - it has a single band. So, the audio aspect of the jacket had to be removed. Finally, the team decided to split the heating elements to focus on heating the body's core to ensure adequate heating throughout the jacket. In the jacket, we decided to use cotton as a base layer, because it's a breathable material, and wool to stuff it (due to high melting point) as the insulating layer. It

is possible for wool to burn at very high temperatures of about 400 °F, but it is self-extinguishing. The outer layer is going to be covered using Gore-tex® - which is a waterproofing material to prevent water from rain or melted snow to penetrate the jacket. Overall, the current design has satisfied most of our requirements, testing and heat modeling was conducted to ensure that the jacket’s functionality is as proposed. Below is the current system diagram:

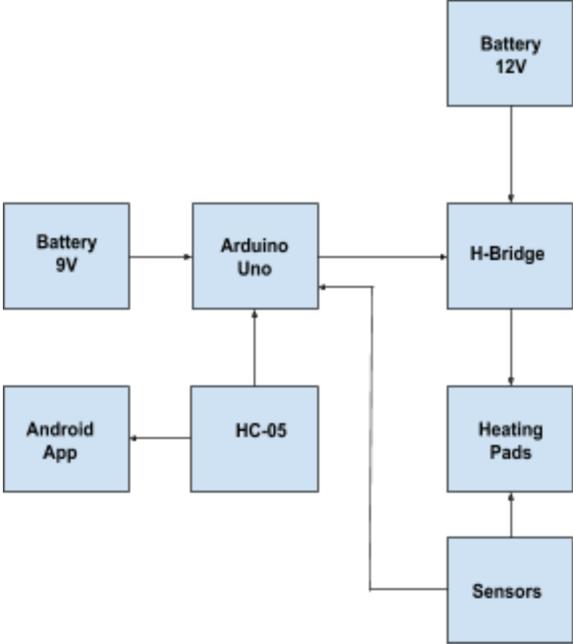


Figure 1. System Block diagram

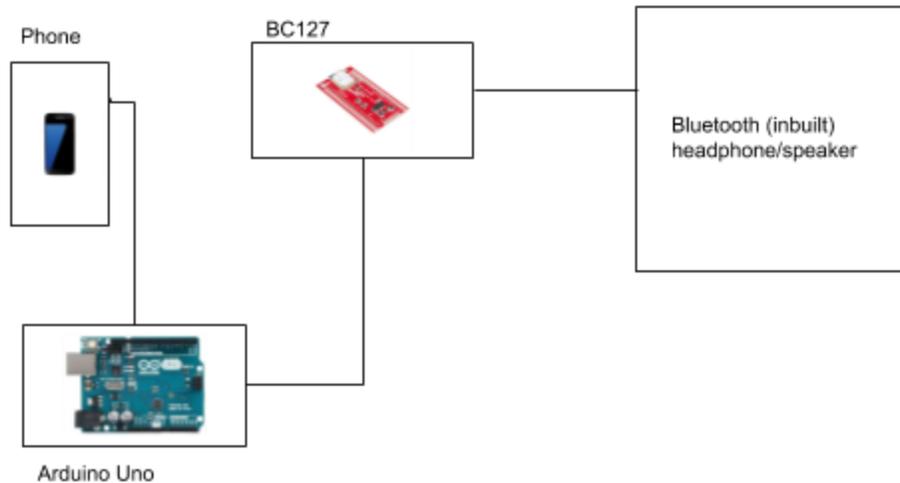


Figure 2. Lab Proposed Block diagram for Headphones

2.2 Project Background

This project design originated from solving basic needs for those living in temperate regions. Therefore, the key features are heating and sound delivery. From those requirements, the team branched to the idea of a centralized on-jacket control system that could implement temperature control and an external control, i.e. an Android device that could be used for multiple smart features.

2.2 Research Techniques

The team has been using research papers as well as commercial projects that have used the features that would be good to incorporate in our design. Currently, electric carbon fiber heating and metal mesh systems are being implemented in the projects that are commercially available. However, with enough consideration, and to suit our design process, we have decided to use heating pads which were constructed with a mesh of Polyester filament and Micro Metal Conductive Fiber folded into protective Polyimide Film. The team aims to build the most efficient designs and use data collection to optimize our design

incrementally. Simple projects like Motor Control using an Arduino, sensor readings, as well as communication with Bluetooth, were all researched and implemented into our projects. Since there was not a lot of information regarding how to control a heating system with an H-Bridge using pulse width modulation, the team modelled the system using motor control, but instead of controlling the system in two directions controlled it as one.

2.3 Design Methodology

The aim of our team is to incorporate better and more efficient components or technologies when they are discovered. The team aspires to keep the core design the same but incrementally improve each design iteration whether it be through research, simulation or experimental data.

2.4 Design Criteria

The team considered two parts of the project, considering what the end user needs to see and feel from the jacket and the hardware functions that are needed to execute those. The following are the current criteria:

2.5 Design Criteria for Entire System

2.5.1 Generic

- Ability to wear and be assured of comfort and safety
- Transferable - ability to replicate design on another jacket/item of clothing
- Ability to communicate and control features using an Android app

2.5.2 General

- Ability to heat using the inbuilt element for a safe duration and throttle when required
- Price is less than other heating jackets
- Ability of system to work with and without all features
- Ability to add and remove heating element and bone conduction headphones

2.5.3 Accessibility

- Data (ex. temperature, up-time, etc.) should be accessible through app
- Detailed data (voltage, battery health, etc.) can be found through using advanced mode

2.5.4 User-Friendly

- Sensors should be seamlessly integrated in the circuit, and;
- Analog keys/switches in jacket and app should be intuitive

2.6 Design Criteria for Hardware

2.6.1 Heating Element

- The device and circuits are light and have the ability to uniformly heat all parts of the jacket
- Heating is enough to keep the whole upper body warm and works for a reasonable duration
- Lithium battery is waterproof and non-penetrable.
- Temperature is easy to control and system is responsive

2.6.2 Bone Conduction headphones

- Device works seamlessly to provide a high quality audio experience
- App integration and ability to implement all modern personal audio controls (ex. pause, next, etc)

2.6.3 Real-Time Updates

- Updates temperature, battery remaining and stores data
- System changes (ex. temperature decrease) are indicated and notifications are sent

2.6.4 Central Management

- App stores all data collected and user inputs to implement a better automated system

2.6.5 Sensors

- Used to determine if jacket is being used
- To detect if heating device gets above certain temperature
- Battery remaining
- Easy enough to make system integrated with any unsightly change
- To be able to speak to other elements in the jacket
- Bluetooth pairing is done, headphones are turned on and connected

2.6.6 Display Interfaces

- Android app will show most of the information
- Few on-jacket touch buttons and LED indicators will be used to show status of heating element (internal temperature), battery and headphone connection

2.6.7 Server

- A reliable computer or network to connect and store data from app, running 24x7
- Host the website and mobile application that users will fetch and store data in
- Store the database of information for temperature and data collected from the sensors

3 Design Constraints

There are several parts to this project that act as constraints to the final design. Expense and time are the major constraints in that the team does not have the liberty to experiment with new technologies or make our system very complicated. Since the project is something that is in direct physical contact with users, factors like comfort, safety and ease of use have precedence over smart features. The team also considers the environmental footprint of the project and therefore would like to make the design sustainable and not use toxic materials even if they deliver better performance. The design should also be repeatable therefore the manufacturing process should be easy. Most of the above considerations are usually made in the area of smart clothing so, this project does not have any major impeding constraints that could halt the project's progress.

4 Design Decisions

4.1 Heating element

There a number of heating technologies in the market that can be used for the purpose we want; the ones that we could build upon are silicone rubber heaters, ultraflex heaters, Polyester Flexible heaters and electric carbon fiber. However, with enough consideration, and to suit our design process, we have decided to use heating pads which were constructed with a mesh of Polyester filament and Micro Metal Conductive Fiber folded into protective Polyimide Film. The heating pad works at an operating voltage of 5V DC and an operating Current: $\sim 600\text{mA}$ ($\sim 8.3\Omega$) and we are working with two different sizes, the 5 x

15 cm heaters and the 5 x 10 cm heaters. Three 5 x 15 cm heating pads were used; two were placed around the chest area, while another one was at the upper part of the back. Also, two 5 x 10 cm heating pads were used and placed at the middle of the back area.



Figure 3: Jacket with heating pads on the inside

4.2 Battery

A 12V, 6000mAh battery was used to power the whole system, the battery pack was initially supposed to power the Arduino as well, but the team did not want to tamper with the wire connection in place for the ready-made battery pack. Instead, a non-rechargeable 9V Alkaline battery was connected straight to the Arduino board, and was used to power it. The 12V battery was purchased from Amazon (but it has an operating voltage of 11.1V, and a nominal voltage of 12V), running at maximum power the battery would run for about one and a half hours on a power rating of 5000 mAh. Another problem with the battery pack is that it weighed about 1 lb, which was pretty heavy on the jacket, if we were to make changes, we would

reduce or redistribute the weight of the battery and only use one. As seen in Table 1 below, Lithium-Ion had the highest internal resistance, no memory effect, medium environmental impact, and a very wide power rating range which made it the best choice.

Table 1: Battery Comparison

	Lead Acid	NiCd	NiMH	Li Ion	Li Polymer	Alkaline
cap/vol (Wh/kg)	10-50	10-80	30-90	30-300	100-265	80-175
Specific Power	25-415	50-1000	6.02-1100	8-2000	250-730	4.35-35
Memory effect	Low	High	Medium	None	None	None
Min V before damage (per cell)	1.97	1	1	3	3.2	1.1
Internal resistance	0.08	115m	320m	778m	6m	0.15
Environmental impact	High	High	High	Medium	High	Medium

Source: <https://rogershobbycenter.com/lipoguide/>

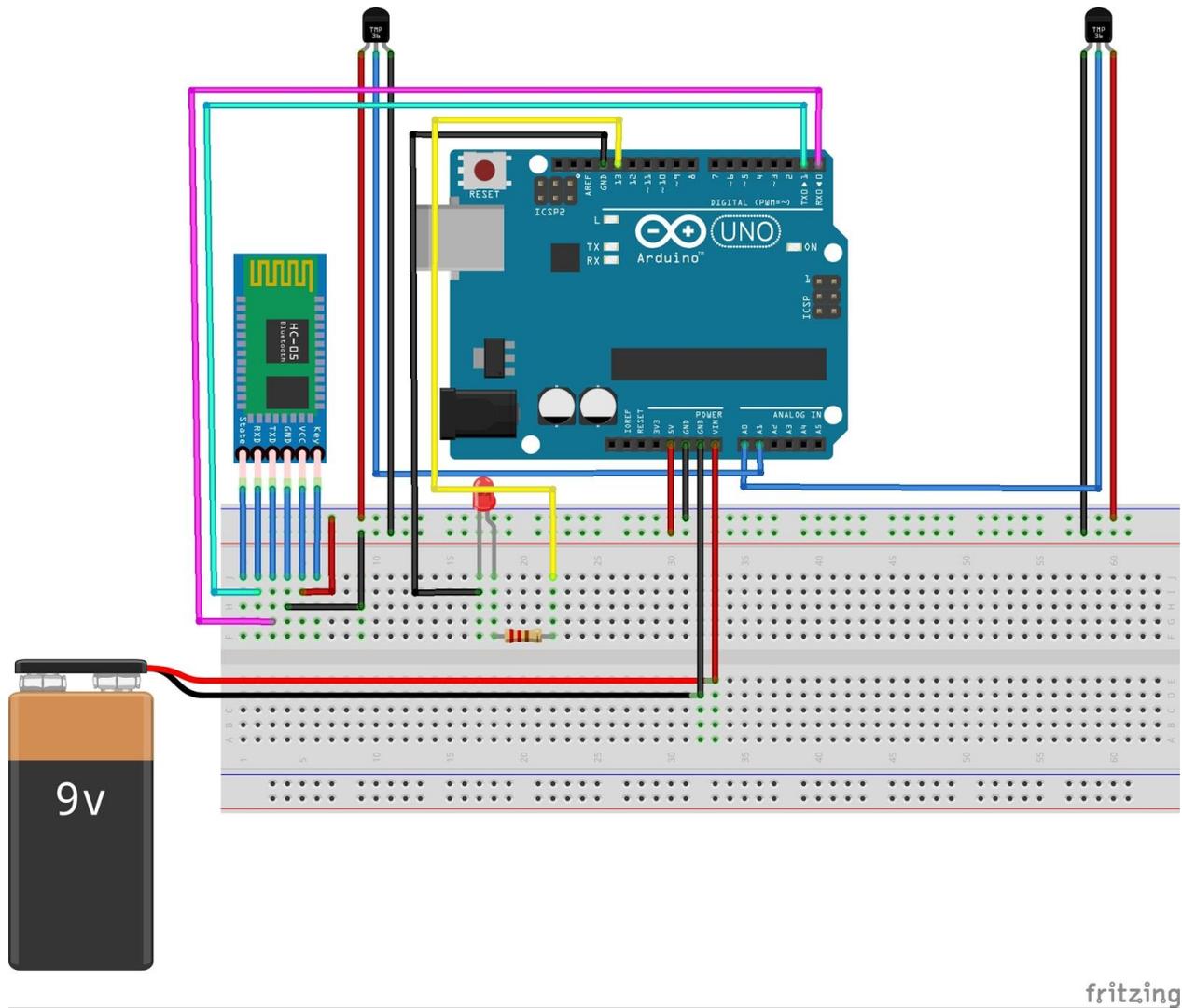


Figure 4. Fritzing diagram of Bluetooth and temperature sensor circuit

4.3 Microcontroller

The team used an Arduino Uno as the microcontroller for the system. It was used to: control the LED to show brightness proportional to heat intensity of the jacket; to control the duty cycle delivered to the H-Bridge thereby controlling the power entering the heating pads, and establish connection between the jacket, HC-05 and the Android application. Refer to Appendix for code.

Sensors: The team used TMP-36 sensors to check the temperature on each one of the heating pads and the information was read into the arduino. The sensors are low voltage temperature sensors that provide a voltage output linearly proportional to the temperature. It also covers a range of -40°C to $+125^{\circ}\text{C}$, which is very sufficient for our project. By Senior Design Night, we did nothing with the information given to the Arduino, but we extended the code to be able to read what the temperature on the sensors and display it on the app. The code has been modified to adjust the jacket to a certain temperature if the sensors read a range between certain temperatures, to accommodate change.

4.4 Heating

Heating is the most important design aspect of this project. To first break it down, the team examined the hazards that come with creating a heating garment and preventive measures with the different approaches. The jacket would heat two main areas: the chest and the back as seen in Figure 5 below.



Figure 5: Heating areas for hot-JAC

Source: <https://www.ventureheat.com/winter-sports/127-men-s-heated-insulated-jacket-only.html>

Heating these two areas should be enough to warm up most of the body, and keep a steady temperature. The reason why the middle chest and upper back areas were chosen is because, the chest area getting cold can lead to a lot of health difficulties, like asthma, so it is important to keep that area warm. Also, the jacket pockets can help keep fingers warm from the heating in the front part. The fingers are one of the many parts of the body that lose heat very quickly. Also for the back, it is related to breathing as well, so keeping that area warm can lessen chances of catching a cold, or asthma. To be able to trap the heat created by the heating layer, the design will be surrounded by cotton and wool on the innermost layers. This would help reduce heat loss and allow the jacket run for a longer time while also maintaining a certain temperature in the jacket. The heating range of the jacket would typically range around 60 degrees Fahrenheit, but could reach a maximum of 90 degrees Fahrenheit. Anything higher could cause discomfort to the human skin, the heating pads are set to a temperature to limit of 90 degrees Fahrenheit to avoid this.

For the heating element, the team narrowed it down to two different options based on their research. The team was looking for something flexible that could fit perfectly into clothing. Also, we were looking for something that was very affordable, something that would fit into the system without having contact with the person's skin, and finally something that could be controlled with an Arduino to enable a functioning controllable system. The two elements narrowed down were Carbon heater tape and heating pads. These two systems fit most of our requirements but the team later chose Sparkfun heating pads because we were sure the pads could be controlled by an ardition and we were not too sure about the tape. Also, there was more information about the heating pads online than for the tape, so if we ran into error we had places to refer to for guidance.

“The DC powered heating pads are perfect for near-body heating applications. They get warm to the touch but not too hot as long as you provide the appropriate voltage. Simply apply 5VDC to the wire leads and within minutes, the pad will begin to warm up. Refer to Figure 4 for Heating chart:

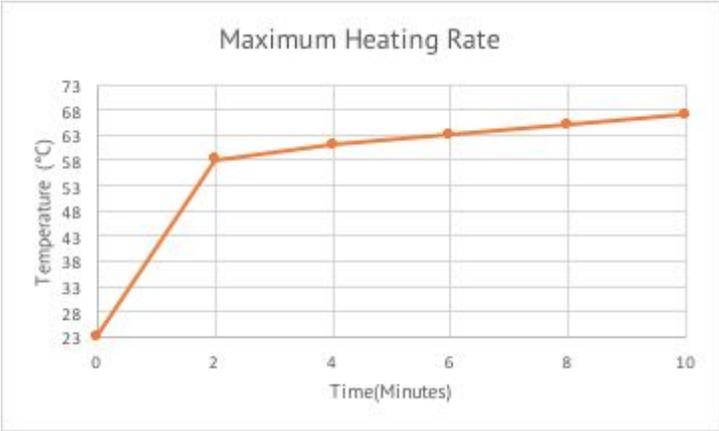


Figure 6: Heat Rating with Time of Individual Heating Pad

These heating pads are constructed using a mesh of Polyester filament and Micro Metal Conductive Fiber folded into a protective Polyimide Film. The fact that these are low power, flexible and draw little power makes them ideal for things like hand-warmers and other heated garments.” (Sparkfun, 2015)

Refer to Appendix A.3 for Arduino code details on the PWM duty cycle conversion to temperature.

4.5 Bluetooth Bone Conduction Headphones

For future addition to the jacket, the team plans to add wireless bone conduction headphones to the hood of the jacket. Bone conduction allows you to hear sound through vibration of the bones on your face.

When this happens, sound waves are directly stimulating the inner ear. For a long time, bone conduction has been used as a tool to help those with hearing loss. In this case, the team would be using it to allow only the wearer of the jacket to listen to music, and answer phone calls using Bluetooth connection to the user’s device.

The team plans to build their own bone conduction earphones using some sort of elastic headband, to be able to grip the user’s head, flat wire to prevent bulging, an electric microphone, and a Bluetooth audio module. The headphones would be powered by main power supply - the lithium ion batteries that are attached to the jacket.

The headphones would be least priority in the design, this is because bone conduction headphones do not sound as good as regular headphones, because sending vibrations through the jaw, is much harder than sending it through the air. Also, the feeling of bone conduction headphones can be a bit unsettling, the vibrations could feel like something tapping by the side of your head, which can eventually get annoying, but one can get used to it.

The team has also decided, if the bone conduction headphones can not be built by the time the project is due, an already made pair of headphones could be attached to the material, or the headphone idea could be completely scrapped.

4.6 Battery and Charging Elements

The team used a Lithium-ion battery pack to power the jacket. For safety reasons, this battery pack is detachable from the whole system. This detachable option also helps make it easy for the battery to be charged. Lithium-ion battery was chosen as the best choice because it has an extended shelf-life, high capacity, low internal resistance and reasonably short charge time, compared to Lead acid battery and Lithium-Polymer.

Lithium-ion batteries have also been commonly used for clothing, therefore they come in very flexible forms. The battery produces between 11.1V and the jacket produces about 11W of power to be able to keep a human warm. This was done based on what is commonly used in electric textiles.

The battery would be charged using a power cable provided to the customer/owner of the jacket. Full battery charging would usually take from 2-3 hours, depending on the charge voltage per cell. As mentioned earlier, the battery would last about 2-3 hours, because it would not be using the maximum capacity as the PWM values have been limited by the Arduino.

4.7 Software Design

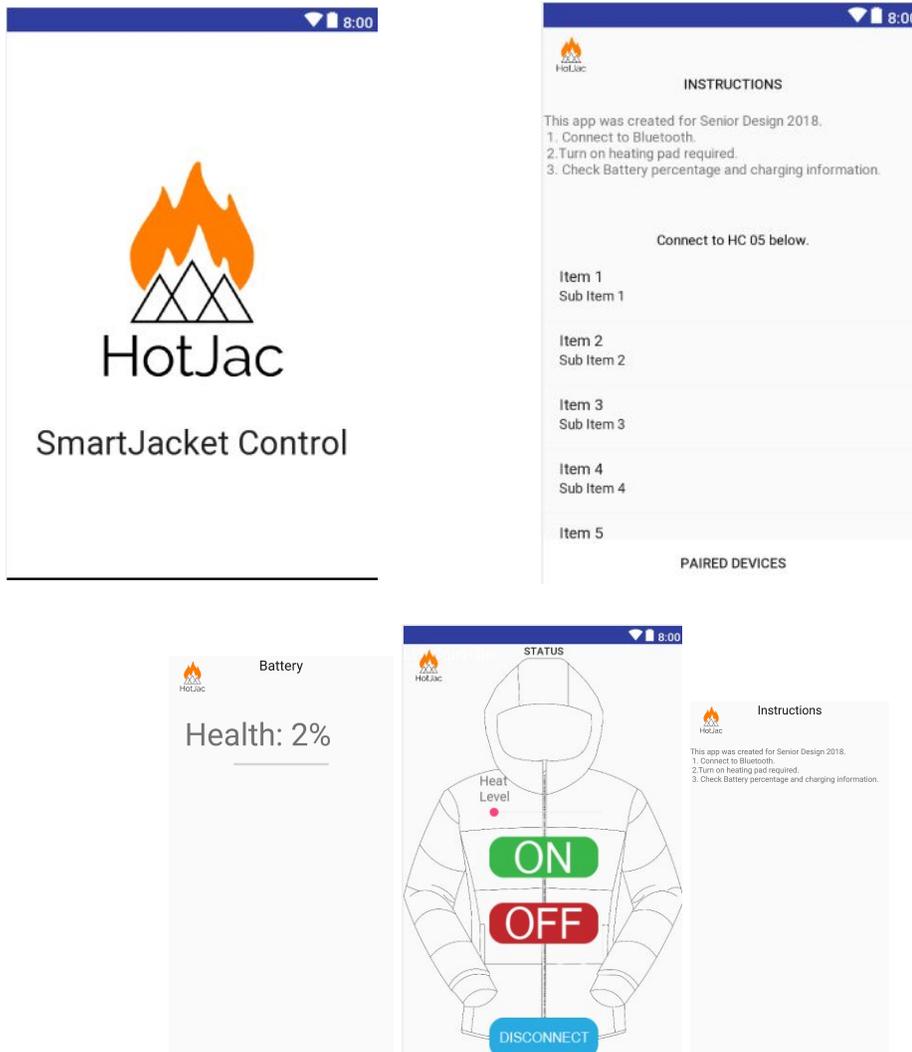


Figure 7. UI For heatControl Activity (with fragments)

For the Android application, the app was built in android studio using Arduino and Bluetooth libraries. Most of the classes and functions are built into android. The first activity, the splash screen is used to initialize a new instance of the device list search. The Device list class initiates the bluetooth protocol checking whether the device currently supports bluetooth and lists them if available. Further, when paired

the heat control class takes over and continues to check the bluetooth serial protocol in the background and turns on/off and throttles the temperature level using commands sent to the arduino through the serial protocol using the slider. The battery activity checks the battery capacity and outputs it as a percentage. Finally there exists an Instruction fragment which contains directions on how to use the app which runs concurrently with the control/status activity allowing the user to easily go back and forth between features. The disconnect button allows the user to connect to a different device or reconnect to the same one as well as close the serial connection to the HC-05 module (and Arduino).

5 Safety

5.1 Possible Hazards

There are quite a few hazards associated with our current design. Since we are using an active heating element connected to a lithium battery with additional circuitry and plan to make the final prototype waterproof and rugged there are multiple concerns:

5.1.1 Battery

Lithium-ion batteries may fail in both non-energetic and energetic states. These may occur as a result of:

- Thermal abuse: if external heating from the element reaches the battery and affects its temperature
- Mechanical abuse: folding and puncturing of the battery could cause rapid reaction with air
- Electrical abuse: improper wiring could cause overcharge, external short circuit, over discharge
- Poor electrochemical design: problems with the battery cathode-anode balance
- Internal cell faults: manufacturing defects (e.g. Metallic particles)

Usually one of the following occur:

- Loss of capacity: extremely low temperatures could cause complete discharge

- Internal impedance increase: the internal resistances or improper contacts could cause overheating and lead to thermal runaway or buildup of charge outside the battery
- Activation of permanent disabling mechanism: the battery could be overcharged, to an extent that the internal fuse is burnt and the battery is disabled
- Thermal runaway: the biggest concern is a puncture or leak causing rapid oxidation causing a fire hazard leading to chemical burns, this is the most important hazard that the system must protect against, the casing of the battery is not easily penetrated and would take forceful and purposely effort to cause the casing to be punctured.

5.1.2 Audio interface and bone conduction headphones

Usually, there are no major failures or hazards associated with headphones since they use relatively low power, but since we are using bone conduction technology, we need to make sure that our design is safe to use for extended wear and proper circuitry so there is no shock hazard.

5.1.3 Heating element

The heating element is going to be physically close to the user and needs to perform smoothly.

Overheating is a major concern and multiple controls must be deployed so burning or melting do not occur, but this has been protected by limiting the maximum temperature.

5.1.4 Power supply and microcontroller

Inaccurate readings and too much power delivery to the system is the major concern in this part as it could cause multiple failures and disable the system completely. Redundant measures must be taken here so that the power delivery is perfect.

6 Testing

The system as whole needed to be tested as one entity because the heating pads react differently depending on which environment it is placed in. The team conducted basic testing of the functionality of the jacket; if it turned on and off with the app and if it was able to attain the specified temperatures when the slider was moved around. The lithium ion battery was tested at various temperatures to see if it could power the different temperatures. The microcontroller and Bluetooth were tested indoors and outdoors to see how well connections operate in various conditions, which worked fine in both conditions. If we had more time we would have tested the jacket in extreme and controlled temperatures, not just in the lab, but in a freezer, with wind applied.

7 Results

So far the Bluetooth connection has worked properly allowing the Android App to talk to the jacket. With this, the user is able to control the temperature to three different settings 70 °F , 80°F and 90 °F. The heating pads can also get to these temperatures at a relatively fast rate all at the same time.

8 Future Work

If the team had more time, we would have integrated the circuit into a basic integrated circuit that would include the microcontroller, a Bluetooth module, an H-Bridge, and a power control. This would reduce the size of our design making it fit easier into the jacket. We would also use a smaller battery pack as mentioned earlier, to make the jacket lighter.

Finally, we would go through more thorough testing in fixed temperatures, in extreme weather conditions, how it responds to water, drop tests, and further battery capacity testing.

In terms of software, we would use a more complex control system that takes in data from a weather API and then calculates the output current and voltage from the PWM. In addition, more features to the app and failsafe arduino code would be tested to ensure efficiency and dependability.

9 Design Norms

9.1 Our Approach

For our design, certain standards were kept in mind while developing. These standards include trust, integrity and care. These standards were chosen because they felt most important to the team to develop a well rounded reputable product.

9.2 Design Norms Explained

9.2.1 Trust

Since the design deals with heat, there are a lot of risks involved with battery powering. So, the team decided to use trust as a design norm, to show awareness of the possible dangers and commit to prevent hazards as much as possible through enough research, testing and well known series of prevention methods.

9.2.2 Care

The team is planning to make the project the most cost-effective jacket, so that lower income people would be able to afford the jacket. Since people from socio-economic backgrounds experience the cold from winter, with an affordable jacket it should be available and affordable.

9.2.2 Integrity

The team stands with integrity because the norm ties care and trust together. The design will stand to be very honest, and transparent to the customers.

10 Project Management

10.1 Team Organization

As a team, we decided to split responsibilities and tasks by interests and strengths. Chinelo Ezenwelu worked on the integrated circuit, heating element and power management as well as the business aspect of the project, team management and the requested deliverables for the project. Temitope Ogunnoiki, with her computer science background used her skills in the software design, Android development and Bluetooth interfacing. Shurjo Maitra also with his CS background and skills worked on the app design and integration of the system. We also divided roles for logistical tasks, Chinelo was active in organizing team meetings and managing deadlines. Temi handled parts of the research and worked on the team's budget and website. Shurjo focused on the software design and contributed in areas where needed. The functions described are not purposed to restrict roles but to allow team integration, efficiency and transparency in tasks.

We held meetings during our scheduled work period in the syllabus, usually between 2:30 - 3:30pm, Monday - Friday. Our meetings had three different structure; we met on some occasions to discuss the project, results and next steps. We also met on other occasions to work on the design and lastly, we worked individually on parts of the design. Besides the class scheduled worked period, we met frequently to work on the design.

We stored most of our research findings, code and other design documents in the shred group drive and created a duplicate for public viewing in the shared department folder whose path is the following:
file:///S:/Engineering/Teams/Team05.

A chart showing the organization of team 05 is shown in Figure 5 below:

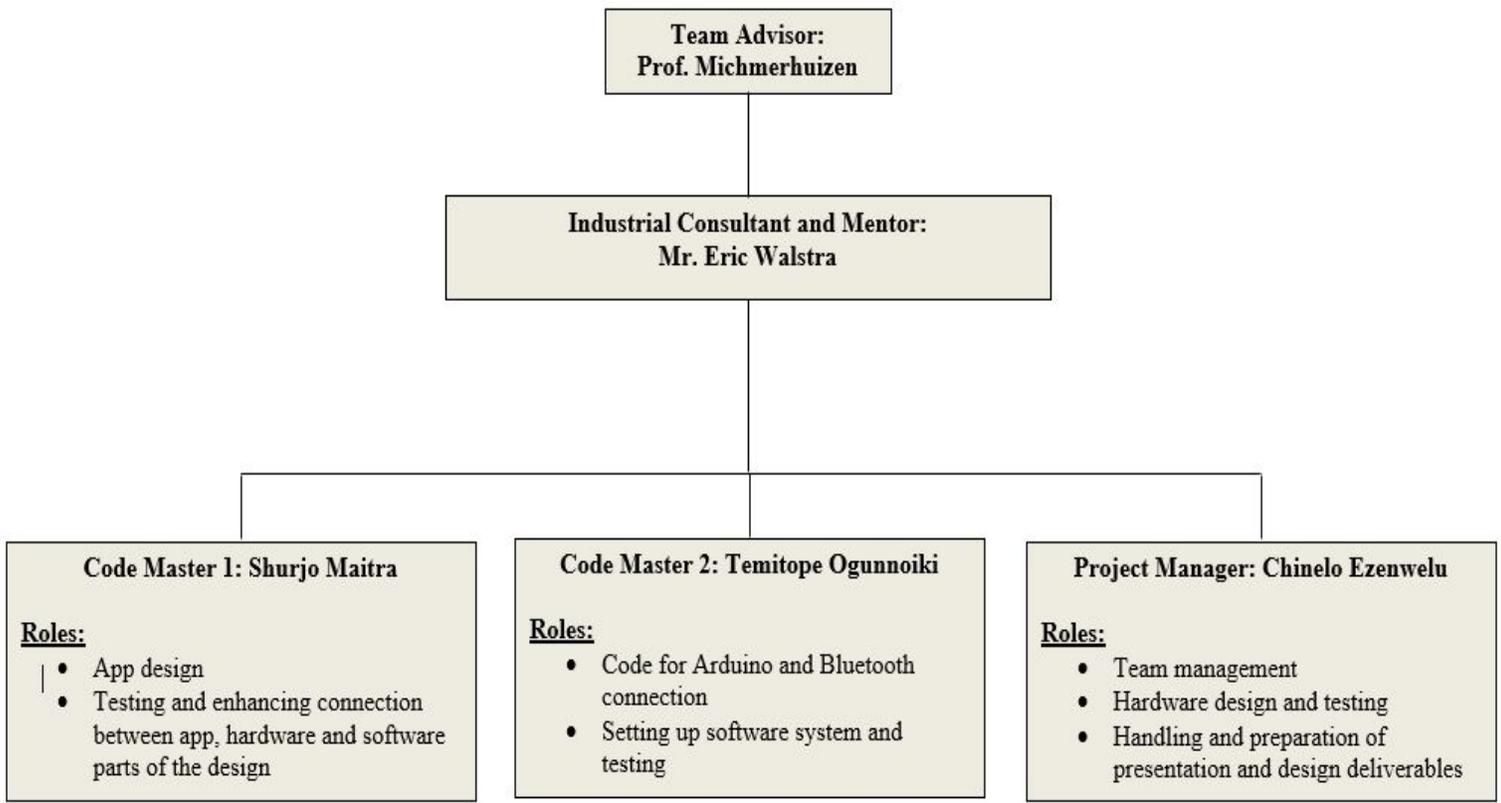


Figure 8: Team 05 Organization Chart

10.2 Advisors

The team was advised by Prof. Mark Michmerhuizen at Calvin College. He has worked at Johnson Controls and Dematic most recently as a project manager including other roles. The team’s industrial advisor is Eric Walstra, an Engineering Manager at Gentex. Prof. Leonard De Rooy acts as the faculty advisor for the senior design course. Each advisor provided important insight and referred the team to resources that aided the progress of the project. Additionally, the team had access to the engineering faculty as well as library resources.

10.3 Schedule

The team designed a work schedule using Microsoft Project Gantt Chart to break down the tasks required. The schedule is located in Appendix A.2 . Also the team used Trello Board, to assign tasks and alert the group upon designation of completed task. See appendix A.2. for more information. During the course of the design phase it was difficult to adhere strictly to the schedule however as a team we were flexible and adapted to whatever changes we made.

11 Business Plan

11.1 Budget

See appendix A.1. for budget analysis. Due to time constraint and the complexity of our project, we made some changes to our design hence, purchased less items that we projected. More information on our expenditure can be seen in A.1.

11.2 Market Analysis

We plan on utilizing more cost effective materials for our product to bring down our costs and potentially the price of our product in comparison to the current market price.

11.2.1 Market Survey

One of the things we did during the research phase was to carry out a survey by word of mouth to see if there was a demand or need for our design. The moment we created an awareness for the problem of

bulking during the winter, we found that there was a great need for our product however people were not willing to invest in a heating jacket at the current market price. We interacted with about 150 people on Calvin College's campus and used the data from their response to further our design.

11.3 Competition

A fair amount of companies such as Polar Sealand Milwaukee already have existing products in the market that provide heating in various clothing at quite a high price. Levis has a commuter jacket that has smart navigation features powered by Google. But so far there is no single product that delivers a practical and smart solution to clothing that consumers increasingly demand. There are a few startups that are aimed at creating a similar product but so far, no single outerwear company has monopolized on this demand or has developed technology that is satisfactory to consumers of all income levels. Figure 6 below shows the different jackets currently in the market and the pricing for these jackets.

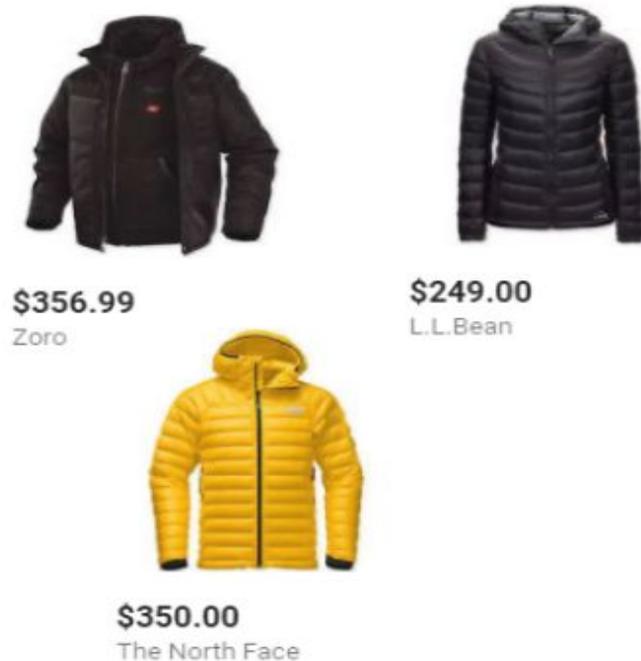


Figure 9: Competing Products in the Market (see ref in Appendix)

11.4 Cost Estimate

The costs estimated in this section comprise mostly the expenses incurred in designing the prototype as well as the engineering design hours used in the production process. Additional costs, however, would be required to improve the aesthetics, safety and comfort of the product before putting it out in the market. Therefore, our expenses in the design process can be defined as development and production costs.

11.5 Development Costs

The Team's development costs comprises two parts: time costs and money costs. The time costs include costs specific to the preliminary research, coding, design, and testing.

Preliminary research costs include all of the time spent on research which includes research on component parts for the design such as the heating pads, HC-05, arduino and so on. This also includes time spent on

researching the compatibility of the system and potential source codes for our design.

The preliminary research includes mostly the time spent determining project feasibility both at the beginning of and during the project. Approximately, a total of 150 hours (at roughly 5 hrs per week per team member over the course of 7 months from Sept - Mar) from all team members was dedicated to this kind of research. The monetary equivalent of the team's time is valued at a rate similar to that of an hourly-waged entry level engineer. At \$30/hr, a total of \$4,500 was spent on preliminary research. Coding costs could potentially be incorporated into design costs however this comprised the bulk of the team's time. The team spent about 600 hours (at roughly 20 hrs per week over the course of 9 months from Sept - May) on this part of the design process. This included time spent researching useful code which was not entirely helpful as there wasn't a lot of code available online for our design and so the team developed code from scratch for the project. This also required a lot of debugging and testing. Lastly, the coding design costs included the time spent on app design, code design for the Arduino and the system, and time spent testing and debugging. Using the standard of \$30/hr, this results in a monetary cost of \$18,000.

Design and Testing costs come down to approximately 300 hrs. This includes hardware design, testing and completion of the system. At \$30/hr the monetary costs are approximately \$9,000.

Total time costs come to \$31,500 in monetary value.

The money/actual costs for the design are shown in the table below

The total development cost then equals \$31,659.91 rounded to \$40,000 to cover for miscellaneous activities.

Table 3: Actual Costs for Design

Date	Purchaser	Team No.	Website	Part	Qty	PPQ	Total Price	Shipping	Total
11/13/17	C. Ezenwelu	5	Digi-Key	Sparkfun PurpleTooth Jamboree	1	\$49.95	\$49.95	\$8.99	\$58.94
2/14/18	Temi Ogunnoiki	5	SparkFun	Com-11289	2	\$4.95	\$9.90	\$6.16	\$16.06
3/12/18	Temi Ogunnoiki	5	Amazon.com	HC-05 (DSD)	1	\$8.99	\$8.99		\$8.99
4/6/18	Temi Ogunnoiki	5	SparkFun	Heating Pad	3	\$4.95	\$14.85	\$9.59	\$24.44
4/7/18	Temi Ogunnoiki	5	Amazon.com	Motor Drive Module	1	\$15.50	\$15.50		\$15.50
5/2/18	Temi Ogunnoiki	5	Amazon.com	Talentcell Rechargeable	1	\$29.99	\$29.99	\$5.99	\$35.98
								TOTAL	\$159.91

11.6 Production Costs

Variable Cost of Goods Sold:

Raw Material = \$100.97/unit (minus PurpleTooth Jamboree: -\$58.94)

Direct Labor = \$8.15/hour (0.5 hours/unit)

Utilities (Electricity, Water & Gas) = \$21/hour (0.5 hours/unit)

Total Cost per Unit = \$115.54/unit

Selling Price of Hot-JAC = \$120

Profit Per Unit = \$5

12 Conclusion

Based on the success of our present prototype, our team believes that there is a future for smart heating jackets that are affordable for consumers. At our price of \$120, we believe we achieved our goal of producing a jacket at a price lower than the market standard. If more companies try to enter the market, the prices would surely go down. The implementation of a number of cheap parts has made the system affordable. During our presentation on senior design night, we had almost everyone that mentioned they would love to buy the jacket if it was on the market. Hot-JAC would be especially useful in a region like Michigan, and would definitely be a profitable business if invested in.

13 Acknowledgements

The team would like to officially thank everyone who has been a direct and indirect part of the success and growth of this project, starting with our professor and advisor, Prof. Mark Michmerhuizen, for his patience in working with our team and his constructive feedback throughout the project.

The team is also grateful for our industrial Consultant, Mr. Eric Walstra, for his profession wisdom and advice; Audrey Waldron, the team's consultant at the career center, for helping us prepare adequately for the transition between student and professional life; David Malone, the team's consultant at the library for the help with research and discovery materials important for the project; all the senior design professors - Prof. VanAntwerp, Prof. De Rooy, Prof. Tubergen and Prof. Nielsen - for coaching all the senior design teams throughout this process and lastly, our peers for continuous support and encouragement.

14 References

“Ultraflex Heaters.” Thermo Heating Elements, thermo-llc.com/ultraflex-heaters/

Senol, Yavuz, et al. "Active T-Shirt." *International Journal of Clothing Science and Technology* 23.4 (2011): 249-57. ProQuest. Web. 5 Dec. 2017.

“Understanding the BC127 Bluetooth Module”.

<https://learn.sparkfun.com/tutorials/understanding-the-bc127-bluetooth-module/arduino-library>

Jaeho Lee, D. W. Gerlach and Y. K. Joshi, "Parametric thermal modeling of heat transfer in handheld electronic devices," *2008 11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, Orlando, FL, 2008, pp. 604-609.

K. Kukkonen, T. Vuorela, J. Rantanen, O. Ryyndnen, A. Siffi and J. Vanhala, "The design and implementation of electrically heated clothing," *Proceedings Fifth International Symposium on Wearable Computers*, Zurich, 2001, pp. 180-181.

Codepath. “Codepath/android_guides.” GitHub, github.com/codepath/android_guides/wiki/ViewPager-with-FragmentPagerAdapter.

Girish, Mayoogh. “Bluetooth Basics: How to Control an LED Using a Smartphone and Arduino | Arduino.” *Maker Pro*, Maker Pro, 9 Apr. 2018, maker.pro/arduino/tutorial/bluetooth-basics-how-to-control-led-using-smartphone-arduino.

hishriTaha. “HishriTaha/AndroidLEDcontrol.” *GitHub*, github.com/hishriTaha/AndroidLEDcontrol.

Schneider, Brian, et al. “A Guide to Understanding LiPo Batteries.” Roger's Hobby Center, 5 Mar. 2015, rogershobbycenter.com/lipoguide/.

Vallerand, A.L., Savourey, G., Hanniquet, AM. et al. *Europ. J. Appl. Physiol.* (1992) 65: 286. <https://doi.org/10.1007/BF00705095>

15 Appendices

A.1 Budget Review

Table 2 : Budget Analysis

Part required	Sub-parts	Current part scoped	Unit Price	No.	Source	Use	Comments
Microcontroller	-	Raspberry Pi 3	\$34.49	2	Amazon.com	To connect all components (heating element, Bluetooth headphones, power supply etc) and communicate to phone app.	Still not finalized, Arduino Uno was tested initially so current choice is tentative.
Bone conduction Transducer and amplifier	Audio amplifier	Stereo 2.8W Class D Audio Amplifier - I2C Control AGC - TPA2016	\$9.95	1	https://www.adafruit.com/product/1712	Needed to drive the transducers and deliver a high power audio experience and pass signals.	Not totally settled on this amplifier might need a lower power one depending on the output from the microcontroller.
	Transducer	Bone Conductor Transducer with Wires - 8 Ohm 1 Watt	\$8.95	2	https://www.adafruit.com/product/1674	To produce sound and attach to the amplifier or microcontroller and receive sound through bluetooth.	Not finalized as, usability and reliability is unknown for this particular model.
Battery	Battery and charger	<u>Li-ion battery and charger</u>	\$29.99	2	ebay.com	To provide power and energy to the whole system	-
	Battery Casing	Lithium Battery Casing	\$16.99	1	https://www.pine64.org/?product=lithium-battery-casing	To give the battery protection from puncture	
Power management system	Li-ion power IC	<u>Li-ion Power System</u>	\$40	1	http://www.microchipdirect.com/product/search/all/MCP7383XRD-PPM	Needed to supply power from the Li battery to the other components and to record information on the system	<u>Datasheet for future Design Reference</u>

Material and clothing costs	Gore-tex/waterproof material	EMMAKITES 60"x36"(WxL) 40D Ripstop Nylon	\$8.95	1	amazon.com	Used as the external layer to provide protection from the elements	This fabric might be sewn onto a jacket or just used for testing. A jacket will be used to test the entire system.
	Fleece layer	LA-Linen: 60-Inch Solid Anti-Pill Polar Fleece Fabric By	\$4.98	1	amazon.com	Used as bottom most layer to provide comfort and insulation. The heating element	This specific fabric has not yet been tested with the heating element and
		The Yard, Black				will also be placed in between this and the waterproof layer	might change in the future, depending on safety and heat conservation
	Sewing materials	CP-nine Bonded Nylon Sewing Thread 1500 Yard Size #69 T70	\$8.69	1	amazon.com	Used to bond together the materials and ensure the layers don't fall apart. Also used to hold the various components in place	This will be an essential tool to fit components well onto the jacket.
	Jacket	Port Authority Men's Lightweight Jacket(M)	\$12.49	1	amazon.com	Used to construct our final prototype and build the components onto	The current choice is a generic choice, likely to change in the future
Switches/buttons /peripherals	-	Calvin 3D printing	~\$15			Budget for 3D printing of buttons and switches	-
Cables and misc. Electrical components	-	Heat resistant wires, capacitors, resistors, LEDs, misc. modules/ connectors	~\$50	-	Various sources	To connect to breadboard and between components and use indicators as flags to test proper functioning of system.	So far these have not been quantified and will most likely be obtained from lab storage.
Bluetooth interface	-	Sparkfun BC127 breakout board	\$50		Digikey.com	Used to transfer high quality audio over Bluetooth 4.2	Did not function as intended for Engr. 325 project, will probably have to switch to different board
PCB fabrication/ processing	-	Sparkfun PCB	\$2.95	3	Digikey.com	For fabrication purposes	-
Heating Element	-	sparkfun heating material	\$100	10	Sparkfun	Sewn into material	

TOTAL			\$463.81				
-------	--	--	----------	--	--	--	--

A.2 Schedule

Project Management for "Smart Hoodie"	174 days?	Wed 9/6/17	Sat 5/5/18		
Initiating	15 days	Wed 9/6/17	Tue 9/26/17		
Primary Research	9 days	Wed 9/6/17	Mon 9/18/17		
Identify Goals and Objectives	1 day	Wed 9/6/17	Wed 9/6/17		Everyone
Pitch in Design Ideas	1 day	Thu 9/7/17	Thu 9/7/17	3	Everyone
Evaluate different Project Options	1 day	Fri 9/8/17	Fri 9/8/17	4	Everyone
Decide on a Final project	6 days	Mon 9/11/17	Mon 9/18/17	5	Everyone
Design Research	6 days	Tue 9/19/17	Tue 9/26/17	2	
Research on other similar products	1 day	Tue 9/19/17	Tue 9/19/17		Shurjo
Determine need for Hoodie	1 day	Wed 9/20/17	Wed 9/20/17	8	Shurjo
Determine Usability	1 day	Thu 9/21/17	Thu 9/21/17	9	Temi
Find related research	2 days	Fri 9/22/17	Mon 9/25/17	10	Everyone
Prepare Preliminary Project Scope Statement	1 day	Tue 9/26/17	Tue 9/26/17	11	Everyone
Planning	31 days	Wed 9/27/17	Wed 11/8/17	1	
Design Research	11 days	Wed 9/27/17	Wed 10/11/17		
Determine which material would be used (Hardware & Software)	5 days	Wed 9/27/17	Tue 10/3/17		Everyone
Create Initial Part List	3 days	Wed 10/4/17	Fri 10/6/17	15	Chinelo
Analyse different possible solutions to go about project	3 days	Mon 10/9/17	Wed 10/11/17	16	Temi
Define Scope	5 days	Thu 10/12/17	Wed 10/18/17	14	
Identify feasible features for hoodie	2 days	Thu 10/12/17	Fri 10/13/17		Everyone
Specify Deliverables and Acceptance Criteria	1 day	Mon 10/16/17	Mon 10/16/17	19	Shurjo
Define Scope	1 day	Tue 10/17/17	Tue 10/17/17	20	Chinelo
Document Assumptions	1 day	Wed 10/18/17	Wed 10/18/17	21	Temi
Develop Project Schedule	5 days	Thu 10/19/17	Wed 10/25/17	18	
Build Work Breakdown Structure	3 days	Thu 10/19/17	Mon 10/23/17		Temi
Determine Budget estimate	1 day	Tue 10/24/17	Tue 10/24/17	24	Chinelo
Start Trello to keep track of project tasks	1 day	Wed 10/25/17	Wed 10/25/17	25	Shurjo
Develop Risk Plans	3 days	Thu 10/26/17	Mon 10/30/17	23	
Identify Risks	1 day	Thu 10/26/17	Thu 10/26/17		Everyone
Analyze Risks	1 day	Fri 10/27/17	Fri 10/27/17	28	Everyone
Fill in Hazard Form	1 day	Mon 10/30/17	Mon 10/30/17	29	Temi

4 Organize Project Resources	1 day	Tue 10/31/17	Tue 10/31/17	27	
Develop Organization Structure	1 day	Tue 10/31/17	Tue 10/31/17		Shurjo & Chienlo
4 Project Development	6 days	Wed 11/1/17	Wed 11/8/17	31	
Determine Procurement Requirements	1 day	Wed 11/1/17	Wed 11/1/17		Shurjo
Create Poster for Engineering Day	3 days	Thu 11/2/17	Mon 11/6/17	34	Everyone
Create Proposal	2 days	Tue 11/7/17	Wed 11/8/17	35	Shurjo & Chienlo
4 Executing	11 days	Thu 11/9/17	Thu 11/23/17	13	
4 Design	8 days	Thu 11/9/17	Mon 11/20/17		
Obtain Resources	1 day	Thu 11/9/17	Thu 11/9/17		Everyone
Create and Design Schematics	6 days	Fri 11/10/17	Fri 11/17/17	39	
Assign Responsibilities	1 day	Mon 11/20/17	Mon 11/20/17	40	Everyone
Start Building prototype	1 day	Thu 11/9/17	Thu 11/9/17		
Test as you go	1 day	Thu 11/9/17	Thu 11/9/17		
4 Manage Team Performance	2 days	Tue 11/21/17	Wed 11/22/17	38	
Be sure to have bonding time outside project!	1 day	Tue 11/21/17	Tue 11/21/17		Everyone
Recognize Success	1 day	Wed 11/22/17	Wed 11/22/17	45	Everyone

Manage Project Communication	1 day	Thu 11/23/17	Thu 11/23/17	44	
Review Progress	1 day	Thu 11/23/17	Thu 11/23/17		Everyone
Monitoring and Controlling	14 days	Fri 11/24/17	Wed 12/13/17	37	
Monitor and Control Project Work	3 days	Fri 11/24/17	Tue 11/28/17		
Authorize Work	1 day	Fri 11/24/17	Fri 11/24/17		Everyone
Manage Action Items	1 day	Mon 11/27/17	Mon 11/27/17	51	Shurjo
Manage Project Records	1 day	Tue 11/28/17	Tue 11/28/17	52	Chinelo
Schedule Control	3 days	Wed 11/29/17	Fri 12/1/17	50	
Track Status	1 day	Wed 11/29/17	Wed 11/29/17		Everyone
Maintain Project Schedule	1 day	Thu 11/30/17	Thu 11/30/17	55	Everyone
Maintain Work Plans	1 day	Fri 12/1/17	Fri 12/1/17	56	Everyone
Manage Finances	1 day	Mon 12/4/17	Mon 12/4/17	54	
Monitor Cost/Schedule Variance	1 day	Mon 12/4/17	Mon 12/4/17		Everyone
Perform Quality Control	5 days	Tue 12/5/17	Mon 12/11/17	58	
Build Tests	5 days	Tue 12/5/17	Mon 12/11/17		Shurjo
Monitor Cost/Schedule Variance	1 day	Mon 12/4/17	Mon 12/4/17		Everyone
Perform Quality Control	5 days	Tue 12/5/17	Mon 12/11/17	58	
Build Tests	5 days	Tue 12/5/17	Mon 12/11/17		Shurjo
Participate in Testing	1 day	Tue 12/5/17	Tue 12/5/17		Everyone
Measure Quality Levels	1 day	Wed 12/6/17	Wed 12/6/17	62	Everyone
Manage Project Team	2 days	Tue 12/12/17	Wed 12/13/17	60	
Manage Communications	1 day	Tue 12/12/17	Tue 12/12/17		Chinelo
Conduct Team Performance Reviews	1 day	Wed 12/13/17	Wed 12/13/17	65	Everyone with Professor
Closing	103 days	Thu 12/14/17	Sat 5/5/18	49	
Close Project	3 days	Thu 12/14/17	Mon 12/18/17		
Feedback on Team performance	1 day	Thu 12/14/17	Thu 12/14/17		Everyone with Professor
Make sure budget List is updated properly	1 day	Fri 12/15/17	Fri 12/15/17	69	Everyone
Start working on Project Report	1 day	Mon 12/18/17	Mon 12/18/17	70	Everyone
Contract Closure	88.5 days	Wed 1/3/18	Sat 5/5/18	68	
Senior Design Night	1 day	Sat 5/5/18	Sat 5/5/18		Everyone

Figure 4: Timeline for Project

A.3 Arduino Code

```
#include <SoftwareSerial.h>

char data = 0;           //Variable for storing received data
int sensorPin = A0;
SoftwareSerial BTserial(0, 1); // RX | TX

int RPWM=5;
int LPWM=6;

// timer 0
int L_EN=7;
int R_EN=8;

void setup()
{
  pinMode(13, OUTPUT);
  for(int i=5;i<9;i++){
    pinMode(i,OUTPUT);
  }
  Serial.begin(9600);           //Sets digital pin 13 as output pin
  BTserial.begin(9600);        //Sets the data rate in bits per second (baud) for
  serial data transmission
}

void loop()
{
  float reading, voltage, degreesC, degreesF;
  reading = analogRead(sensorPin);
  voltage = reading * 5.0;
  voltage /= 1024.0;
  degreesC = (voltage - 0.5) * 100.0;
  degreesF = degreesC * (9.0/5.0) + 32.0;

  if(degreesF < 49){           //reading values from sensor and adjusting duty cycle
    analogWrite(RPWM, 20);
  }
  if(degreesF > 49 and degreesF < 51){
    analogWrite(RPWM, 25);
  }
  if(degreesF > 51 and degreesF <= 53){
    analogWrite(RPWM, 30);
  }
  if(degreesF > 53 and degreesF <= 55){
```

```

        analogWrite(RPWM, 40);
    }
    if(degreesF > 55 and degreesF <= 57){
        analogWrite(RPWM, 45);
    }
    if(degreesF > 57 and degreesF <= 59){
        analogWrite(RPWM, 55);
    }
    if(degreesF > 59 and degreesF <= 61){
        analogWrite(RPWM, 65);
    }
    if(degreesF > 61 and degreesF <= 63){
        analogWrite(RPWM, 75);
    }
    if(degreesF > 63 and degreesF <= 65){
        analogWrite(RPWM, 85);
    }
    if(degreesF > 65 and degreesF <= 70){
        case 1;
    }
    if(degreesF > 70 and degreesF <= 80){
        case 2;
    }
    if(degreesF > 80 and degreesF <= 90){
        case 3;
    }
    }
    if(degreesF > 90){
        data == '0'; // go off
    }
}

if(Serial.available() > 0) // Send data only when you receive data:
{
    data = Serial.read(); //Read the incoming data and store it into variable data
    Serial.println(data); //Print Value inside data in Serial monitor
    Serial.println("EN High");

    if(data == '1'){ //Turns on system
        digitalWrite(R_EN,HIGH);
        digitalWrite(L_EN,HIGH);
        digitalWrite(13, HIGH); //If value is 1 then LED turns ON
        for(int i=0;i<256;i++){
            analogWrite(RPWM,i);
            //analogWrite(LPWM,i);
        }
    }
    int mode = mode == 3 ? 0: mode; //should be replaced to assign the nodes

```

```

switch (mode) {
case 1:
    analogWrite(RPWM, 90); //35% duty cycle
    digitalWrite(R_EN,HIGH);
    digitalWrite(L_EN,HIGH);
    break;

case 2:
    analogWrite(RPWM, 100); //50% duty cycle
    digitalWrite(R_EN,HIGH);
    digitalWrite(L_EN,HIGH);
    break;

case 3:
    analogWrite(RPWM, 110); //66% duty cycle
    digitalWrite(R_EN,HIGH);
    digitalWrite(L_EN,HIGH);
    break;
}
}
else if(data == '0') { //Turns off system
digitalWrite(13, LOW); //If value is 0 then LED turns OFF
Serial.println("EN LOW");
digitalWrite(R_EN,LOW);
digitalWrite(L_EN,LOW);
}

}
}

```

A.4 Android code

```

package edu.calvin.sm47.hotjac;
/* heatControl.java contains bluetooth methods to display jacket status
 * Shurjo Maitra, for Senior Design Project at Calvin College.
 * Date: 5.10.18
 */

import android.app.ProgressDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.view.ViewPager;

```

```

import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ImageButton;
import android.widget.SeekBar;
import android.widget.Toast;

import java.io.IOException;
import java.io.OutputStream;
import java.util.UUID;

import static edu.calvin.sm47.hotjac.R.id.seekHeatControl;

public class heatControl extends AppCompatActivity {

    // Button btnOn, btnOff, btnDis;
    ImageButton On, Off, Disconnect;
    String address = null;
    private ProgressDialog progress;
    BluetoothAdapter myBluetooth = null;
    BluetoothSocket btSocket = null;
    private boolean isBtConnected = false;
    OutputStream taOut;
    SeekBar seekHeatLevel;

    //SPP UUID. Look for it
    static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        //Makes the title bar disappear
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_status);

        //Adds pager adapter support to main activity
        PagerAdapter adapter = new PagerAdapter(getSupportFragmentManager());
        ViewPager pager = (ViewPager) findViewById(R.id.pager);
        pager.setAdapter(adapter);
        pager.setCurrentItem(1);

        Intent newint = getIntent();
        address = newint.getStringExtra(DeviceList.EXTRA_ADDRESS); //receive the address of the bluetooth
device

        //view of the padControl
        setContentView(R.layout.activity_heat_control);

```

```

//call the widgets
On = (ImageButton)findViewById(R.id.on);
Off = (ImageButton)findViewById(R.id.off);
Disconnect = (ImageButton)findViewById(R.id.disconnect);
seekHeatLevel = (SeekBar) findViewById(seekHeatControl);

seekHeatLevel.setMax(3);

new ConnectBT().execute(); //Call the class to connect

// getting seekbar current value

seekHeatLevel.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{
    int currentVal = 0 ;
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b)
    {
        currentVal = i ;
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

        sendToBluetooth( 100, currentVal );

        Toast.makeText(getApplicationContext(), "Heat: "+ currentVal, Toast.LENGTH_SHORT).show();
    }
});

//commands to be sent to bluetooth
On.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        turnOnPad(); //method to turn on
    }
});

Off.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v)
    {
        turnOffPad(); //method to turn off
    }
});

Disconnect.setOnClickListener(new View.OnClickListener()

```

```

    {
        @Override
        public void onClick(View v)
        {
            Disconnect(); //close connection
        }
    });
}

private void sendToBluetooth(int heat, int intensity) {
    {
        //make sure there is a paired device
        if ( btSocket != null )
        {
            try
            {
                taOut = btSocket.getOutputStream();
                taOut.write(heat + intensity);

                taOut.flush();
            }catch(IOException ioe)
            {
                Log.e( "app>" , "Could not open a output stream "+ ioe );
            }
        }
    }
}

private void Disconnect()
{
    if (btSocket!=null) //If the btSocket is busy
    {
        try
        {
            btSocket.close(); //close connection
        }
        catch (IOException e)
        { msg("Error");}
    }
    finish(); //return to the first layout
}

private void turnOffPad()
{
    if (btSocket!=null)
    {
        try
        {
            btSocket.getOutputStream().write("0".toString().getBytes());
        }
        catch (IOException e)
        {
            msg("Error");
        }
    }
}
}

```

```

private void turnOnPad()
{
    if (btSocket!=null)
    {
        try
        {
            btSocket.getOutputStream().write("1".toString().getBytes());
        }
        catch (IOException e)
        {
            msg("Error");
        }
    }
}

// fast way to call Toast
private void msg(String s)
{
    Toast.makeText(getApplicationContext(),s, Toast.LENGTH_LONG).show();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_led_control, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

private class ConnectBT extends AsyncTask<Void, Void, Void> // UI thread
{
    private boolean ConnectSuccess = true; //if it's here, it's almost connected

    @Override
    protected void onPreExecute()
    {
        progress = ProgressDialog.show(heatControl.this, "Connecting...", "Please wait."); //show a
        progress dialog
    }
}

```

```

    @Override
    protected Void doInBackground(Void... devices) //while the progress dialog is shown, the connection
    is done in background
    {
        try
        {
            if (btSocket == null || !isBtConnected)
            {
                myBluetooth = BluetoothAdapter.getDefaultAdapter();//get the mobile bluetooth device
                BluetoothDevice dispositivo = myBluetooth.getRemoteDevice(address);//connects to the
                device's address and checks if it's available
                btSocket = dispositivo.createInsecureRfcommSocketToServiceRecord(myUUID);//create a serial
                protocol connection
                BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                btSocket.connect();//start connection
            }
        }
        catch (IOException e)
        {
            ConnectSuccess = false;//if the try failed, you can check the exception here
        }
        return null;
    }
    @Override
    protected void onPostExecute(Void result) //after the doInBackground, it checks if everything is ok
    {
        super.onPostExecute(result);

        if (!ConnectSuccess)
        {
            msg("Connection Failed. Try restarting the app.");
            finish();
        }
        else
        {
            msg("Connected.");
            isBtConnected = true;
        }
        progress.dismiss();
    }
}
}

```

```

package edu.calvin.sm47.hotjac;

```

```

/* DeviceList.java contains bluetooth methods to initiate serial protocol with slave device
 * Shurjo Maitra, for Senior Design Project at Calvin College.
 * Date: 5.10.18
 */

```

```

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;

```

```

import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Set;

public class DeviceList extends AppCompatActivity
{
    //widgets declaration
    Button btnPaired;
    ListView devicelist;

    //Bluetooth adapter decalred
    private BluetoothAdapter myBluetooth = null;
    private Set<BluetoothDevice> pairedDevices;
    public static String EXTRA_ADDRESS = "device_address";

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        //Makes the title bar disappear
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_status);
        setContentView(R.layout.activity_device_list);

        //Calling widgets
        btnPaired = (Button)findViewById(R.id.button);
        devicelist = (ListView)findViewById(R.id.listView);

        //If the master device has bluetooth
        myBluetooth = BluetoothAdapter.getDefaultAdapter();

        if(myBluetooth == null)
        {
            //Toast that there is no bluetooth available on the device being used
            Toast.makeText(getApplicationContext(), "Bluetooth Device Not Available",
                Toast.LENGTH_LONG).show();

            //Finish loading apk
            finish();
        }
        //Otherwise if there is bluetooth continue
        else if(!myBluetooth.isEnabled())
        {
            //Ask to the user turn the bluetooth on

```

```

        Intent turnBTon = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(turnBTon,1);
    }

    btnPaired.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v)
        {
            pairedDevicesList();
        }
    });
}

//find the paired devices from previously paired bluetooth devices
private void pairedDevicesList()
{
    pairedDevices = myBluetooth.getBondedDevices();
    ArrayList list = new ArrayList();

    if (pairedDevices.size()>0)
    {
        for(BluetoothDevice bt : pairedDevices)
        {
            list.add(bt.getName() + "\n" + bt.getAddress()); //Get the device's name and the address
        }
    }
    else
    {
        Toast.makeText(getApplicationContext(), "No Paired Bluetooth Devices Found.",
Toast.LENGTH_LONG).show();
    }

    final ArrayAdapter adapter = new ArrayAdapter(this,android.R.layout.simple_list_item_1, list);
    devicelist.setAdapter(adapter);
    devicelist.setOnItemClickListener(myListClickListener); //Method called when the device from the
list is clicked
}

//
private AdapterView.OnItemClickListener myListClickListener = new AdapterView.OnItemClickListener()
{
    public void onItemClick (AdapterView<?> av, View v, int arg2, long arg3)
    {
        // Get the device MAC address, the last 17 chars in the View
        String info = ((TextView) v).getText().toString();
        String address = info.substring(info.length() - 17);

        // create intent to start activity
        Intent i = new Intent(DeviceList.this, heatControl.class);

        //Change the activity
        i.putExtra(EXTRA_ADDRESS, address); //this will be received at heatControl (class) Activity
startActivity(i);
    }
};

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    // create menu and add items to device list
    getMenuInflater().inflate(R.menu.menu_device_list, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    //action bar clicks are expanded from here
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
}

package edu.calvin.sm47.hotjac;

/* PagerAdapter.java allows the project to use fragments in addition to activities
 * Shurjo Maitra, for Senior Design Project at Calvin College.
 * Date: 5.10.18
 */
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentStatePagerAdapter;

/**
 * Created by sm47 on 3/28/2018.
 */

public class PagerAdapter extends FragmentStatePagerAdapter {
    private static final int NUM_ITEMS = 3;
    public PagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        switch(position) {
            case 0: return new InstructionFragment();
            case 1: return new StatusFragment();
            case 2: return new BatteryFragment();
            default: return new StatusFragment();
        }
    }

    @Override
    public int getCount() {

```

```

        return NUM_ITEMS;
    }
}

package edu.calvin.sm47.hotjac;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;

/* SplashScreen.java initializes app with an image toast
 * Shurjo Maitra, for Senior Design Project at Calvin College.
 * Date: 5.10.18
 */
public class SplashScreen extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        //Makes the title bar disappear
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_status);
        setContentView(R.layout.splash);

        //Runs the splash screen for 600 ms
        Thread timerThread = new Thread(){
            public void run(){
                try{
                    sleep(600);
                }catch(InterruptedException e){
                    e.printStackTrace();
                }finally{
                    Intent intent = new Intent(SplashScreen.this,DeviceList.class);
                    startActivity(intent);
                }
            }
        };
        timerThread.start();
    }

    @Override
    protected void onPause() {
        // TODO Auto-generated method stub
        super.onPause();
        finish();
    }
}

package edu.calvin.sm47.hotjac;
/* InstructionFragment.java runs continuously to make instructions easily accessible
 * Shurjo Maitra, for Senior Design Project at Calvin College.
 * Date: 5.10.18

```

```

*/

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ListView;

import java.util.Set;

public class InstructionFragment extends Fragment {
    Button btnPaired;
    ListView devicelist;

    private BluetoothAdapter myBluetooth = null;
    private Set<BluetoothDevice> pairedDevices;
    public static String EXTRA_ADDRESS = "device_adress";

    public InstructionFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_instruction, container, false);
    }
}

package edu.calvin.sm47.hotjac;

/* BatteryFragment.java allows battery data to be read in
 * Shurjo Maitra, for Senior Design Project at Calvin College.
 * Date: 5.10.18
 */

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.BatteryManager;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class BatteryFragment extends Fragment {

```

```

private TextView batteryTxt;
int health = 0;

public BatteryFragment() {
    // Required empty public constructor
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    View rootView = inflater.inflate(R.layout.fragment_battery, container, false);

    return rootView;
}

@Override
public void onResume() {
    super.onResume();

    batteryTxt=(TextView) getActivity().findViewById(R.id.batteryTxt);
    setare();
}

public void setare(){

    batteryTxt.setText("Health: " + health + "%");

}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    getActivity().registerReceiver(this.batteryInfoReceiver, new
IntentFilter(Intent.ACTION_BATTERY_CHANGED));
}

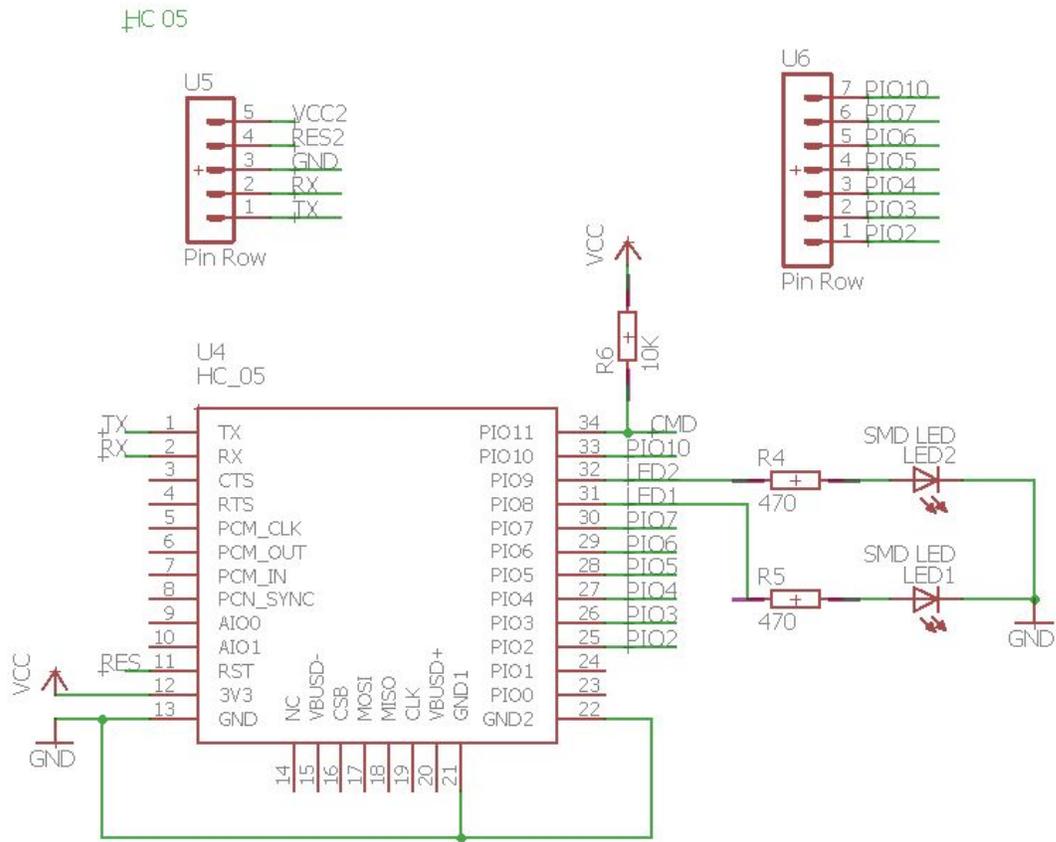
private BroadcastReceiver batteryInfoReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {

        health= intent.getIntExtra(BatteryManager.EXTRA_HEALTH,0);

    }
};
}

```

A.5 PSpice diagrams of Components



Arduino Uno

